On Contention Resolution Protocols and Associated Probabilistic Phenomena*

P. D. MacKenzie *C. G. Plaxton* *R. Rajaraman*

Department of Computer Science
University of Texas at Austin

Abstract

Consider an on-line scheduling problem in which a set of abstract processes are competing for the use of a number of resources. Further assume that it is either prohibitively expensive or impossible for any two of the processes to directly communicate with one another. If several processes simultaneously attempt to allocate a particular resource (as may be expected to occur, since the processes cannot easily coordinate their allocations), then none succeed. In such a framework, it is a challenge to design efficient contention resolution protocols.

Two recently-proposed approaches to the problem of PRAM emulation give rise to scheduling problems of the above kind. In one approach, the resources (in this case, the shared memory cells) are duplicated and distributed randomly. We analyze a simple and efficient deterministic algorithm for accessing some subset of the duplicated resources. In the other approach, we analyze how quickly we can access the given (nonduplicated) resource using a simple randomized strategy. We obtain precise bounds on the performance of both strategies. We anticipate that our results will find other applications.

1 Introduction

Let k balls be thrown independently and uniformly at random into n bins, and let random variable X denote the maximum number of balls landing in any single bin. If $k = \Theta(n)$, it is well known that $X = \Theta(\lg n/\lg\lg n)$ wvhp. (Throughout this paper, we use "wvhp" to mean "with probability at least $1 - n^{-c}$ for *any* positive constant c", and we use "whp" to mean "with probability at least $1 - n^{-c}$ for *some* positive constant c".) If $k = \Theta(n \lg n)$, it is similarly well known that $X = \Theta(\lg n)$ wvhp. (These claims are straightforward to prove using standard bounds on the tail of the binomial distribution [5].) These sharp threshold phenomena have important consequences in a wide variety of hashing-related applications. In this paper, we explore two similar (but more complex) threshold phenomena, one arising in each of two fundamental families of contention resolution protocols.

To make our discussion of contention resolution protocols more concrete, we will focus our attention primarily on a single application, namely that of efficiently emulating an EREW PRAM on a c-collision crossbar network. We begin by reviewing the definitions of these two computational models. An *EREW PRAM* is a collection of n (say) processors along with a global shared memory. Input and output are provided in the shared memory. In a single computational step, each processor can read or write one memory location. The sole restriction is that no two processors are allowed to access the same memory location in a single step. (If two processors attempt to access the same memory location in a single step, then the machine halts.)

A *c-collision crossbar network* (or simply, a c-collision crossbar) is a more realistic model of parallel computation in which the global shared memory is distributed over n disjoint memory modules. Input and output are provided in the distributed memory. Each computational step consists of a read/write phase followed by an acknowledgement phase. During the read/write phase each processor can issue one read or write request for a specific memory location. If the total number of read/write requests involving memory locations stored in any particular memory module M is less than or equal to c, then all requests involving M succeed and are acknowledged during the acknowledgment phase. On the other hand, if more than c processors attempt to access memory locations stored in the same memory module M, then all requests involving M fail and no corresponding acknowledgements are sent. A *c-arbitrary crossbar* is defined in the same manner as a c-collision crossbar, except that if more than c processors attempt to access some memory module M, then an arbitrary subset of c of the requests succeed and are acknowledged.

The c-collision and c-arbitrary crossbar models have

*This research was supported by NSF Research Initiation Award CCR–9111591, and the Texas Advanced Research Program under Grant No. 003658–461 and 003658–480.

been studied previously under different names. (Our terminology is new.) The local memory PRAM model of Anderson and Miller [4], later studied under the name OCPC (optical communication parallel computer) in [7, 8, 9], corresponds to the 1-collision crossbar model. Valiant's S*PRAM model [20] corresponds to the 1-arbitrary crossbar model. Assuming a complete interconnection between processors and memory modules, the c-collision (resp., c-arbitrary) DMM models of [6] corresponds to the c-collision (resp., c-arbitrary) crossbar model.

We now return to the question of efficiently emulating an EREW PRAM on a c-collision crossbar. More specifically, assume that we wish to emulate a k-processor EREW PRAM on an n-processor c-collision crossbar, where k is some multiple of n. If we map k/n EREW PRAM processors to each processor of the crossbar, and employ a random hash function to map each location of the EREW PRAM shared memory to some memory module of the crossbar, we can easily see the connection between the random "balls and bins" experiment stated at the outset and the desired emulation: Each of the at most k read or write requests generated in a single step of the EREW PRAM computation corresponds to a ball, and each memory module corresponds to a bin.

If $k = n$ and $c = O(1)$, we can conclude that any scheme based on a single hash function requires $\Omega(\lg n/\lg\lg n)$ time to emulate one step of the EREW PRAM. On the other hand, Dietzfelbinger and Meyer auf der Heide [6] have recently shown that a bound of $O(\lg\lg n)$ time per EREW PRAM step is attainable for the same settings of k and c. They present a contention resolution protocol that minimizes the effect of the inevitable "hot-spot" memory modules (e.g., those memory modules receiving $\Theta(\lg n/\lg\lg n)$ requests under a given hash function) by employing three different hash functions. Thus, at the expense of increasing the total storage requirement by a factor of 3, the running time of the emulation is exponentially decreased. The fast performance of the Dietzfelbinger and Meyer auf der Heide emulation relies on a sharp threshold phenomenon that is the focus of Section 2. An overview of our results in this area is given in Section 1.1.

If $k = n\lg n$ and $c = O(1)$, the second "balls in bins" claim made at the outset of the paper shows that the emulation of a single EREW PRAM step will correspond to a $\Theta(\lg n)$-relation routing problem wvhp. (An *h-relation* routing problem is one in which each processor is the source of at most h packets and each memory module is the destination of at most h packets.) Thus, a variety of authors have considered the complexity of h-relation routing on the 1-collision crossbar. We are particularly interested in a natural randomized h-relation routing algorithm proposed by Geréb-Graus and Tsantilas [7]. The idea of their algorithm is to use randomization to break the symmetry between sets of processors with packets to be sent to the same same memory module: In a given "round" a processor with p packets left to send attempts to send a randomly chosen packet with probability roughly p/h, and does nothing with probability roughly $1-p/h$. The resulting running time is $\Theta(h + \lg n \lg h)$. (Remark: the running time is stated as $\Theta(h + \lg n \lg\lg n)$ in [7], since they only considered the case $h \geq \log n$.) Thus, the algorithm leads to a work-optimal EREW PRAM emulation only for $h = \Omega(\lg n \lg\lg n)$.

The symmetry-breaking idea employed by Geréb-Graus and Tsantilas is central to many other randomized algorithms and communication protocols (e.g., the standard Ethernet protocol [17] and the classic ALOHA packet radio network protocol [1]). As we have seen, such randomized symmetry-breaking does not always lead to performance that is obviously optimal (i.e., that matches a trivial lower bound). (For example, we might have hoped that the algorithm of Geréb-Graus and Tsantilas would run in $\Theta(\lg n)$ time for the case $h = \Theta(\lg n)$.) The main result of the Section 3 is a tight lower bound on the running time of certain randomized symmetry-breaking procedures. In particular, with respect to the problem of randomized h-relation routing on the 1-collision crossbar, we have completely characterized the power of the natural symmetry-breaking paradigm. An overview of our results in this area is given in Section 1.2.

Due to space limitations, most of the proofs have been omitted from this version of the paper. Complete proofs may be found in [15].

1.1 Redundancy-Based Protocols

In this section we give a brief overview of our emulation results related to protocols employing multiple hash functions. As mentioned earlier, Dietzfelbinger and Meyer auf der Heide [6] have presented a protocol using 3 hash functions that emulates an n-processor EREW PRAM in $O(\lg\lg n)$ time on an n-processor c-collision crossbar. Under their protocol, a read or write operation of memory location x by EREW PRAM processor i is emulated by having processor i of the c-collision crossbar access 2 out of the 3 copies corresponding to memory location x. (A similar protocol was presented in [13]; the idea that accessing 2 out of 3 copies is sufficient for the purposes of such an emulation was first used by Upfal and Wigderson [19].) The analysis presented in [6] requires some slack in the constants; in particular, they require $c \geq 3$, and are only able to analyze the protocol when it is used to emulate εn processors at a time, where ε is a sufficiently small positive constant. (Thus, the overall running time of the protocol is increased by a factor of $1/\varepsilon$.)

The protocol of [6] is easily generalized to the case where b hash functions are used, and each processor of the c-collision crossbar is required to access a out of b copies of a particular memory location, $a < b$. In Section 2.2, we focus on the case $a = 1$, and pinpoint the asymptotic complexity of the resulting protocol for all possible choices of the parameters b and c. (Furthermore, our analysis goes through with $\varepsilon = 1$, that is, we consider the most basic form of the protocol in which the action of all n EREW PRAM processors is emulated at once.) For $c = 1$, we prove that the protocol runs in $\Theta(\lg\lg n)$ time whp if $b \geq 3$. For $c = 1$ and $b = 2$, we prove that the protocol runs in $\Omega(\lg n)$ time wvhp. For any $b \geq 2$ and any constant $c \geq 2$, we prove that the protocol runs in $\Theta(\lg\lg n)$ time whp. (The protocol will run faster for non-constant c. It would not be difficult to extend our analysis to obtain tight bounds for non-constant c.) In the case of a c-arbitrary crossbar, the protocol runs in $\Theta(\lg\lg n)$ time whp for $b \geq 2$.

In Section 2.3, we observe that any "a out of b" problem with $a > 1$ can be efficiently reduced to a number of "1 out of ℓ" problems, where $\ell = 2$ or $\ell = 3$. Thus, we are able to easily upper bound the complexity of a (new) protocol for essentially any "a out of b" problem. One might suspect that a reduction of this sort, while making the analysis easier, is only doing so at the expense of a significant constant factor in performance. Interestingly, this is not the case; rather, as discussed in Section 2.3, our reduction yields a faster "a out of b" protocol than is obtained via the natural generalization of [6] for virtually all possible values of a and b.

The main idea underlying the aforementioned results may be explained as follows. While the process of throwing k balls independently and uniformly at random into n bins is well understood (e.g., we can easily compute a sharp bound on the number of bins receiving exactly one ball), we find that the sequence of distributions arising in the analysis of any "a out of b" protocol quickly deviates from such simple behavior. In [6, 18], this problem is attacked by studying certain structures related to the sequence of distributions. We are able to provide a more accurate analysis by precisely characterizing this sequence of distributions in terms of "truncated k balls in n bins" distributions (e.g., throw k balls into n bins and then remove all balls contained in bins with less than or equal to c balls).

1.2 Symmetry-Breaking Protocols

In this section we give a brief overview of our results on lower bounds for symmetry-breaking protocols in multiple access channels and for h-relation routing on a 1-collision crossbar.

There has been considerable effort in proving bounds on symmetry-breaking protocols to resolve contention in Ethernet-like multiple access channels [11, 12, 14, 16]. Specifically, it is assumed that some h of n stations wish to transmit to a single shared channel, but a station succeeds in its transmission if and only if it is the only station transmitting at that time. A symmetry-breaking protocol generates a schedule of transmission attempts for each station so that all k stations eventually transmit successfully. Previously studied protocols assume that all stations receive a feedback of 0, 1, or ≥ 2 at each step, depending on how many stations attempt to transmit. We call this the *Ethernet model.* For the Ethernet model, a lower bound of $\Omega(h+(h/\log h)\log n)$ was shown for the time of any deterministic protocol [12], and it was shown that an $O(h\log n)$ time (non-adaptive) deterministic protocol exists [14]. We study protocols in which only the transmitting stations receive feedback (1 or ≥ 2) at a given step. We call the general contention resolution problem on this model the *Control Tower problem.* Protocols to solve the Control Tower problem correspond to non-adaptive protocols for contention resolution on the Ethernet model. We show a lower bound of $\Omega(h + \log h \log n)$ for randomized protocols for the Control Tower problem that succeed with probability at least $1 - n^{-3/4}$. (Naturally, this provides the same lower bound for non-adaptive randomized protocols in the Ethernet model.) From a technical standpoint, it is most natural to view the Control Tower problem as a problem on hypergraphs. Our lower bound relies on a combinatorial argument for extracting "thick" hypergraphs, and a probabilistic argument showing a non-trivial probability of the existence of contention-generating "near transversals" in random sets of vertices.

The lower bound shown for the Control Tower problem can be used to prove a tight lower bound of $\Omega(h + \log h \log n)$ for direct randomized h-relation routing on a 1-collision crossbar. (In a *direct* algorithm for a given routing problem, the messages to be routed can only be sent directly from the source to the destination without any intermediate hops, and no additional information can be sent between the processors.) Direct algorithms have the advantage of simplicity and low overhead. While non-direct algorithms may have better asymptotic behavior, it is likely that this improved asymptotic behavior is only achieved at the expense of large constant factors. The previously mentioned h-relation routing algorithm of Geréb-Graus and Tsantilas [7] is a direct algorithm, and direct h-relation routing algorithms have also been studied in [8, 9]. We refer the reader to these previous papers for further details. (For previous work on non-direct h-relation routing on the OCPC, see [4, 8, 9, 20]; for recent work that also incorporates redundancy-based techniques, see [10].)

2 Multiple Hash Functions

In this section, we address the a out of b problem discussed in Section 1 on c-arbitrary and c-collision crossbars. In an *a out of b* problem on an n-processor crossbar, each shared memory cell is randomly and independently hashed b times into the memory modules of the crossbar, and each processor has to access a out of the b copies of a particular shared memory cell.

Consider the 1 out of ℓ problem, $\ell \geq 2$. Let the ℓ hash functions be labeled h_i, $0 \leq i < \ell$, and the shared memory request of processor j be for cell x_j. So processor j needs to access one of the memory locations $h_i(x_j)$, $0 \leq i < \ell$. To solve this problem, the following simple sequence of ℓ rounds can be repeated until each processor has had one successful access: In the jth round, if processor i has not successfully accessed any copy of x_i, then processor i accesses $h_j(x_i)$. (This is analogous to Access Schedule 2 of [6], defined for the 2 out of 3 problem.) On a c-collision crossbar, processor j succeeds on its access if and only if there are at most $c-1$ other processors accessing the same memory module. Each round is executed in a synchronous fashion. We refer to this protocol as the 1 out of ℓ protocol.

We analyze the above process in an equivalent balls-and-bins setup. Let n balls labeled 0 through $n-1$ represent the messages, and n bins labeled 0 through $n-1$ represent the destinations. Each hash function, a random function from $[n]$ to $[n]$, is equivalent to a random throw of n balls uniformly and independently into n bins. Let h^A denote the function h with domain restricted to the set $A \subseteq [n]$. Let R_i denote the set of balls remaining after round i. For convenience, define R_{-1} to be the set of balls left before round 0, i.e., $R_{-1} = [n]$. Note that for $i \geq 0$, R_i is the subset of R_{i-1} given by the following recurrence:

$$R_i = \{j \in R_{i-1} : |f^{-1}(f(j))| > c\},$$

where $f = h^{R_{i-1}}_{i \bmod \ell}$.

Recall that a bag (or multiset) is an unordered set in which repetition is allowed. For any set A we define a bag B to be an *A-bag* if every element of B is also an element of A.

Let $\mathcal{F}_{m,n}$ denote the set of functions from $[m]$ to $[n]$. For each $f \in \mathcal{F}_{m,n}$, note that the bag $\{f(j) : j \in [m]\}$ is an m-size $[n]$-bag. Hence, the uniform distribution over $\mathcal{F}_{m,n}$ induces a probability distribution, which we denote $\mathcal{D}_{m,n}$, over the set of all m-size $[n]$-bags. For any bag B and $A \subseteq [n]$, let $\mathcal{B}_{A,B} = \{f^A : f \in \mathcal{F}_{n,n} \text{ and } f(A) = B\}$. Let S_i and T_i denote the bags $h_{i \bmod \ell}(R_{i-1})$ and $h_{i \bmod \ell}(R_i)$, respectively. Let $t_i = |T_i| = |R_i|$ (thus $t_{-1} = n$) and $s_i = |S_i|$. Note that $\langle t_i \rangle$ is a nonincreasing sequence. The protocol terminates after the first round i for which $t_i = 0$. The protocol fails to terminate if and only if $t_i = t_{i+\ell} > 0$ for some $i \geq -1$. (In such a case, the protocol enters an infinite loop with $t_j = t_i$ for all $j \geq i$.) The goal of our analysis is twofold: (i) to bound the probability that the protocol fails to terminate, and (ii) to analyze the number of rounds required by the protocol when it does terminate. We begin our analysis by establishing some properties of $\mathcal{D}_{m,n}$ and $\mathcal{B}_{A,B}$.

Let random variable X be drawn from $\mathcal{D}_{m,n}$, B be an arbitrary $[n]$-bag of size m, and m_i denote the number of copies of element i in B, $0 \leq i < n$. Then

$$\Pr[X = B] = \frac{m!}{m_0! \cdots m_{n-1}!} \cdot \frac{1}{n^m}. \tag{1}$$

Lemma 2.1 Let a, m, and n be integers such that $0 \leq t \leq m \leq n$. Let X be a random variable drawn from $\mathcal{D}_{m,n}$. Let $Y \mid X$ be a random a-size subbag of X. Then Y is a random variable with probability distribution $\mathcal{D}_{a,n}$. ■

Lemma 2.2 Let $h \mid R, B$ be a function drawn uniformly at random from $\mathcal{B}_{R,B}$. For arbitrary $A \subseteq R$, $h(A) \mid R, B$ is a random $|A|$-size subbag of B. ■

Using Lemmas 2.1 and 2.2, we prove the following claims related to the 1 out of ℓ protocol.

Lemma 2.3 For all $i \geq 0$, the random variable $h^{R_i}_{i \bmod \ell} \mid R_i, T_i$ is drawn uniformly at random from $\mathcal{B}_{R_i,T_i}$. ■

Lemma 2.4 Let S'_i denote the random variable $S_i \mid \{(R_j, T_j) : 0 \leq j < i\}$. For $0 \leq i < \ell$, S'_i is drawn from $\mathcal{D}_{t_{i-1},n}$ and for $i > \ell$, S'_i is a t_{i-1}-size random subbag of $T_{i-\ell}$. ■

Now we are ready to describe the protocol in terms of the S_i's and T_i's alone. Let $RandomBag(m,n)$ return a bag drawn from $\mathcal{D}_{m,n}$. Let $RandomSubbag(B,m)$ return a new bag that is a random m-size subbag of B. Let $PrunedBag(B,c)$ return a bag that contains exactly those elements of S that have more than c copies. By Lemma 2.4, **Alg1**(n,ℓ,c) describes the random process occuring in the 1-out-of-ℓ protocol on a c-collision n-processor crossbar.

Alg1(n, ℓ, c)
  $i := 0$;
  repeat
    if $i < \ell$ **then**
      $S_i := RandomBag(|T_{i-1}|, n)$
    else
      $S_i := RandomSubbag(T_{i-\ell}, |T_{i-1}|)$;
    $T_i := PrunedBag(S_i, c)$;
    $i := i + 1$
  until $|T_{i-1}| = 0$

In order to analyze **Alg1** we will estimate the size of T_i after round i. We propose a modified version of the above algorithm that simplifies the estimation of $|T_i|$. Observe that for $0 \le i < \ell$, S_i is the bag obtained by throwing $|S_i|$ balls at random into n bins, and T_i is $PrunedBag(S_i, c)$. Below we present the modified algorithm **Alg2**(n, ℓ, c) that approximately maintains this invariant after *every* round, under a suitable redefinition of S_i. The analysis in Section 2.2 will make this precise. **Alg2** is the same as **Alg1** except that Lines (1.5) and (1.6) are replaced by Lines (2.1) through (2.7), stated below.

(2.1) **else** {
(2.2) $\quad S_i, T_i := S_{i-l}, T_{i-\ell}$;
(2.3) $\quad$ **while** $|T_i| > |T_{i-1}|$ {
(2.4) $\quad\quad$ "Select x at random from S_i";
(2.5) $\quad\quad S_i, T_i := S_i \setminus \{x\}, T_i \setminus \{x\}$
(2.6) $\quad$ }
(2.7) };

Since each element x in line (2.4) is selected at random from S_i, any element selected from T_i is also random in T_i. Moreover exactly $|T_{i-1}|$ of the elements from $T_{i-\ell}$ are retained after the execution of the **while** loop.

Lemma 2.5 Let S_i^1, T_i^1 (resp., S_i^2, T_i^2) denote bags S_i, T_i in **Alg1** (resp., **Alg2**) after round i, $i \ge 0$. Then T_i^1 and T_i^2 have the same probability distribution. ■

In the remainder of this section, we analyze **Alg2** under different assignments to the parameters ℓ and c. In Subsection 2.1 we analyze certain "balls and bins" experiments. Subsection 2.2 uses these results to analyze **Alg2**$(n, \ell, 1)$. Among other results, we show that **Alg2**$(n, 3, 1)$ terminates in $\Theta(\log\log n)$ rounds whp. The analysis can be easily generalized to apply to **Alg2**(n, ℓ, c) for other values of ℓ and c. Subsection 2.3 presents simple ideas for extending the above results to general a out of b problems.

2.1 Lemmas on Balls and Bins

In **Alg2**$(n, \ell, 1)$, T_i is that subbag of S_i, each element of which has at least 2 copies. We call such elements *non-singletons*. In Section 2.2, we show that the probability distribution of S_i is approximately $\mathcal{D}_{s_i,n}$. Thus, t_i is approximately the number of non-singletons in a random bag drawn from $\mathcal{D}_{s_i,n}$. In this section, we estimate the number of non-singletons in a random bag drawn from $\mathcal{D}_{m,n}$. By linearity of expectation, the expected number of non-singletons in a random bag S drawn from $\mathcal{D}_{m,n}$ is given by $f(m, n)$, where

$$f(m,n) \quad = \quad m\left(1-\left(1-\frac{1}{n}\right)^{m-1}\right).$$

Throughout this section n will be fixed, so we use $f(m)$ to denote $f(m, n)$. It is easy to verify that for $3 \le m \le n$, we have $m^2/(3n) \le f(m) \le m^2/3n$. Let

$$\begin{aligned}\delta &= 1 - 1/\log^3 n, \text{ and}\\ \Delta &= 1 + 1/\log^3 n.\end{aligned}$$

Now we bound the probability that the number of non-singletons in a random bag drawn from $\mathcal{D}_{m,n}$ deviates from the mean $f(m)$. In Lemma 2.6, we use a martingale analysis [3] to bound deviations to within a $o(1)$ factor for m sufficiently large. Lemma 2.7 bounds deviations to within a constant factor for all m.

Lemma 2.6 Let m and n be integers such that $3 \le m \le n$ and S be a random bag drawn from $\mathcal{D}_{m,n}$, and t be the number of non-singletons in S. If $m \ge n^{2/3}\log^3 n$, then $\delta f(m) \le t \le \Delta f(m)$ wvhp. ■

Lemma 2.7 Let m and n be integers such that $3 \le m \le n$, and S be a random bag with probability distribution $\mathcal{D}_{m,n}$. Let t represent the number of non-singletons in X. Then,

1. For $\sqrt{n}\log^5 n \le m \le n$, we have $t \le 4m^2/n$ wvhp.
2. For $m \le \sqrt{n}\log^5 n$, we have $t \le 4\log^{10} n$ wvhp.
3. For $m \le n^{1/2-\beta}$, where β is real, $0 < \beta \le 1/2$, t is zero with probability at least $1 - 1/n^{2\beta}$.

■

2.2 Analysis of Alg2

In this section, we analyze the number of rounds **Alg2** takes before termination. For $0 \le i < \ell$, we have $s_i = t_{i-1}$. Lemmas 2.8 and 2.9 establish bounds on s_i in terms of the s_j''s, $0 \le j < i$. Lemma 2.10 relates t_i to s_i for $i \ge \ell$.

Lemma 2.8 In **Alg2**(n, ℓ, c), if $i \ge \ell$, $s_{i-\ell}t_{i-1}/t_{i-\ell} \ge 2n^{2/3}\log^3 n$ and $t_{i-\ell} \ge s_{i-\ell}^2/4n$, then wvhp,

$$\delta s_{i-\ell}t_{i-1}/t_{i-\ell} \le s_i \le \Delta s_{i-\ell}t_{i-1}/t_{i-\ell}.$$

■

Lemma 2.9 Let $i \ge \ell$. In **Alg2**(n, ℓ, c), if $t_{i-1} \ge \log^2 n$, then $s_i \le 3s_{i-\ell}t_{i-1}/t_{i-\ell}$ wvhp. If $t_{i-1} \le \log^2 n$, then $s_i \le 3s_{i-\ell}(\log^2 n)/t_{i-\ell}$ wvhp. ■

Lemma 2.10 In **Alg2**$(n, \ell, 1)$, let $i \ge \ell$ and $d, u \ge 0$ be integers such that $d \le s_i \le u$ wvhp.

1. If $d, u \ge n^{2/3}\log^3 n$, then $\delta f(d) \le t_i \le \Delta f(u)$ wvhp.
2. If $u \ge \sqrt{n}\log^5 n$, then $t_i \le 4u^2/n$ wvhp.
3. If $u \le \sqrt{n}\log^5 n$, then $t_i \le 4\log^{10} n$ wvhp.

4. If $u \le n^{1/2-\beta}$ for some real β, $0 < \beta \le 1/2$, then t_i is zero whp.

■

Using the above results, we show that the probability that **Alg2** deviates significantly from the "expected" behavior is polynomially small. Let s'_i be defined as follows:

$$s'_i = \begin{cases} n & \text{if } i = 0, \\ f(s'_{i-1}) & \text{if } 0 < i < \ell, \text{ and} \\ s'_{i-\ell}\frac{f(s'_{i-1})}{f(s'_{i-\ell})} & \text{otherwise.} \end{cases}$$

Let $t'_i = f(s'_i)$ for all $i \ge 0$. (Note that for all $i \ge 0$, s'_i is the expected value of s_i given that $(s_j, t_j) = (s'_j, t'_j)$ for $0 \le j < i$ and t'_i is the expected value of t_i given that $(s_j, t_j) = (s'_j, t'_j)$ for $0 \le j < i$ and $s_i = s'_i$.)

Lemma 2.11 In $\mathbf{Alg2}(n, \ell, 1)$, for all $0 \le i \le \frac{5}{2}\log_3 \log n$, if $s'_i \ge 4n^{2/3}\log^3 n$ and n is sufficiently large, then wvhp,

$$\delta^{3^i} s'_i \le s_i \le \Delta^{3^i} s'_i, \text{ and} \tag{2}$$
$$\delta^{2\cdot 3^i+1} t'_i \le t_i \le \Delta^{2\cdot 3^i+1} t'_i. \tag{3}$$

■

By Lemma 2.11, we can analyze **Alg2** by studying how s'_i decreases as i increases.

Lemma 2.12 For all $1 \le i < \ell$, if s'_{i-1} and n are sufficiently large, then

$$\frac{1}{3^{i-1}} \prod_{0 \le j < i} \frac{s'_j}{n} \le \frac{s'_i}{n} \le \prod_{i \le j < i} \frac{s'_j}{n}.$$

For $i \ge \ell$, if s'_{i-1} and n are sufficiently large, then

$$\frac{1}{3^{\ell-1}} \prod_{i-\ell+1 \le j < i} \frac{s'_j}{n} \le \frac{s'_i}{n} \le \prod_{i-\ell+1 \le j < i} \frac{s'_j}{n}.$$

■

Lemma 2.12 can be used to analyze $\mathbf{Alg2}(n, \ell, 1)$ for any $\ell \ge 2$. In the remainder of this subsection, we restrict our attention to the case $\ell = 3$. Let $w_i = \log_r(n/s_i)$ and $w'_i = \log_r(n/s'_i)$, where $r = n/f(n)$. (Note that $r \ge e/(e-1)$ for all $n \ge 2$.)

Lemma 2.13 In $\mathbf{Alg2}(n, 3, 1)$, for all $i > 0$, if $s'_{i-1} \ge 3$, then

$$w'_{i-2} + w'_{i-1} \le w'_i \le w'_{i-2} + w'_{i-1} + 2\log_r 3.$$

■

Lemma 2.14 In $\mathbf{Alg2}(n, 3, 1)$ for all $i > 0$, if $s'_{i-1} \ge 3$, then $p_1^{i-1} \le w'_i \le p_2^{i-1}$, where $p_1, p_2 > 1$ satisfy

$$p_1^2 - p_1 - 1 \le 0, \text{ and} \tag{4}$$
$$p_2^2 - p_2 - 2\log_r 3 - 1 \ge 0. \tag{5}$$

■

Theorem 1 $\mathbf{Alg2}(n, 3, 1)$ terminates in $O(\log\log n)$ rounds whp.

Proof: Let $\phi = (1+\sqrt{5})/2$. Since $\phi^2 - \phi - 1 = 0$,Lemma 2.14 implies that $w'_i \ge \phi^{i-1}$ for all $i > 0$. Let $k = \min\{i : w'_i \ge \log_r(\frac{n^{1/3}}{4\log^3 n})\}$. For $i = \lceil \log_\phi \log_r \frac{n^{1/3}}{4\log^3 n}\rceil + 1$, $w'_i \ge \log_r(\frac{n^{1/3}}{4\log^3 n})$. Therefore, $k \le \log_\phi \log_r \frac{n^{1/3}}{4\log^2 n} + 2$. (Also note that since $w'_2 \le 1 + 2\log_r 3$, $k \ge 3$ for n sufficiently large.) Since $\phi^{5/2} > 3$, $k \le 5/2\log_3 \log n$ for n sufficiently large. Thus, Equations 2 and 3 of Lemma 2.11 hold for all $i < k$. (Also note that $s'_k = n/r^{w'_k} \le 4n^{2/3}\log^3 n$.)

By Lemma 2.11, $t_{k-1} \ge \delta^{2\cdot 3^{k-1}+1} t'_{k-1}$ wvhp. Since $t'_{k-1} = f(s'_{k-1}) \ge (s'_{k-1})^2/3n$, $t_{k-1} \ge 16\delta^{2\cdot 3^{k-1}+1}(n^{1/3}\log^6 n)/3 \ge \log^2 n$ wvhp for n sufficiently large. By Lemma 2.9, $s_k \le 3s_{k-3}t_{k-1}/t_{k-3}$ wvhp. Substituting appropriate bounds on s_{k-3}, t_{k-3}, and t_{k-1} from Lemma 2.11, we have wvhp

$$s_k \le 3\Delta^{2\cdot 3^{k-1}+5\cdot 3^{k-3}+4} s'_k \le 3\Delta^{3^k} s'_k \le 4s'_k. \tag{6}$$

The last step follows from the fact that $\Delta^{3^k} < \alpha$ for any real $\alpha < 1$ and for n sufficiently large. We consider three different cases depending on the value of s'_k.

Case 1: $s'_k \le n^{1/2-\beta}$ for some $0 < \beta \le 1/2$. By Equation 6, $s_k \le 4n^{1/2-\beta}$. By Part 4 of Lemma 2.10, we conclude that t_k is zero whp.

Case 2: $s'_k \le \sqrt{n}\log^5 n$. By Equation 6, $s_k \le 4\sqrt{n}\log^5 n$ wvhp. Therefore, by Part 3 of Lemma 2.10, $t_k \le 64\log^{10} n$ wvhp. We consider two cases. If $t_k \ge \log^2 n$, by Lemma 2.9, $s_{k+1} \le 3s_{k-2}t_k/t_{k-2}$ wvhp. If $t_k \le \log^2 n$, then $s_{k+1} \le 3s_{k-2}\log^2 n/t_{k-2}$. In any case, $s_{k+1} \le (192 s_{k-2}\log^{10} n)/t_{k-2}$ wvhp. Now we substitute appropriate bounds on s_{k-2} and t_{k-2} from Lemma 2.11 and obtain wvhp,

$$\begin{aligned} s_{k+1} &\le \frac{192\Delta^{3^{k-2}} s'_{k-2}\log^{10} n}{\delta^{2\cdot 3^{k-2}+1} t'_{k-2}} \\ &\le \frac{576 n\Delta^{5\cdot 3^{k-2}+2}\log^{10} n}{s'_{k-2}} \\ &\le 144\Delta^{3^k} n^{1/3}\log^7 n, \end{aligned}$$

for n sufficiently large. (Note: The last step follows from the fact that $3^k \ge 5\cdot 3^{k-2}+2$ and $s'_{k-2} \ge 4n^{2/3}\log^3 n$.) Applying Part 4 of Lemma 2.10, t_{k+1} is zero whp.

Case 3: $s'_k \geq \sqrt{n}\log^5 n$. By Equation 6, $s_k \leq 4s'_k$ wvhp. We again consider two cases, depending on whether $t_k \leq \log^2 n$ or $t_k \geq \log^2 n$.

If $t_k \leq \log^2 n$ then Lemma 2.9 implies that $s_{k+1} \leq 3s_{k-2}\log^2 n/t_{k-2}$ wvhp. Arguing as in Case 2, t_{k+1} is zero whp.

If $t_k \geq \log^2 n$ then Lemma 2.9 implies that $s_{k+1} \leq 3s_{k-2}t_k/t_{k-2}$ wvhp. Since $s_k \leq 4s'_k$, by Part 2 of Lemma 2.10, $t_k \leq 64(s'_k)^2/n \leq 192t'_k$ wvhp. Substituting this upper bound on t_k and appropriate bounds on s_{k-2} and t_{k-2} obtained from Lemma 2.11, we have $s_{k+1} \leq 1000s'_{k+1}$ wvhp for n sufficiently large. We now derive an upper bound on s'_{k+1}.

By Lemma 2.13, $w'_k \leq w'_{k-1} + w'_{k-2} + 2\log_r 3$. Since $w'_{k-1} \geq w'_{k-2}$, we have

$$w'_{k-1} \geq \frac{1}{2}(w'_k - 2\log_r 3) \geq \frac{\log_r n}{6} - \frac{3\log_r \log n}{2} - \log_r 6.$$

Thus, by Lemma 2.13,

$$\begin{aligned} w'_{k+1} &\geq w'_k + w'_{k-1} \\ &\geq \frac{\log_r n}{2} - \frac{9\log_r \log n}{2} - \log_r 24, \text{ and} \\ s'_{k+1} &\leq \sqrt{n}\log^5 n, \end{aligned}$$

for n sufficiently large. Now we apply an analysis similar to Case 2 with k replaced by $k+1$ to establish that t_{k+2} is zero whp. Cases 1, 2, and 3 show that whp, **Alg2** terminates in at most $k+2$ rounds. Since $k = O(\log\log n)$, the desired result follows. ■

Theorem 2 **Alg2**$(n,3,1)$ terminates in $\Omega(\log\log n)$ rounds wvhp.

Proof: A solution to Equation 5 is $p_2 = 3$. Thus Lemma 2.14 implies that $w'_i \leq 3^{i-1}$ for all $i > 0$. After $k = \lfloor \log_3((\log_r n)/4)\rfloor$ rounds, $w'_k \leq (\log_r n)/4$, and $s'_k \geq n^{3/4}$. For n sufficiently large, $n^{3/4} \geq 4n^{2/3}\log^3 n$. Therefore, by Lemma 2.11, we have $t_k \geq \delta^{c^k} t'_k \geq \delta^{c^k}(s'_k)^2/3n \geq n^{1/3}$ wvhp for n sufficiently large. Thus, **Alg2**$(n,3,1)$ executes at least $(\log_3 \log_r n)/4 = \Omega(\log\log n)$ rounds wvhp. ■

The recurrence in Lemma 2.12 for $\ell = 2$ yields $s'_{i+1}/n \geq s'_i/3n$ for all $i \geq 0$. Thus $w'_i = O(i)$. Using an analysis similar to the Theorem 4 we establish an $\Omega(\log n)$ lower bound for **Alg2**$(n,2,1)$.

Theorem 3 **Alg2**$(n,2,1)$ terminates in $\Omega(\log n)$ rounds wvhp. ■

For a 2-collision crossbar, we can do a similar analysis and establish the following result.

Theorem 4 For $\ell \geq 2$, **Alg2**$(n,\ell,2)$ terminates in $\Theta(\log\log n)$ rounds whp. ■

2.3 Generalizations

Alg1 can be generalized to apply to any a out of b problem by changing the *RandomSubbag* and *PrunedBag* routines appropriately; after each step, we need to keep track of how many successes each processor has had, and only those processors with fewer than a successes participate. In the following discussion, we refer to this protocol as the *generic* protocol. For given a and b, the analysis of the generic protocol can be done using the approach of Subsection 2.2, but involves more complicated calculations and recurrences. In [6], the authors use a different analysis of this protocol for the 2 out of 3 case and show an $O(\log\log n)$ upper bound when the collision factor is greater than 3. In this section, we present a simple variant of the generic protocol that solves any a out of b problem on a 2-collision crossbar in $O(\log\log n)$ time whp.

In particular we can solve any a out of $a+1$ problem by running **Alg1**$(n,2,1)$ with $\binom{a+1}{2}$ different hash-function pairs. Since each run fails with a polynomially small probability, and there are only a constant number of runs, the entire algorithm succeeds whp. For instance, in the case of 2 out of 3, we simply perform 3 runs of **Alg1**. At first glance, it may appear that this revised protocol is only of interest because it is simpler to analyze. Actually, the new protocol is competitive with the generic one for small a and is much faster for large a. Comparing it for the 2 out of 3 problem, we first note that since each of the 3 runs use 2 hash functions only, while the generic protocol uses 3, the revised protocol will be at most twice as slow as the generic one. Moreover, the 1 out of 2 problem is clearly a simpler problem than the 2 out of 3 problem. So each run will involve a fewer rounds than in the generic algorithm. For large a, this phenomenon is exaggerated. For a generic a out of $a+1$ protocol to make "progress", a number of processors must have a large number of successes. But at the outset, the fraction of processors that have succeeded on $d \leq a$ hash functions decreases exponentially with d. Therefore, while the revised protocol experiences only a quadratic slowdown in running time, the generic procotol will suffer an exponential increase in running time with increasing a.

The basic idea outlined above can be used to solve any a out of b problem by choosing any $a+1$ hash functions and solving the corresponding a out of $a+1$ problem.

Theorem 5 For integer constants a and b with $1 \leq a < b$, the corresponding a out of b problem can be solved on a 2-collision crossbar in $O(\log\log n)$ time whp. ■

The above generalizations can be made for the 1-collision crossbar as well. Since **Alg1** solves the 1 out of 3 problem on a 1-collision crossbar in $O(\log\log n)$ time whp, any a out of $a+2$ problem can be solved in the same asymptotic time bound by running $\mathbf{Alg1}(n, 3, 1)$ on $\binom{a+2}{3}$ different triples of hash functions.

Theorem 6 For integer constants a and b with $1 \leq a < b-1$, the corresponding a out of b problem can be solved on a 1-collision crossbar in $O(\log\log n)$ time whp. ■

It is worth noting that by the above result, a 1-collision crossbar can solve the 3 out of 5 problem and hence can simulate an EREW PRAM with n processors in $O(\log\log n)$ time by using 5 hash functions. Thus a 1-collision crossbar is asymptotically as powerful as a 2-collision one.

3 Symmetry Breaking

In this section we analyze randomized algorithms for the Control Tower problem. In the Control Tower problem, there is one central control tower and n remote stations, h of which contain a message destined for the control tower. Each station can only transmit a message to and receive a message from the control tower, and only in discrete time slots. When two or more stations attempt to transmit a message in the same time slot, all of the transmitted messages are lost. Note that the control tower can transmit a message to only one remote station in one time slot. We will consider each time slot to be a *step*; the goal is to transmit all h messages to the control tower in as few steps as possible. (Throughout our analysis, we will assume that the remote stations are numbered from 1 to n, and that the control tower sends an acknowledgement upon receipt of a message.)

The Control Tower problem is related to the problem of direct routing of h-relations on a 1-collision crossbar, as discussed in Section 1. Specifically, since each processor is only permitted to transmit a message directly to its destination, we are able to analyze each destination independently. Transmitting a set of at most h messages to a destination is then equivalent to the Control Tower problem. We show how the lower bound we obtain for the Control Tower problem can be used to obtain a lower bound for direct h-relation routing.

To give some intuition into the Control Tower problem, let us first examine the case when $h = 2$. In this case, the problem is that two stations are trying to transmit messages, but if they transmit at the same time, then both of the transmissions are blocked. Note that some sort of symmetry breaking is required, or the messages may never be successfully transmitted. A simple randomized strategy to break the symmetry would be for each station to flip a coin and transmit if and only if it comes up "heads". Then the expected number of steps before both messages are successfully transmitted is constant. To achieve success wvhp requires $\Theta(\log n)$ steps, however. Still, this turns out to be the best strategy possible, as shown by Goldberg, Jerrum, Leighton, and Rao [9]. Generalizing to $h > 2$, we can easily verify that if there are k messages left to transmit ($k \leq h$), then by having each station transmit with probability $1/k$, we maximize the probability of obtaining a successful transmission. The difficulty is that the stations are not able to communicate with each other, and consequently they do not know exactly how many messages are left to transmit at any intermediate step of the algorithm. If the number of messages left to transmit were known to all processors at every step, then the Control Tower problem could be solved in $\Theta(h + \log n)$ steps wvhp. Even with this uncertainty, Geréb-Graus and Tsantilas [7] showed that the Control Tower problem could be solved in $O(h + \log h \log n)$ time wvhp, but it was not known whether the extra $\log h$ factor could be eliminated. We show that the extra factor of $\log h$ is indeed necessary.

The Control Tower problem can most easily be studied in the context of hypergraphs, and so here we review the definition of a hypergraph and related concepts.

Let V be a set of elements, called *vertices*. Let E be a set of nonempty subsets of V, called *edges*. Then a *hypergraph* is given by a pair (V, E). Given a hypergraph $H = (V, E)$, the hypergraph *induced* by a set of vertices $V' \subseteq V$ is $H' = (V', E')$, where $E' = \{e \in E : e \cap V' \neq \emptyset\}$. Also the hypergraph *induced* by a set of edges $E' \subseteq E$ is $H' = (V, E')$. A subset of vertices $T \subseteq V$ *covers* an edge $e \in E$ if $e \cap T \neq \emptyset$. A *transversal* of H is a set of vertices $T \subseteq V$ that covers every edge in E. For convenience, we define an *a-transversal* to be a set of vertices $T \subseteq V$ that covers at least a edges of E.

We define a hypergraph $H = (V, E)$ to be *a-thick* if $\min_{e\in E} |e| \geq a$, Note that if E is empty, then H is a-thick for every a. A hypergraph $H = (V, E)$ is (a, b)-*thick* if H is a-thick and $\sum_{k\geq 0} f_k 2^{-k} \leq b$, where f_k denotes the number of edges $e \in E$ such that $a2^k \leq |e| \leq a2^{k+1}$, $k \geq 0$.

3.1 Randomized Lower Bound

A randomized algorithm can solve the Control Tower problem in expected $O(h)$ steps. However, for many applications, including the analysis of direct h-relation routing on a c-collision crossbar, it is necessary to determine the number of steps required to achieve a polynomially small probability of failure. That is what we analyze in this section.

Our lower bound will make use of a result of Yao [21], which states that any lower bound for deterministic algorithms on random inputs implies the same lower bound for randomized algorithms on worst case inputs. In particular, we will proceed by developing a lower bound for deterministic algorithms solving the Control Tower problem, where we assume that the h stations that have messages to transmit are chosen randomly from the n stations.

To begin, we convert the problem into the hypergraph domain. Then we show how to reduce the hypergraph corresponding to a given deterministic algorithm to a thick hypergraph, and show that with a significant probability, the processors corresponding to vertices in this thick hypergraph contain messages that are not successfully transmitted.

The following lemma is useful for hypergraphs with a small number of edges.

Lemma 3.1 Any hypergraph $H = (V, E)$ with n vertices and m edges has a subset V' of V with $|V'| \geq n2^{-m}$ in which every vertex in V' covers exactly the same set of edges. ■

For hypergraphs with a larger number of edges, we will make use of the following lemma, that generalizes a result of Alon, Bar-Noy, Linial and Peleg [2, Lemma 3.1].

Lemma 3.2 Given $r_1, r_2 \geq 1$, where $r_1 < r_2$, let $r = r_2 - r_1$ and $H = (V, E)$ be a hypergraph with n vertices and m edges. Then for some k, $r_1 \leq k < r_2$, there is a subset V' of V with $|V'| \geq ne^{-8m2^{-k}/r}$ that induces a $(|V'|2^{-k}, 4m/r)$-thick hypergraph $H' = (V', E')$. ■

Next we show that large random sets of vertices in a thick hypergraph cover many edges with non-trivial probability.

Lemma 3.3 Let $x \geq 1$ and $t = \lceil 13x \rceil$. Given an $(n/x, y)$-thick hypergraph $H = (V, E)$ with n vertices and m edges, there are at least $\binom{n}{t}(2x)^{-t}$ subsets of vertices of size t that cover, for every $j \geq 0$, all but $y2^{-(j+1)}$ of the edges with thickness in the range $[n2^j/x, n2^{j+1}/x)$. ■

Using Lemma 3.3 we establish a lower bound on the probability that a random set of vertices contains two near transversals covering exactly the same set of edges.

Lemma 3.4 Let $x \geq 1$ and $t = \lceil 13x \rceil$. Given an $(n/x, y)$-thick hypergraph $H = (V, E)$ with n vertices and m edges, where $n \geq 4t$, the probability that two randomly chosen disjoint t-size subsets of V cover exactly the same set of edges is at least

$$\frac{1}{2(2x)^t}\left(\frac{1}{2y^{\log y}(2x)^t e^{18y}} - \frac{4t^2}{n}\right).$$

■

Now we can place a bound on the number of transmission steps required to solve the Control Tower problem with probability of failure polynomially small in n.

Theorem 7 Let $\mathcal{A}$ be a deterministic algorithm for the Control Tower problem where $h \geq 2$ messages are placed randomly at h of the n stations. If $\mathcal{A}$ succeeds in $T(n, h)$ steps with probability at least $1 - n^{-3/4}$, then for some constant $\varepsilon > 0$, $T(n, h) \geq \varepsilon \log h \log n$.

Proof: Let $\varepsilon = 10^{-5}$. For $h \geq \log n \log\log n$, the result is trivial, since $T(n, h) \geq h \geq 10^{-5} \log h \log n$. So assume $2 \leq h \leq \log n \log\log n$. Let V be the set of n stations, and E be a set of edges, where edge i contains each station that would attempt to transmit a message in step i, if it had a message to transmit. Let $m = |E|$, and assume that $m < 10^{-5} \log h \log n < 10^{-4} \log h \ln n$. In what follows, we show that with probability at least $n^{-3/4}$ there are two disjoint sets of stations that attempt to transmit at exactly the same time steps. In terms of hypergraphs, this is equivalent to showing that with probability at least $n^{-3/4}$, a random set of h vertices contains two disjoint sets of vertices that cover exactly the same set of edges.

If $h < 2^{1000}$, then $m \leq \frac{1}{10} \log n$. If $n < 2^{10}$, then $m = 0$, and the theorem holds trivially. Otherwise, by Lemma 3.1, we can find a group of $n^{9/10}$ vertices that cover exactly the same set of edges. The probability that two of the h randomly placed messages lie in this group is at least

$$n^{-1/10}((n^{9/10} - 1)/(n - 1)) \geq n^{-3/4}.$$

(Goldberg, Jerrum, Leighton, and Rao [9] prove a similar result.)

If $2^{1000} \leq h \leq \log n \log\log n \leq \log^2 n$, then $n \geq 2^{2^{500}}$ and $h \leq n^{1/10}$. If $m \leq \log h$ then $m \leq \frac{\log n}{10}$, and the argument in the previous paragraph holds; otherwise, from Lemma 3.2 with $r_1 = \frac{\log h}{10}$ and $r_2 = \frac{3 \log h}{10}$, there is a subset V' of V with

$$|V'| \geq ne^{-(40m)/(x \log h)},$$

that induces a $(|V'|/x, 20m/\log h)$-thick hypergraph, for some x, $h^{1/10} \leq x \leq h^{3/10}$. By our assumption that $m \leq 10^{-4} \log h \ln n$, we have

$$|V'| \geq ne^{-\log n/(250x)} = n^{1-(250x)^{-1}},$$

and the induced hypergraph is $(|V'|/x, (\ln n)/500)$-thick. Note that for $h \geq 2^{1000}$, we have $|V'| \geq 52x + 4$ and $h^{1/2}/5 \geq 2\lceil 13x \rceil$. Furthermore, with y set to $(\ln n)/500$ and t set to $\lceil 13x \rceil$, we have

$$\frac{1}{2(2x)^t}\left(\frac{1}{2y^{\log y}(2x)^t e^{18y}} - \frac{4t^2}{n}\right) \geq n^{-1/5}.$$

Hence, the probability that at least $2\lceil 13x \rceil$ of the stations with messages belong to V' is at least

$$\begin{aligned} \left(\frac{n^{1-(250x)^{-1}}}{2n}\right)^{2\lceil 13x \rceil} &\geq n^{-1/5} 2^{-2\lceil 13x \rceil} \\ &\geq n^{-2/5}. \end{aligned}$$

By Lemma 3.4, the probability that two disjoint $\lceil 13x \rceil$-size random subsets of these vertices each cover exactly the same set of edges is at least $n^{-1/5}$. Hence, the probability that two disjoint subsets of the randomly chosen set of h vertices covers exactly the same set of edges is at least $n^{-2/5}n^{-1/5} \geq n^{-3/4}$. ∎

Corollary 7.1 Let $\mathcal{A}$ be a randomized algorithm for the Control Tower problem with $h \geq 2$ messages and n processors. If $\mathcal{A}$ succeeds in $T(n,h)$ steps with probability at least $1 - n^{-3/4}$, then for some constant $\varepsilon > 0$, $T(n,h) \geq \varepsilon \log h \log n$. ∎

Corollary 7.2 Let $h \geq 2$, and let $\mathcal{A}$ be a randomized algorithm for direct routing of an h-relation on a 1-collision crossbar with n processors. If the expected time of $\mathcal{A}$ is $T(n,h)$ then for some constant $\varepsilon > 0$, $T(n,h) \geq \max\{h, \varepsilon \log h \log n\}$. ∎

4 Concluding Remarks

As mentioned earlier, complete proofs of all of the claims made in this paper can be found in [15].

In Section 2, we assumed that the hash functions were fully independent. In [15], we extend our results to the case of hash functions with limited independence.

We have obtained an $O((h \log h) \log n)$-step upper bound for direct deterministic h-relation routing [15], which is close to the $\Omega((h/\log h)\log n)$ lower bound of [12]. The previous best upper bound for direct deterministic h-relation routing was $O(h^2 \log n)$. This result is implicit in [8].

Acknowledgements

The authors would like to thank Noga Alon, Leslie Goldberg, Yossi Matias, Friedhelm Meyer auf der Heide, Aravind Srinivasan, Torsten Suel, and David Zuckerman for many helpful discussions.

References

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

[2] N. Alon, A. Bar-Noy, N. Linial, and D. Peleg. A lower bound for radio broadcast. *JCSS*, 43:290–298, 1991.

[3] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley-Interscience, New York, NY, 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 the 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 *Proceedings of the 5th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 110–119, June 1993.

[7] M. Geréb-Graus and T. Tsantilas. Efficient optical communication in parallel computers. In *Proceedings of the 4th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 41–48, June 1992.

[8] L. A. Goldberg and M. Jerrum. A sub-logarithmic communication algorithm for the completely connected optical communication parallel computer. Technical Report ECS–LFCS–92–234, Laboratory for Foundations of Computer Science, Department of Computer Science, University of Edinburgh, September 1992.

[9] 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 *Proceedings of the 5th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 300–309, June 1993.

[10] L. A. Goldberg, Y. Matias, and S. B. Rao. An optical simulation of shared memory. In *Proceedings of the 6th Annual ACM Symposium on Parallel Algorithms and Architectures*, June 1994. To appear.

[11] A. G. Greenberg, P. Flajolet, and R. E. Ladner. Estimating the multiplicities of conflicts to speed their resolution in multiple access channels. *Journal of the ACM*, 34:289–325, 1987.

[12] A. G. Greenberg and S. Winograd. A lower bound on the time needed in the worst case to resolve conflicts deterministically in multiple access channels. *Journal of the ACM*, 32:589–596, 1985.

[13] R. Karp, M. Luby, and F. Meyer auf der Heide. Efficient PRAM simulation on a distributed memory machine. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 318–326, May 1992.

[14] J. Komlós and A. G. Greenberg. An asymptotically nonadaptive algorithm for conflict resolution in multiple-access channels. *IEEE Transactions on Information Theory*, IT–31:302–306, 1985.

[15] P. D. MacKenzie, C. G. Plaxton, and R. Rajaraman. On contention resolution protocols and associated probabilistic phenomena. Technical Report TR–94–06, University of Texas at Austin, Department of Computer Science, April 1994.

[16] C. U. Martel and T. P. Vayda. The complexity of selection resolution, conflict resolution, and maximum finding on multiple access channels. In *Proceedings of the 3rd International Workshop on Parallel Computation and VLSI Theory*, pages 401–410, 1988.

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

[18] F. Meyer auf der Heide, C. Scheideler, and V. Stemann. Fast, simple dictionaries and shared memory simulations on distributed memory machines; upper and lower bounds. Manuscript, January 1994.

[19] E. Upfal and A. Wigderson. How to share memory in a distributed system. *Journal of the ACM*, 34:116–127, 1987.

[20] L. G. Valiant. General purpose parallel architectures. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, pages 943–971. Elsevier/MIT Press, 1990.

[21] A. C. Yao. Lower bounds by probabilistic arguments. In *Proceedings of the 24th Annual IEEE Symposium on Foundations of Computer Science*, pages 420–428, October 1983.