Available online at www.sciencedirect.com
SCIENCE DIRECT®
J. Parallel Distrib. Comput. 65 (2005) 1007–1021
Journal of Parallel and Distributed Computing
www.elsevier.com/locate/jpdc

A collision model for randomized routing in fat-tree networks

Volker Strumpen[a,*], Arvind Krishnamurthy[b]

[a]*Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, USA*
[b]*Department of Computer Science, Yale University, USA*

Received 24 February 2003; received in revised form 31 January 2005; accepted 11 April 2005
Available online 17 June 2005

Abstract

We present a proof that in a model of a fat-tree network with n processing nodes $m \leqslant n$ messages with randomly chosen, distinct sources and independently and randomly chosen destinations are delivered within $O(\lg m)$ delivery rounds with high probability. More succinctly, we establish that m messages are delivered in $O(\lg m + \ln 1/\varepsilon)$ delivery rounds with probability $1 - \varepsilon$ for any small $\varepsilon > 0$. Unlike previously applied proof methods, we use an approximating model for the collision behavior of the network amenable to concise yet simple theoretical analysis. We justify the accuracy of the approximation by means of behavioral simulations based on a gate-level implementation of a fat-tree network.

Keywords: Fat-tree network; Randomized routing; Collision model; Probabilistic analysis

1. Introduction

Fat-tree networks are established as area-universal communication networks due to the seminal work of Leiserson [13,4], culminating in the implementation of the Connection Machine CM-5 at Thinking Machines Corporation [14]. Today, advances in semiconductor technology enable us to integrate multiprocessor machines on a single chip, as explored in the Raw project [22], for example. As the number of processors on a chip increases, employing one or more fat-tree networks as interconnection medium presents an attractive design alternative.

The theoretical properties of fat-tree networks constitute a compelling reason to consider them for single-chip multiprocessors. In this article, we reevaluate the theoretical performance of a fat-tree network with respect to delivery times of messages. We present a proof that $m \leqslant n$ messages with randomly chosen, distinct sources and destinations can be delivered in a fat-tree network with n processing nodes within $O(\lg m)$ delivery rounds with high probability. Our result improves on previously published bounds based on the number of processing nodes n rather than the number of messages m. Leiserson [13] derived a bound using the *load factor* λ of a set of messages. The load factor of a set of messages is the largest ratio of the number of messages passing through one channel and the capacity of that channel considering all channels of the fat-tree network. Leiserson has shown that the number of delivery rounds required to deliver a set of messages, where the sources and destinations are known in advance, is $O(\lambda \lg n)$. Greenberg and Leiserson [4] have derived a bound $O(\lambda + \lg n \lg \lg n)$ for the number of delivery rounds when the sources and destinations of messages are unknown, assuming that the probability of congesting a channel follows the binomial distribution, however.

Empirical evidence shows that these bounds are conservative. To prove our tighter bound, we develop a model for the collision behavior of messages. Since this model merely approximates the actual occurrence of collisions, we present empirical evidence that it reflects reality accurately enough to justify our time bound. We have developed a gate-level implementation of the fat-tree network and a behavioral

* Corresponding author.
E-mail addresses: strumpen@csail.mit.edu (V. Strumpen), arvind@cs.yale.edu (A. Krishnamurthy).

0743-7315/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2005.04.005

simulator that permits us to scale our simulations up to large numbers of processing nodes. Our simulations show that $O(\lg m)$ is not only an upper bound for randomly chosen message sources and destinations, but for many regular communication patterns as well.

Our goal is to derive a suitable model of the collision behavior of a fat tree that approximates reality with sufficient accuracy and permits a concise yet simple theoretical analysis at the same time. Previous work, such as [4,1,6,8,11,12,15], suggests that an exact analysis requires significant theoretical armory. Our approach to the problem of network analysis is motivated by the simplicity of traditional approaches based on queueing theory [10,20]. Queueing theoretical models result in simple, algebraic equations for network analysis. However, these models are typically limited to statements about the average case behavior. In contrast, algorithmic analysis is typically applied to a routing algorithm for a particular network rather than an approximating model. Algorithmic techniques enable us to analyze the worst- and average-case behavior of the routing algorithm [2], or conduct an amortized analysis [21]. The results obtained by algorithmic analysis tend to be more insightful than queueing theoretical models. Yet, many routing algorithms require exceedingly complex mathematical treatment, and many have escaped rigorous analysis as of todate, cf. [11].

Probabilistic analyses of networks as well as algorithmic analyses of networks with probabilistic routing algorithms have been studied in the past. In general, these analyses are based on simplifying assumptions that constitute a *network model*. For example, [19,8,3,18,23,5,7] assume that a set of message sources of the network is chosen randomly, the source nodes transmit messages independently, destinations are selected independently and with respect to a uniform distribution, and collisions occur when multiple messages attempt to traverse one network wire at the same time. Resolution of collisions in circuit-switched networks is handled by dropping all but one of the colliding messages and sending a notification to the corresponding source nodes. The dependencies due to the resulting retransmissions are ignored. Empirical analysis suggests that these assumptions are justified indeed, yet we are not aware of a conclusive study that would assess the error margins incurred by such network models. We do not attack this problem either, but provide a reasonably simple analysis for a circuit-switched fat-tree network model based on similar assumptions.

2. Proof outline

Our proof is based on the structural analysis of a particular fat-tree network architecture, which results in the average probability $\Pr[C_2]$ of a collision of two messages with randomly chosen sources and destinations. This probability embodies the structure of the fat tree.

We then model the collision behavior of $m > 2$ messages by means of an approximating balls-and-bins game.

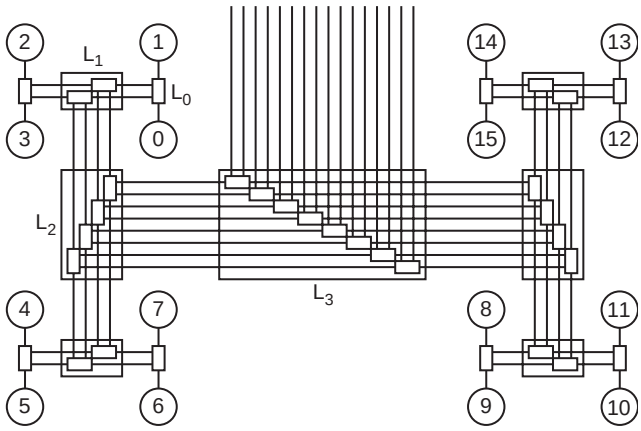

Fig. 1. Structure of fat-tree network with 16 processing nodes.

The simple balls-and-bins game neglects probabilistic dependencies. Nevertheless, in Section 5 we show empirically that neglecting dependencies due to the random selection of message destinations affects the result by a small constant factor only. We calibrate the number of collision bins to reflect the probability $\Pr[C_2]$. Messages correspond to balls tossed into collision bins. A message may be rejected or delivered depending on the outcome of the collision toss. Rejected messages must be retried, leading to a model of subsequent delivery rounds that correspond to a sequence of collision tosses.

We prove the result in two phases, depending on whether the number of messages m is larger than the number of collision bins b or not. We assume that all messages rejected in one delivery round are retried during the subsequent delivery round. In Phase I we prove that the *number* of messages delivered per delivery round for $m > b$ is larger than a constant amount with at least constant probability. In Phase II we prove that the *fraction* of messages delivered per delivery round for $m \leqslant b$ is larger than a constant amount with at least constant probability. In both phases, the expected number of delivery rounds is $O(\lg m)$. Finally, we use a Chernoff bound to establish the high-probability result for each phase that m messages are delivered within $O(\lg m + \ln 1/\varepsilon)$ delivery rounds with probability $1 - \varepsilon$ for any small $\varepsilon > 0$.

3. Fat-tree architecture

Our proof is restricted to the architecture of the fat-tree network shown in Fig. 1 with the router design described below. [1]

We introduce the following design decisions. The network shall be *circuit-switched*, where messages reserve a path from the source to the destination on their way through the

[1] The fat-tree network under investigation is similar, yet different from a *back-to-back butterfly* or *Beneš* network, because of its connections between the downstream ports.

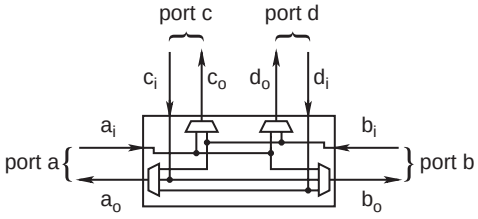

Fig. 2. Router design of a full-duplex fat tree.

network. In contrast to packet routing, this design is particularly suited for pipelining streams of data through an array of processors with register-mapped networks. Applications such as digital signal processing would be a primary beneficiary of this design choice. In a circuit-switched network, an explicit acknowledgment signal is used to release the resources of the reserved path. Consequently, no buffering is needed at the router nodes other than a small, constant number of pipeline registers. Furthermore, each processing node may have at most one message transmission and one, potentially simultaneous message reception in progress. Each of the links in Fig. 1 is a bidirectional link, or full-duplex link, consisting of two sets of wires, each responsible for transmitting signals in one direction. Each router of the network has four ports a, b, c, and d, and each port has an incoming and an outgoing set of wires, as shown in Fig. 2. We call ports a and b the *downstream ports*, and ports c and d the *upstream ports* as obvious from Fig. 1.

Each router is designed to transmit and reject messages according to the following behavior. *Upstream messages* arrive on one of the downstream ports a or b, and are transmitted through one of the upstream ports c or d at random. This upstream port selection is the only source of uniform randomness in the routing process, and embodies the benefits of having the choice among two alternatives [16]. If one upstream port is in use when a second upstream message arrives, the available port is assigned to the second message deterministically. *Downstream messages* are transmitted through one of the downstream ports a or b. Since the downstream ports have only one set of outgoing wires, a_o and b_o, respectively, contention may occur if more than one message shall be transmitted through one of these downstream ports. For example, if two downstream messages arrive on ports c and d, and both shall be transmitted through port a, only one of them may use wire a_o. The other message will be rejected, that is a collision signal will be sent to the sender for notification. The sender is responsible for initiating a retry.

The collision behavior of our router design obeys the simple *message rejection rule*: all but one of the downstream messages with the same outgoing port are rejected. Messages can collide only while traveling downstream. There exist two characteristic collision scenarios. (1) Two downstream messages arrive at the upstream ports to be transmitted through the same downstream port. (2) One downstream message arrives at one of the downstream ports, another downstream message at one of the upstream ports, and both shall be transmitted through the same downstream port. In both scenarios, one message is rejected, and the other passes successfully. If both scenarios happen simultaneously, that is three downstream messages arrive on one downstream port and both upstream ports, then two messages are rejected, and one passes through the router.

Noteworthy is that messages cannot collide while traveling upstream, because the network architecture doubles the amount of wires at each level of router nodes from the leaves towards the root. Therefore, we do not have to be concerned about contention on the upstream paths of messages, even if each processing node injects a message into the tree.

We introduce the following naming scheme for the network routers. We denote a router at level l in the tree a *level-l router* or *L_l-router*. An L_0-router is the parent of two leaf nodes in the tree, which are processing nodes. A *router node* at level l in the fat tree consists of 2^l individual L_l-routers. In Fig. 1, a router node is shown as a rectangle if it comprises more than one router. Furthermore, we have annotated one router node at each level in the tree with the corresponding levels L_0, L_1, L_2, and L_3.

Using the terminology introduced informally above, we define the fat-tree network as follows. A fat-tree network with n processing elements is a complete binary tree with n leaf nodes, which are the processing nodes, and $n-1$ router nodes. Since the height of the complete binary tree is $h = \lg n$, the router node at the root of the tree is an L_{h-1}-router node consisting of 2^{h-1} individual L_{h-1}-routers. We define the structure of the fat tree recursively. Given a router node L_l at level l in the fat tree, connect its downstream ports a to the left subtree and its downstream ports b to the right subtree. There are 2^l bidirectional links connecting the router node at level l with the upstream ports of each of its left and right children at level $l-1$. The children of router nodes at level $l = 0$ are processing nodes. Furthermore, we split the range of processing node identifiers $[0, \ldots, n = 2^h[$, by assigning the lower half to the left subtree and the upper half to the right subtree recursively. At the root node, the lower half $[0, \ldots, 2^{h-1}[$ is passed to the left subtree and the upper half $[2^{h-1}, \ldots, 2^h[$ to the right subtree. When we reach a processing node, only a single identifier remains in the range, which we assign to the processing node.

4. Structural analysis

We analyze the collision behavior of a fat-tree network to compute the probability of a collision between two messages. All random choices involving message sources and destinations as well as upstream port selections are assumed to be with respect to the uniform distribution. However, the source and destination of a message are assumed to be distinct.

Lemma 1 (*2-Message collision probability*). *Two messages with randomly chosen, distinct sources and independently and randomly chosen destinations, and sent at the same time, collide on average with probability*

$$\Pr[C_2] = \frac{1}{(n-1)^3}\left(n^2\left(\frac{1}{2}\lg n - \frac{2}{3}\right) + \frac{2}{3}\right) = \Theta\left(\frac{\lg n}{n}\right)$$

in the fat tree with n processing nodes described in Section 3. *Moreover,* $\Pr[C_2]$ *can be bound as follows for* $n > 0$:

$$\frac{\lg n}{3n} \leqslant \Pr[C_2] \leqslant \frac{\lg n}{2n}.$$

Proof. We employ an accounting argument of basic collision events covering the entire sample space of possible collisions. We fix the sender of message m_1 at node 0 of the fat tree without loss of generality. This gives us a choice of $n-1$ destinations for m_1, $n-1$ possible sources of message m_2, and $n-1$ possible destinations for m_2. Hence, our sample space comprises $(n-1)^3$ distinct elementary events.

We utilize the symmetry of the fat tree to account for entire subtrees at a time. In particular, we consider v-subtrees with 2^v nodes and denote as $\Pr[k, i, j]$ the probability that m_1 with source node 0 and its destination node in the i-subtree collides with m_2 with its source node in the k-subtree and its destination node in the j-subtree. The subtrees are uniquely specified such that all nodes in a v-subtree have the same dilation $2(v+1)$ from the respective reference node, where *dilation* shall be the number of links between two nodes. In other words, pick a reference node, and collect the set of nodes with dilation v from the reference node, then observe that this set of nodes is a complete subtree of the fat tree. The following discussion illustrates the concept of the v-subtree.

The destination subtrees of m_1 are the i-subtrees. Since the source of m_1 is fixed at node 0, we can easily identify the i-subtrees with respect to node 0. For $i = 0$, the only node with dilation $2(0+1) = 2$ is node 1; cf. Fig. 1. Thus, the $(i = 0)$-subtree is $\{1\}$. For $i = 1$, the nodes with dilation $2(1+1) = 4$ are 2 and 3. Therefore, the $(i = 1)$-subtree is $\{2, 3\}$. Analogously, the $(i = 2)$-subtree is $\{4, 5, 6, 7\}$, the $(i = 3)$-subtree is $\{8, \ldots, 15\}$, etc. We observe that, in general, the $(i = v)$-subtree is the set of nodes $\{2^v, \ldots, 2^{v+1} - 1\}$.

The k-subtrees contain the possible source nodes of m_2 with respect to source node 0 of m_1. Therefore, the k-subtrees are identical to the i-subtrees. The j-subtrees contain the possible destination nodes of m_2 with respect to its source node. The j-subtrees depend on the particular choice of the source node of m_2. For example, consider node 10 as the source of m_2. The $(j = 0)$-subtree is $\{11\}$, the $(j = 1)$-subtree $\{8, 9\}$, the $(j = 2)$-subtree $\{12, 13, 14, 15\}$, the $(j = 3)$-subtree is $\{0, \ldots, 7\}$, and so on. With respect to source node 0 of message m_1, node 10 is an element of the $(k = 3)$-subtree $\{8, \ldots, 15\}$.

We can compute $\Pr[C_2]$ by summing up the individual probabilities $\Pr[k, i, j]$, presuming that $\Pr[k, i, j]$ accounts for the *average probability* of a collision for *all* source nodes of m_2 in the k-subtree, *all* destination nodes of m_1 in the i-subtree, and *all* destination nodes of m_2 in the j-subtree. Since for a fat tree with n processing nodes the largest subtree contains $n/2$ processing nodes, we obtain for $\Pr[C_2]$:

$$\Pr[C_2] = \frac{1}{(n-1)^3}\sum_{k=0}^{\lg n-1} 2^k \sum_{i=0}^{\lg n-1} 2^i \sum_{j=0}^{\lg n-1} 2^j \Pr[k, i, j]. \tag{1}$$

A structural analysis based on a particular choice of k allows us to determine $\Pr[k, i, j]$ for all i and j. This analysis results in the following matrix of probabilities for a particular k, for $0 \leqslant i \leqslant \lg n - 1$, and for $0 \leqslant j \leqslant \lg n - 1$:

$$\Pr[k, i, j] = \begin{array}{c} \\ 0 \\ \vdots \\ i{=}k \\ \vdots \\ \lg \frac{n}{2} \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & \cdots & & j{=}k & \cdots & \lg \frac{n}{2} \end{array} \\ \begin{pmatrix} & & \vdots & & & \\ & 0 & \frac{i+2}{2^{k+1}} & & & \\ & & \vdots & & 0 & \\ \cdots & \frac{j+2}{2^{k+1}} \cdots & 0 & & & \\ & & & \ddots & & \\ & 0 & & & \frac{k+2}{2^{i+1}} & \\ & & & & & \ddots \end{pmatrix} \end{array}. \tag{2}$$

We can derive Eq. (2) by means of a case-by-case analysis depending on the relationship between i, j, and k. For the sake of clarity, we discuss each case in detail. Although this results in a lengthy analysis, it is nothing but a straightforward accounting of elementary events:

$i < k \wedge j < k$: First, consider the example $i = 2$, $k = 3$, and $j = 1$. The destination subtree of m_1 is the $(i = 2)$-subtree $\{4, 5, 6, 7\}$. The source node of m_2 is in the $(k = 3)$-subtree $\{8, \ldots, 15\}$. The destination subtree of m_2 is the $(j = 1)$-subtree with respect to the source node of m_2. It can be one of the four subtrees $\{8, 9\}$, $\{10, 11\}$, $\{12, 13\}$, or $\{14, 15\}$ only. For any choice of the source of m_2, this subtree is contained in the k-subtree. Thus messages m_1 and m_2 travel through disjoint subtrees, and cannot collide. In general, the destination subtree of m_1 is the i-subtree $\{2^i, \ldots, 2^{i+1} - 1\}$. The source node of m_2 is in the k-subtree $\{2^k, \ldots, 2^{k+1} - 1\}$, which is disjoint from the i-subtree for $i < k$. Finally, for $j < k$, the j-subtree is a proper subset of the k-subtree, and therefore disjoint from the i-subtree as well. As a consequence, $\Pr[k, i, j] = 0$ for $i < k \wedge j < k$.

$i = k = j$: Consider the case $i = k = j = 2$. The i-subtree and the k-subtree are $\{4, 5, 6, 7\}$. The j-subtree is $\{0, 1, 2, 3\}$, independent of the particular source of m_2 within the k-subtree. Message m_1 travels from node 0 to one of the nodes in the i-subtree, and m_2 from one of the nodes in the k-subtree to one of the nodes in the j-subtree. Both messages traverse in opposite directions through one of the L_2-routers in Fig. 1, potentially the same router. Since all

links are bidirectional, the tree supports this *criss-crossing* message pattern without collisions, even if the messages traverse through the same router. We can easily generalize this case and see that $\Pr[k, i, j] = 0$ for all $i = k = j$.

$i > k > j$ or $i < k < j$: Consider the example $i = 3$, $k = 2$, and $j = 1$. The i-subtree is $\{8, \ldots, 15\}$ and the k-subtree is $\{4, 5, 6, 7\}$. For any choice of the source of m_2 in the k-subtree, its destination is in the $(j = 1)$-subtree, which must be either $\{4, 5\}$ or $\{6, 7\}$, and is a proper subset of the k-subtree. Thus, message m_2 is confined to the k-subtree, whereas m_1 travels through the tree to the i-subtree without traversing any of the routers connecting the nodes of the k-subtree. The fact that messages m_1 and m_2 never traverse the same router is easily generalized for $i > k > j$. The case $i < k < j$ is symmetric. Therefore, $\Pr[k, i, j] = 0$ for $i > k > j$ and $i < k < j$.

$i > j > k$ or $j > i > k$: Consider $i = 3$, $j = 2$, and $k = 1$. The i-subtree is $\{8, \ldots, 15\}$ and the k-subtree is $\{2, 3\}$. There exists exactly one j-subtree for all choices of the source of m_2 in the k-subtree, which is the j-subtree $\{4, 5, 6, 7\}$. The key observation here is that both messages m_1 and m_2 travel upstream, partially in parallel, until they reach a router where m_1 travels further upstream towards the i-subtree whereas m_2 turns downstream towards the j-subtree. In the example, this happens at one of the L_2-routers in Fig. 1. Since collisions cannot happen on the upstream paths of two messages, m_1 and m_2 do not collide. The case where $j > i > k$ is similar, except that m_2 travels further upstream than m_1. We find that $\Pr[k, i, j] = 0$ for all $i > j > k$ and all $j > i > k$.

$j < i = k$ or $i < j = k$: These cases correspond to the nonzero elements in row $i = k$ and column $j = k$ of the matrix in Eq. (2), respectively. We discuss the case $j < i = k$. Case $i < j = k$ holds by symmetry. The destination node of m_1 with source node 0 is in the i-subtree $\{2^i, \ldots, 2^{i+1} - 1\}$. Since $i = k$, the k-subtree equals the i-subtree, and the source node of m_2 is a node of the i-subtree. Without loss of generality, we consider node $2^i = 2^k$ to be the source node of m_2. The destination of m_2 is in the j-subtree, which is a proper subset of the k-subtree, because $j < k$. For example, for $i = k = 2$ and $j = 1$ both the i- and k-subtree are $\{4, 5, 6, 7\}$. If we pick the source of m_2 to be node $2^2 = 4$, the $(j = 1)$-subtree containing the potential destinations of m_2 is $\{6, 7\}$. In general, we find that the destination of m_2 must be in the j-subtree $\{2^k + 2^j, \ldots, 2^k + 2^{j+1} - 1\}$ if the source of m_2 is node 2^k.

Let us study the possible collision scenarios for messages m_1 and m_2 by means of the preceding example. If m_1 has destination 4 or 5, m_2 may travel to destination 6 or 7 simultaneously without collision. If both m_1 and m_2 have the same destination, which may be node 6 or node 7, the messages will collide with probability 1. This collision may happen either at the L_0-router connecting the destination node, or at one of the L_1-routers connecting subtrees $\{4, 5\}$ and $\{6, 7\}$ if both messages attempt to traverse the same router. In case that the destinations of m_1 and m_2 are different, say m_1 is destined for node 6 and m_2 for node 7, a collision may occur at one of the L_1-routers connecting subtrees $\{4, 5\}$ and $\{6, 7\}$ if both messages attempt to traverse the same router. If m_1 and m_2 travel though different L_1-routers, they can travel collision-free through the L_0-router to their destinations.

We can generalize the observations from this example assuming that the source of m_2 is 2^k and the destination of m_2 is $2^k + 2^j$. We dissect the j-subtree $\{2^k + 2^j, \ldots, 2^k + 2^{j+1} - 1\}$ into r-subtrees $\{2^k + 2^j + 2^r, \ldots, 2^k + 2^j + 2^{r+1} - 1\}$ for $0 \leqslant r < j$. For example, with $i = k = 3$ and $j = 2$, the source of m_2 is node 8, the destination of m_2 is node 12, and the $(j = 2)$-subtree is $\{12, 13, 14, 15\}$. Then, the $(r = 0)$-subtree is $\{13\}$ and the $(r = 1)$-subtree is $\{14, 15\}$.

We observe that if the destination of m_1 is in the r-subtree, then m_1 and m_2 cannot collide at any of the L_v-routers for $0 \leqslant v \leqslant r$. Thus, collisions may occur only at routers at level $r + 1$ or higher in the j-subtree. We account for the collisions of m_1 and m_2 due to all routers at level $r + 1$ and higher by counting all *paths* of m_1 and m_2 that reach a router node at level $r + 1$ and traverse the same L_{r+1}-router. The message paths are determined randomly due to the port selections on the upstream paths. Since the upstream paths of m_1 and m_2 are disjoint for $j < i = k$, the random selections are independent. Due to this independence, and since there are 2^{r+1} L_{r+1}-routers on the downstream paths of m_1 and m_2, the probability that m_1 or m_2 traverse a particular L_{r+1}-router is $1/2^{r+1}$, respectively. Thus, the probability that the paths of both m_1 and m_2 traverse the same router of a router node at level $r + 1$ is $2^{r+1} \cdot 1/2^{r+1} \cdot 1/2^{r+1} = 1/2^{r+1}$. Consequently, the probability of a collision of m_1 and m_2 at a router node at level $r + 1$ or higher is $1/2^{r+1}$.

To compute probability $\Pr[k, i, j]$, we sum up the probabilities of the independent collision scenarios that may occur for m_1 and m_2. We fix the source of m_2 at 2^k and the destination at $2^k + 2^j$. By renumbering the nodes in the k-subtree, we find that the collision probability for all possible sources and destinations of m_2 is equal to this particular choice. Therefore, it is sufficient to account for all destinations of m_1 in the i-subtree with a fixed source and destination of m_2 to compute the average collision probability. The collision probability is 1, if m_1 chooses the same destination $2^k + 2^j$ as m_2. This happens with probability $1/2^i$ since there are 2^i possible destinations for m_1. If the destination of m_1 is outside of the j-subtree, the collision probability is 0. Otherwise, message m_1 may choose one of the 2^r destination nodes in the r-subtree, which is a proper subset of the j-subtree. The 2^r destinations may be chosen by m_1 with probability $2^r/2^i$, each of which has the collision probability $1/2^{r+1}$ derived above. We need to sum up the probabilities over the disjoint r-subtrees for $0 \leqslant r < j$. Therefore, we obtain

$$\Pr[k, i, j] = \frac{1}{2^i} 1 + \sum_{r=0}^{j-1} \frac{2^r}{2^i} \frac{1}{2^{r+1}} = \frac{j+2}{2^{i+1}}.$$

Since we are considering the case $i = k$, we have $\Pr[k, i, j] = (j+2)/2^{i+1}$ for $j < i = k$, yielding the elements in the matrix of Eq. (2) for row $i = k$. The column elements for $i < j = k$ follow by symmetry.

$i = j > k$: This case corresponds to the nonzero elements on the main diagonal of the matrix in Eq. (2). The i-subtree is equal to the j-subtree $\{2^j, \ldots, 2^{j+1} - 1\}$. Similar to the previous case, we fix the destination of m_2 at node 2^j, and dissect the j-subtree $\{2^j, \ldots, 2^{j+1}-1\}$ into r-subtrees $\{2^j + 2^r, \ldots, 2^j + 2^{r+1} - 1\}$ for $0 \leqslant r < j$. Now, the accounting of collisions depends on k rather than j, because the source of m_2 determines the routers on the downstream paths of m_1 and m_2 at which collisions can occur.

Let us consider an example first. Assume that $i = j = 3$, that is the $(i = j = 3)$-subtree is $\{8, \ldots, 15\}$, and the destination of m_2 is node $2^j = 8$. For $k = 0$, the $(k = 0)$-subtree is $\{1\}$. The only possible source of m_2 is node 1. Recall that we fix the source of m_1 to be node 0. Inspection of the tree in Fig. 1 reveals that m_1 and m_2 cannot collide except when the destinations of m_1 and m_2 coincide at node 8, because the upstream-port selections are not independent. Note that no collision occurs even for node 9 as the destination of m_1. We may argue that the L_0-router connecting subtrees $\{0\}$ and $\{1\}$ guarantees that m_1 and m_2 travel collision-free through the tree such that they arrive at different L_1-routers of the router node connecting subtrees $\{8, 9\}$ and $\{10, 11\}$. From there, m_1 and m_2 can travel through different upstream ports of the L_0-router connecting nodes 8 and 9 to their destinations.

For $k = 1$, the k-subtree is $\{2, 3\}$. We may choose node 2 as the source of m_2. The destination of m_2 remains fixed at node 8. If the destination of m_1 is node 9, a collision occurs if m_1 and m_2 arrive at the same L_1-router connecting subtrees $\{8, 9\}$ and $\{10, 11\}$. Such a path is possible due to the independent upstream-port selections of the L_0-routers at the source nodes 0 and 2. If these L_0-routers select upstream ports such that m_1 and m_2 arrive at the same L_1-router connecting the L_0-routers, both messages will arrive at the same L_1-router connecting subtrees $\{8, 9\}$ and $\{10, 11\}$, leading to a collision.

The situation is similar for $k = 2$. In this case, collisions may occur at the L_1 or L_2-routers on the downstream paths of m_1 and m_2. The dissection of the destinations of m_1 into r-subtrees restricts the number of message paths of m_1 such that there exists only one out of 2^{r+1} upstream paths that causes a collision on the downstream path. For $r = 0$, the destination of m_1 is node 9. There are $2^{0+1} = 2$ possible upstream paths of m_1 depending on the path selection of the L_0-router at source node 0 of m_1. One of the two upstream paths leads to a collision at an L_1-router or L_2-router on the downstream path. The other upstream path is collision-free. For $r = 1$, one out of 2^2 upstream paths leads to a collision at an L_2-router on the downstream path.

In general, we find that collisions may occur for a particular k at router nodes at level $r + 1$ or higher for $0 \leqslant r < k$. The probability that the messages collide can be derived by considering their upstream paths. Due to the symmetry of the tree, a collision occurs on the downstream path at level $r + 1$ or higher due to independent path selections of the upstream routers at levels below $r + 1$. Assuming that the path of m_2 is fixed, there are 2^{r+1} possible upstream paths for m_1, only one of which can lead to a collision on the downstream path. Since the random upstream-path selections are independent below router level $r + 1$, the probability that m_1 chooses the collision path is $1/2^{r+1}$.

Using the same accounting argument for the paths of m_1 and m_2 as in case $j < i = k$, we find that

$$\Pr[k, i, j] = \frac{1}{2^i} 1 + \sum_{r=0}^{k-1} \frac{2^r}{2^i} \frac{1}{2^{r+1}} = \frac{k+2}{2^{i+1}}.$$

This result coincides with the elements on the diagonal of the matrix in Eq. (2) for $i = j > k$.

We now turn to computing $\Pr[C_2]$ from Eq. (1). The sums over i and j can be computed as a function of k from Eq. (2) by adding up the row elements for $i = k$ and $0 \leqslant j < k$, the column elements for $j = k$ and $0 \leqslant i < k$, and the diagonal elements for $i = j$ and $k < i \leqslant \lg n - 1$:

$$\begin{aligned}
&\sum_{i=0}^{\lg n-1} 2^i \sum_{j=0}^{\lg n-1} 2^j \Pr[k, i, j] \\
&\quad = \sum_{i=0}^{k-1} 2^{i+k} \frac{i+2}{2^{k+1}} + \sum_{j=0}^{k-1} 2^{j+k} \frac{j+2}{2^{k+1}} + \sum_{v=k+1}^{\lg n-1} 2^{2v} \frac{k+2}{2^{v+1}} \\
&\quad = \sum_{i=0}^{k-1} (i+2) 2^i + \sum_{v=k+1}^{\lg n-1} \frac{k+2}{2} 2^v \\
&\quad = k2^k + \frac{k+2}{2} \left(n - 2^{k+1}\right).
\end{aligned}$$

Finally, we compute the sum over k to obtain $\Pr[C_2]$:

$$\begin{aligned}
&(n-1)^3 \Pr[C_2] \\
&\quad = \sum_{k=0}^{\lg n-1} 2^k \left(k2^k + \frac{k+2}{2} \left(n - 2^{k+1}\right)\right) \\
&\quad = \sum_{k=0}^{\lg n-1} k2^{2k} + n \sum_{k=0}^{\lg n-1} \frac{k+2}{2} 2^k - \sum_{k=0}^{\lg n-1} (k+2) 2^{2k} \\
&\quad = \frac{n}{2} \sum_{k=0}^{\lg n-1} (k+2) 2^k - 2 \sum_{k=0}^{\lg n-1} 2^{2k} \\
&\quad = \frac{n^2}{2} \lg n - \frac{2}{3}(n^2 - 1) \\
&\quad = n^2 \left(\frac{1}{2} \lg n - \frac{2}{3}\right) + \frac{2}{3}.
\end{aligned}$$

Proving the upper and lower bounds for $\Pr[C_2]$ is trivial and is left to the reader. □

We can now reap the benefits from the simple yet laborious chore of proving Lemma 1, and apply Lemma 1 to the case where more than two messages enter the network.

5. The balls-and-bins model

We now consider $m > 2$ messages. We assume that m distinct message sources are chosen randomly, and that m potentially identical destinations are chosen independently and at random. All random choices are with respect to the uniform distribution. Since each source can transmit only one message at a time, n is an upper bound for m, and we have $2 < m \leqslant n$. Lemma 1 enables us to model the collision behavior of m messages by means of a classical balls-and-bins game. A message transmission corresponds to a ball that is tossed randomly and independent of other tosses into a *collision bin*. Two messages collide, if the corresponding balls land in the same collision bin. The only piece of information that we supply to the balls-and-bins game is probability $\Pr[C_2]$ according to Lemma 1. The number of collision bins shall reflect this probability, and is therefore chosen as follows.

Corollary 1 (*Bin calibration*). *The number of bins b of the balls-and-bins game, where each ball corresponds to a message and each bin to a collision, is approximated to* $2n/\lg n$.

Proof. We toss two balls independently and at random into b collision bins. The probability that both balls land in the same bin is $1/b$. This probability shall be equal to the average collision probability of two messages $\Pr[C_2]$. Consequently, choosing

$$b = \frac{2n}{\lg n} \leqslant \frac{1}{\Pr[C_2]}$$

yields a conservative analysis, but does not affect our complexity result, because b differs by a small constant factor from the actual value only. □

Recall that $\Pr[C_2]$ is the average probability across all possible distinct sources and potentially identical destinations. Therefore, when considering $m > 2$ messages, more than two balls may land in a particular collision bin. All of the corresponding messages shall collide in the network. Our key approximation of the collision behavior is that only one of these messages shall survive the collisions. Hence, all but one ball in a collision bin correspond to rejected messages, and one ball corresponds to a delivered message. We call the tossing of balls into collision bins a *collision toss* and its equivalent with respect to message transmissions a *delivery round*. All messages rejected during one delivery round are retried in a subsequent delivery round. The number of delivery rounds needed to deliver all messages determines the performance of the network.

The model of the collision behavior by means of the balls-and-bins game described above deserves further discussion. In fact, this model may appear to be unacceptably crude, because it ignores a variety of dependencies, most notably those dependencies imposed by the distribution of message destinations. We argue, however, that we may neglect these dependencies safely. We provide empirical evidence in Section 7 that the balls-and-bins model reflects reality at the level of end-to-end performance with sufficient accuracy, indeed.

As an aside, let us show that the dependencies due to the distribution of message destinations affect the number of delivery rounds by a constant factor only, compared to neglecting them. To account for the distribution of message destinations, we may construct a model of two balls-and-bins games. The first game consists of a single toss, the *destination toss*, of m balls into n *destination bins* representing the random choice of message destinations. The second game consists of repeated *collision tosses* into $1/\Pr[C_2]$ collision bins representing delivery rounds. During the second game we may toss all of the balls representing a single destination into the same collision bin. This construction would express the fact that if there were no messages other than those with the same destination, these messages will surely collide with each other. The destination bin with the maximum number of balls constitutes a *critical path* across the delivery rounds. For $m = n$ messages, the critical-path length is $\Theta(\log n / \log\log n)$ with high probability [17]. Note that, like our original model, this more realistic model is merely an approximation as well, because it treats the collision behavior by means of the average collision probability $\Pr[C_2]$.

Let us call our original, simple balls-and-bins game *Model I*, and the model with the destination toss *Model II*. Simulations show that the number of delivery rounds due to these two models differ by a constant factor only. Fig. 3 shows the number of delivery rounds for $2^4 \leqslant n \leqslant 2^{20}$ processing nodes and for the number of messages $m = n/8$ and n. Both graphs show the minimum, average, and maximum number of delivery rounds as error bars. In addition, we show the ratio of the average number of delivery rounds for Model I and Model II, the ratio of the maximum number of delivery rounds, as well as their mean values over n. The mean values are horizontal lines and are consequently independent of n. Since the data points deviate only slightly from the mean values, we conclude that the number of delivery rounds due to Model I and Model II differ by a small constant factor only.

Using the potential method [21,2], we can construct a proof that considers the dependencies of the destination distribution expressed by Model II. This proof yields the claimed result that the number of delivery rounds is bound by $O(\lg m)$. Although the potential method is an elegant proof technique, we feel that using Model I results in an even simpler, straightforward proof, and it exposes the inherent problem structure clearly. In the following, we are therefore concerned with the analysis of Model I only.

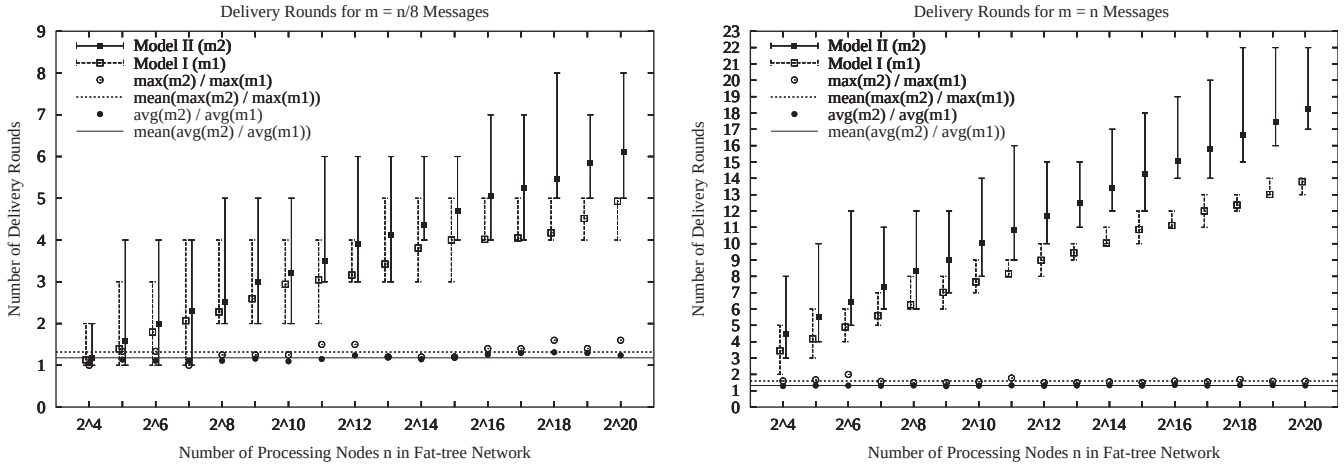

Fig. 3. Simulations of Model I without a destination toss and Model II including the destination toss for $m = n/8$ (left) and $m = n$ (right). The number of delivery rounds due to the two models differ by a small constant factor only.

We now turn our attention to results from basic probability theory about the balls-and-bins game underlying Model I. A *delivery round* corresponds to tossing m balls into b collision bins. We calculate the number of delivered and rejected messages as follows. After tossing m balls, there will be b_e empty bins and $b_n = b - b_e$ nonempty bins. The number of delivered messages corresponds to the number of nonempty bins b_n, because each nonempty bin contains at least one ball. The number of rejected messages corresponds to the number of balls in the nonempty bins minus one ball per nonempty bin which corresponds to a delivered message. Hence, the number of rejected messages is $m - b_n$.

The expected number of empty bins in the balls-and-bins game can be calculated as follows. The probability that a bin remains empty after tossing m balls is

$$\left(1 - \frac{1}{b}\right)^m \leqslant e^{-\frac{m}{b}},$$

since $1 + x \leqslant e^x$ for all x. Let X_i be an indicator variable with value 1 if bin i is empty and with value 0 otherwise. Then, $E[X_i] = (1 - 1/b)^m$. By linearity of expectation, the expected number of empty bins is

$$E[b_e] = \sum_{i=1}^{b} E[X_i] = b\left(1 - \frac{1}{b}\right)^m \leqslant be^{-\frac{m}{b}}. \tag{3}$$

By linearity of expectation, the expected number of delivered messages is

$$\begin{aligned} E[D] &= b - E[b_e] = b\left(1 - \left(1 - \frac{1}{b}\right)^m\right) \\ &\geqslant b\left(1 - e^{-\frac{m}{b}}\right), \end{aligned} \tag{4}$$

and the expected number of rejected messages is

$$E[R] = m - E[D] \leqslant m - b\left(1 - e^{-\frac{m}{b}}\right). \tag{5}$$

The rejected messages of one delivery round are subject to retry in the subsequent delivery round.

6. Proof of time bound

We model the fat-tree network with n processing nodes and the architecture and collision behavior described in Section 3 by means of Model I developed in Section 5. Each of the $m \leqslant n$ messages corresponds to a ball. The collision bins are used to capture the collision behavior of the network. We will establish the following statement:

Theorem 1 (*Bound of delivery rounds*). *In balls-and-bins Model I of a fat-tree network with n processing nodes, $m \leqslant n$ messages with randomly chosen, distinct sources and independently and randomly chosen destinations are delivered within $O(\lg m + \ln 1/\varepsilon)$ delivery rounds with probability $1 - \varepsilon$ for any $\varepsilon > 0$.*

Our proof consists of two phases. In Phase I we prove that the number of messages $m > 2n/\lg n$ is reduced to $2n/\lg n$ messages within $O(\lg m + \ln 1/\varepsilon)$ rounds with probability $1 - \varepsilon$. In Phase II we prove that $m \leqslant 2n/\lg n$ messages are delivered within $O(\lg m + \ln 1/\varepsilon)$ rounds with probability $1 - \varepsilon$. Together, Phases I and II yield the claimed bound.

To facilitate the analysis, we assume that the retry strategy of the network interfaces of the processing nodes is such that the delivery rounds do not overlap. Thus, all network interfaces wait until all messages transmitted at the beginning of one delivery round are either delivered or rejected.

6.1. Analysis of Phase I

The analysis of Phase I for $m > 2n/\lg n$ messages is based on the observation that tossing $m > 2n/\lg n$ balls into $b = 2n/\lg n$ collision bins will inevitably - than one ball

landing in one or more bins. The number of balls landing in each bin is likely to be relatively large, corresponding to a large number of collisions during that round.

The distribution of m balls over b bins follows the binomial distribution. A well-known result, that we may apply to this distribution, is the *Markov Inequality* [17]. It states that for a nonnegative random variable X and any positive real t we have

$$\Pr[X \geqslant t] \leqslant \frac{E[X]}{t}.$$

Since the number of empty bins b_e after a collision toss is a random variable, we may apply the Markov Inequality to b_e. During Phase I, we have $m > 2n/\lg n = b$. Thus, the expected number of empty bins when tossing m balls is according to Eq. (3):

$$E[b_e] \leqslant be^{-\frac{m}{b}} < \frac{b}{e} \quad \text{for } m > b.$$

We choose $t = 2b/e$, and obtain from the Markov Inequality

$$\Pr[b_e \geqslant 2b/e] \leqslant \frac{E[b_e]}{2b/e} \leqslant \frac{1}{2}.$$

Equivalently, the probability that the number of empty bins b_e is less than $2b/e$ is greater than $1/2$.

We define a delivery round to be a *successful delivery round* if less than $2b/e$ collision bins remain empty. Correspondingly, more than $b - 2b/e = b(1 - 2/e)$ messages are delivered in a successful round. Amongst all delivery rounds, a successful delivery round occurs at least with probability $1/2$ according to the Markov Inequality. By definition, for each delivery round of Phase I we have $m > b$. Therefore, in each successful delivery round of Phase I, a constant number of at least $b(1 - 2/e)$ messages is delivered. Considering successful delivery rounds only, Phase I ends after S successful delivery rounds when the number of remaining messages is reduced to b messages. Therefore, Phase I is subject to the *boundary condition*:

$$m - Sb\left(1 - \frac{2}{e}\right) = b.$$

Solving for S, we obtain

$$\begin{aligned} S &= \frac{1}{(1-2/e)}\left(\frac{m}{b} - 1\right) \\ &= \frac{1}{(1-2/e)}\left(\frac{m \lg n}{2n} - 1\right) \qquad (6) \\ &\leqslant \frac{1}{(1-2/e)}\left(\frac{1}{2}\lg m - 1\right) \qquad (7) \\ &= O(\lg m), \end{aligned}$$

since $m/\lg m \leqslant n/\lg n$ for $2 \leqslant m \leqslant n$. Thus, we have established that the number of messages $m > 2n/\lg n$ is reduced to $2n/\lg n$ messages within $O(\lg m)$ successful delivery rounds during Phase I.

It remains to be shown that the number of ordinary delivery rounds R containing S successful delivery rounds is of order $O(\lg m)$ with high probability. To that end we may assume that delivery rounds are independent Bernoulli trials, and apply another well known result, the *Chernoff Inequality* [17]: For a random variable X defined by $\Pr[X = 1] = p$ of n Bernoulli trials with probability p of success, $\mu = E[X] = np$, and $0 < \delta \leqslant 1$, we have

$$\Pr[X < (1-\delta)\mu] < e^{-\frac{\mu\delta^2}{2}}.$$

For convenience we use symbol $p_s = 1/2$ to denote the lower bound for the probability of the occurrence of a successful delivery round. We assume that the number of ordinary delivery rounds R is

$$\begin{aligned} R &= \frac{1}{p_s}(2S - 4\ln\varepsilon) \\ &\leqslant 2\left(\frac{2}{(1-2/e)}\left(\frac{1}{2}\lg m - 1\right) - 4\ln\varepsilon\right) \\ &= O(\lg m - \ln\varepsilon) \end{aligned}$$

for a small value ε. This construction of R is justified below due to the fact that the probabilities following from the Chernoff bound yield the desired result. According to basic probability theory, the expected number of successful delivery rounds within R rounds is at least

$$\mu_s = Rp_s = 2S - 4\ln\varepsilon,$$

because a successful round occurs at least with probability p_s. We use a slight modification of the Chernoff bound

$$\Pr[X < \mu_s - \alpha p_s] < e^{-\frac{(\alpha p_s)^2}{2\mu_s}}, \qquad (8)$$

where $0 < \alpha p_s \leqslant \mu_s$, and choose

$$\alpha = \frac{1}{p_s}(S - 4\ln\varepsilon).$$

Note that the condition $\alpha p_s \leqslant \mu_s$ holds for any ε, since $\alpha p_s = S - 4\ln\varepsilon \leqslant 2S - 4\ln\varepsilon = \mu_s \Leftrightarrow S \leqslant 2S$. We can express μ_s as a function of α as follows:

$$\mu_s = 2\alpha p_s + 4\ln\varepsilon.$$

Now, let X_s be a random variable denoting the number of successful delivery rounds. We apply the Chernoff bound of Eq. (8) to X_s and obtain:

$$\begin{aligned} & & \Pr[X_s < \mu_s - \alpha p_s] &< e^{-\frac{(\alpha p_s)^2}{2\mu_s}} \\ &\Leftrightarrow & \Pr[X_s < 2S - 4\ln\varepsilon - S + 4\ln\varepsilon] &< e^{-\frac{(\alpha p_s)^2}{4\alpha p_s + 8\ln\varepsilon}} \\ &\Rightarrow & \Pr[X_s < S] &\leqslant e^{-\frac{\alpha p_s}{4}} \\ & & &= e^{-S/4 + \ln\varepsilon} \\ & & &\leqslant \varepsilon \\ &\Leftrightarrow & \Pr[X_s > S] &\geqslant 1 - \varepsilon. \end{aligned}$$

Hence, the probability that the number of successful delivery rounds X_s within $R = O(\lg m + \ln(1/\varepsilon))$ delivery rounds exceeds the required number of successful delivery rounds S is greater than or equal to $1-\varepsilon$. Therefore, the number of delivery rounds needed to deliver $m > 2n/\lg n$ messages with $2n/\lg n$ messages remaining is $O(\lg m + \ln(1/\varepsilon))$ with probability at least $1-\varepsilon$ for any $\varepsilon > 0$. We have consequently established the proof for Phase I. □

6.2. Analysis of Phase II

During Phase II we inject $m \leqslant 2n/\lg n$ messages into the network. Correspondingly, in our balls-and-bins model, we toss $m \leqslant 2n/\lg n$ balls into $b = 2n/\lg n$ collision bins. Since the number of balls is less than or equal to the number of bins, we can expect to make progress by delivering a *constant fraction* of the messages in each delivery round. In contrast, we have shown in Section 6.1 that a *constant number* of messages is delivered per delivery round in Phase I.

We apply the Markov Inequality to the number of rejected messages in a delivery round as follows. According to Eq. (5), the expected number of rejected messages is $E[R] = m - E[D] \leqslant m - b\left(1 - e^{-\frac{m}{b}}\right)$. Choosing $t = (1-\alpha)m$ with $0 < \alpha < 1$, we express that the fraction $(1-\alpha)$ of the m balls tossed during the delivery round corresponds to rejected messages. Applying the Markov Inequality to the number of rejected messages R, we obtain for $2 \leqslant m \leqslant b$:

$$\begin{aligned}
\Pr[R \geqslant (1-\alpha)m] &\leqslant \frac{E[R]}{(1-\alpha)m} \\
&\leqslant \frac{1}{(1-\alpha)m}\left(m - b\left(1 - e^{-\frac{m}{b}}\right)\right) \\
&= \frac{1}{(1-\alpha)}\left(1 - \frac{b}{m}\left(1 - e^{-\frac{m}{b}}\right)\right) \\
&\leqslant \frac{1}{(1-\alpha)}\left(1 - \left(1 - e^{-1}\right)\right) \\
&= \frac{1}{(1-\alpha)e}.
\end{aligned}$$

Note that $f(x) = x(1-e^{-1/x}) \geqslant 1 - e^{-1}$ for $x \geqslant 1$, because df/dx decreases monotonically towards 0 for $x \to \infty$ and $df/dx(1) = 1 - 2/e > 0$. To be meaningful, probability $1/((1-\alpha)e)$ must be less than 1, providing us with the condition $\alpha < 1 - e^{-1}$.

Since the number of delivered messages is $D = m - R$, we have

$$R \geqslant (1-\alpha)m \quad \Leftrightarrow \quad D \leqslant m - (1-\alpha)m = \alpha m.$$

We substitute this term in the Markov Inequality to obtain

$$\begin{aligned}
\Pr[R \geqslant (1-\alpha)m] &\leqslant \tfrac{1}{(1-\alpha)e} \\
\Leftrightarrow \quad \Pr[D \leqslant \alpha m] &\leqslant \tfrac{1}{(1-\alpha)e} \\
\Leftrightarrow \quad \Pr[D \geqslant \alpha m] &\geqslant 1 - \tfrac{1}{(1-\alpha)e}.
\end{aligned}$$

We have therefore established that at least a constant fraction α of m messages is delivered with probability at least $1 - 1/(1-\alpha)e$ within a single delivery round. We define a *successful delivery round* for Phase II to be a delivery round in which at least αm messages are delivered. A successful delivery round occurs with probability at least $1 - 1/(1-\alpha)e$ amongst the delivery rounds.

Considering successful delivery rounds only, we know that at most $(1-\alpha)m$ messages are rejected and must be retried in the subsequent round. Hence, after k successful delivery rounds, at most $(1-\alpha)^k m$ messages remain to be delivered. Since the last remaining message will be delivered without any collisions, we have the boundary condition

$$(1-\alpha)^S m = 1.$$

Choosing $\alpha = 1/2$, we obtain the number of successful delivery rounds

$$S = \lg m$$

and the probability for the occurrence of a successful delivery round is $1 - 2/e$.

It remains to be shown that the number of ordinary delivery rounds R is of order $\lg m$ with high probability. Analogous to Phase I, we construct a Chernoff bound argument. With the probability for a successful delivery round $p_s = 1 - 2/e$, we assume that the number of delivery rounds is

$$R = \frac{1}{p_s}(2\lg m - 4\ln\varepsilon).$$

The expected number of successful delivery rounds is then at least

$$\mu_s = Rp_s = 2\lg m - 4\ln\varepsilon.$$

We choose

$$\alpha = \frac{1}{p_s}(\lg m - 4\ln\varepsilon)$$

and apply the Chernoff bound to random variable X_s, which denotes the number of successful delivery rounds:

$$\begin{aligned}
\Pr[X_s < \mu_s - \alpha p_s] &< e^{-\frac{(\alpha p_s)^2}{2\mu_s}} \\
\Leftrightarrow \Pr[X_s < 2\lg m - 4\ln\varepsilon - \lg m + 4\ln\varepsilon] &< e^{-\frac{(\alpha p_s)^2}{4\alpha p_s + 8\ln\varepsilon}} \\
\Rightarrow \quad \Pr[X_s < \lg m] &\leqslant e^{-\frac{\alpha p_s}{4}} \\
&= e^{-\frac{\lg m}{4} + \ln\varepsilon} \\
&\leqslant \varepsilon \\
\Leftrightarrow \quad \Pr[X_s > \lg m] &\geqslant 1 - \varepsilon.
\end{aligned}$$

We have established that the number of successful delivery rounds X_s within $R = O(\lg m + \ln(1/\varepsilon))$ delivery rounds is larger than $\lg m$ with probability at least $1 - \varepsilon$. Because it takes at most $\lg m$ successful delivery rounds to deliver m messages, the number of delivery rounds needed to deliver $m \leqslant 2n/\lg n$ messages is $O(\lg m + \ln(1/\varepsilon))$ with probability at least $1 - \varepsilon$ for any $\varepsilon > 0$. This argument completes the proof for Phase II. □

7. Discussion of result

To bound the number of delivery rounds in a fat-tree network, we have resorted to a proof methodology where we developed an approximating model of the collision behavior of messages that is amenable to rigorous probabilistic analysis. Our balls-and-bins model is not powerful enough to derive statements about the *micro behavior* of the network, for example about the number of collisions at a particular router. However, we may claim the validity of our proof if we can show that our model reflects reality at the level of delivery rounds. To that end, we provide empirical evidence that the simple balls-and-bins Model I does capture the collision behavior with sufficient accuracy, indeed.

Fig. 4 compares three data sets of simulation results for the number of messages $m = n/8$ on the left-hand side and for $m = n$ on the right-hand side. These graphs are representative for a large number of values of m that we have simulated. Comparison of the number of delivery rounds according to Model I with those for round-based fat-tree simulations demonstrates the validity of Model I. For the maximum number of messages $m = n$ that can be in transit during a single round, Model I shows the largest deviation from the fat-tree simulations. However, the number of delivery rounds predicted by Model I and the round-based fat-tree simulation differ by a small constant factor only, analogous to our observation in Fig. 3.

In addition to the round-based simulation results, we show the normalized number of clock cycles for delivering m messages on a fat tree in Fig. 4. These results represent the true performance of our fat-tree design under the assumption that the transmission of $m = n/8$ or n messages starts at the same clock cycle, and that the network interfaces initiate a retransmission of a rejected message one clock cycle after sensing a collision. Thus, these simulations drop the simplifying assumption that retransmission occurs in rounds on all network interfaces simultaneously. For a direct comparison with the round-based simulations, we normalize the number of clock cycles with respect to the transmission time of a single message across the diameter of an n-node fat tree measured in clock cycles. We conclude from these results that our $O(\lg m)$ bound holds for the scenario with immediate retry as well. Our round-based model and simulations are conservative by a constant factor of about two on average. We report that immediate retry delivers the highest performance on our fat-tree architecture compared to other retry strategies, including exponential back-off.

We have developed the fat-tree simulator as a behavioral model of a gate-level implementation in order to scale up to 2^{20} processing nodes. Our router design has a latency of two clock cycles for an advancing message, which includes the path reservation, and one clock cycle for a collision and acknowledgment signal to release and traverse the path in the opposite direction, respectively. We have implemented various retry strategies in our network interfaces, including round-based retry, where all messages transmitted during one delivery round are either delivered or rejected before the rejected messages are retransmitted in the subsequent delivery round. This retry strategy requires a global synchronization capability, and is not expected to be implemented in real systems. However, it allows for a direct comparison with the simulation results of the balls-and-bins model.

Whereas Fig. 4 shows the number of delivery rounds as a function of n for fixed m, Fig. 5 provides a view on the number of delivery rounds as a function of m for fixed n. Like the graphs in Fig. 4, these graphs are representative for a large number of experiments for different values of n. To avoid clutter, we omit the normalized transmission times for the immediate retry strategy. The graphs in Fig. 5 exhibit a number of behavioral details of the fat-tree network that deserve further discussion:

(1) The balls-and-bins model matches the number of delivery rounds due to the fat-tree simulation accurately, in accordance with the results presented in Fig. 4.
(2) The error bars in the plots show the variation of the number of delivery rounds due to the randomized routing strategy in the fat tree. The fact that the variation is relatively small validates our high-probability result.
(3) Our bound $O(\lg m) = c_1 \lg m + c_2$ appears as a straight line in the semi-logarithmic plots of Fig. 5 for $c_1 = 1$ and $c_2 = 0$. At the first glance, even this bound seems to be conservative, although it is significantly tighter than $O(\lg n)$.
(4) The vertical lines in the plots of Fig. 5 represent the boundary between Phase I and Phase II of our proof at $m = 2n/\lg n$.
(5) Recall that the number of delivery rounds in Phase II is $O(\lg m)$ for $m \leqslant 2n/\lg n$. Indeed, we observe this behavior to the left of the vertical line at $m = 2n/\lg n$. The constant factor c_1 in $O(\lg m) = c_1 \lg m + c_2$ is obviously much smaller than 1, as a comparison with the straight line for $\lg m$ reveals.
(6) During Phase I of our proof for $m > 2n/\lg n$, we made use of the inequality $m/\lg m \leqslant n/\lg n$ for $2 \leqslant m \leqslant n$ to bound the number of successful delivery rounds in Eqs. (6) and (7) of Section 6.1. In fact, both Model I and the fat-tree simulations exhibit the behavior

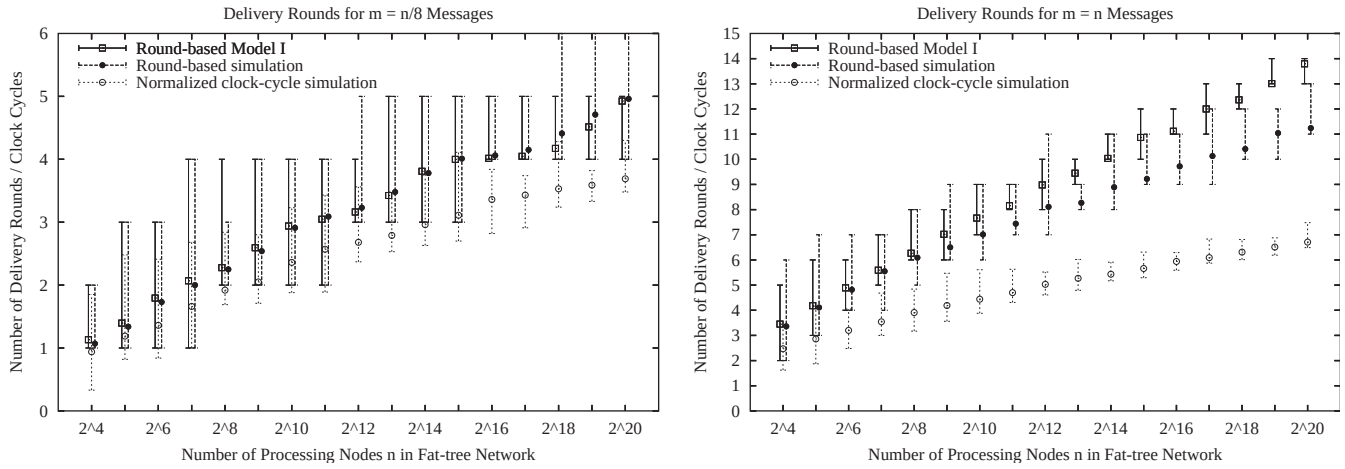

Fig. 4. Comparison of balls-and-bins Model I with round-based fat-tree simulations for $m = n/8$ and n. These graphs are representative for other values of $0 < m \leqslant n$. Model I differs from the round-based fat-tree simulations by a small constant factor only. The normalized performance of a real fat tree with immediate retry is shown in clock cycles with respect to the transmission time of one message across the diameter of an n-node fat tree.

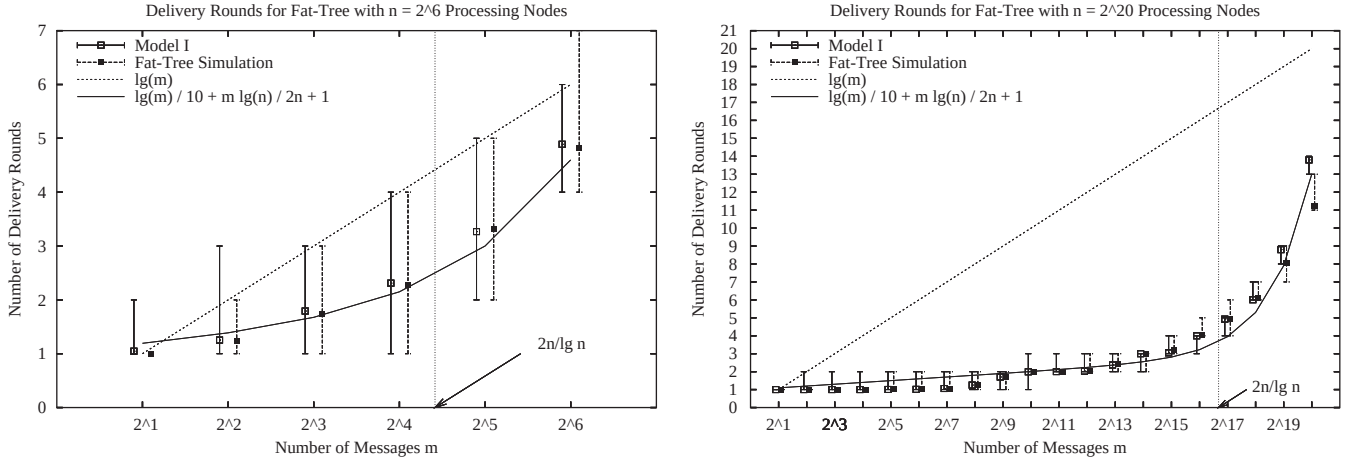

Fig. 5. Comparison of balls-and-bins Model I with round-based fat-tree simulations for $n = 64$ and $n = 2^{20}$. These graphs are representative for other values of $n \leqslant 2^{20}$.

of the tighter bound $m \lg n/n \leqslant \lg m$ for the number of delivery rounds, as we observe to the right of the vertical line at $m = 2n/\lg n$.

(7) Fig. 5 includes an ad hoc curve fit of the number of delivery rounds as a superposition of the models for Phase I and Phase II. For Phase I, we use $m \lg n/2n$, and $\lg m/10 + 1$ for Phase II. The sum of both phases yields the curve displayed in Fig. 5: $\lg m/10 + m \lg n/2n + 1$.

The simulation results in Fig. 5 suggest that $O(\lg m)$ is in fact the optimal bound for Phase II. For Phase I, $O(\lg m)$ is an upper bound of the observed behavior which follows the tighter bound $O(m \lg n/n)$. Consequently, the simulation results provide experimental evidence for our claim that balls-and-bins Model I reflects reality with sufficient accuracy, and that the upper bound for the number of delivery rounds is indeed $O(\lg m)$, independent of number of processing nodes n of the fat tree, and independent of the operational phase determined by the number of messages m.

We have limited our proof of the time bound to the communication scenario, where distinct message sources are chosen randomly and destinations are chosen randomly and independently. The rationale behind this choice has been the feasibility of probabilistic analysis. In practice, many communication patterns can be approximated by this assumption. For other communication patterns this choice appears to be unreasonable. For example, assume that each of m sources sends one message to a single destination node p. Since p may receive one message at a time only, a lower bound for the number of delivery rounds is m. Because the fat-tree architecture guarantees that one mes-

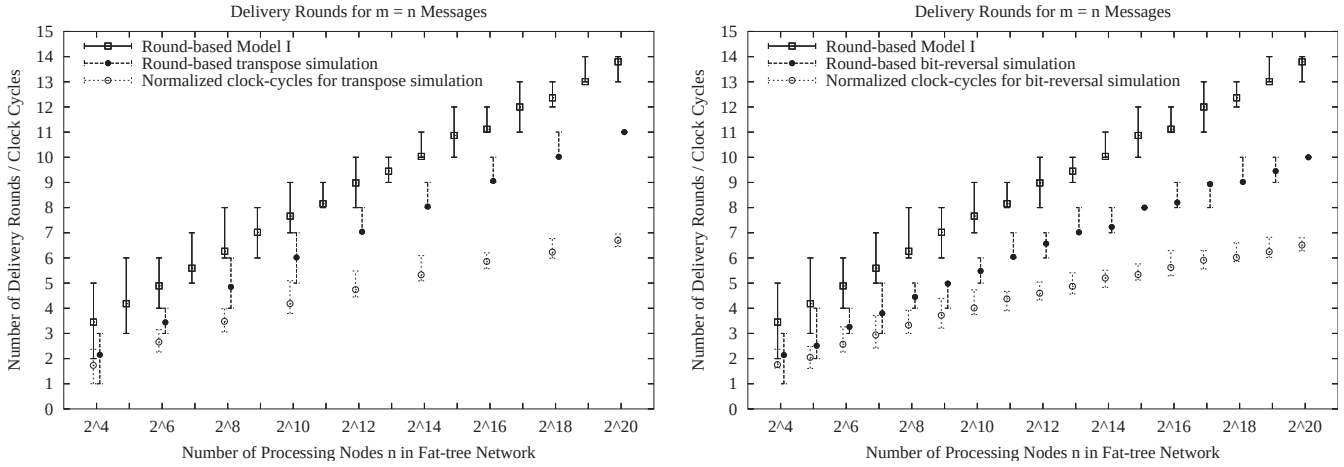

Fig. 6. Comparison of the number of delivery rounds for randomly chosen sources and destinations according to Model I (cf. Fig. 4), a transpose permutation (left), and a bit-reversal permutation (right) with $m = n$ messages. Our simulations show that the number of delivery rounds for the permutations with $m = n$ are strictly larger than for smaller numbers of messages $m < n$. The normalized clock-cycle counts show the real behavior of a fat tree with immediate message retry.

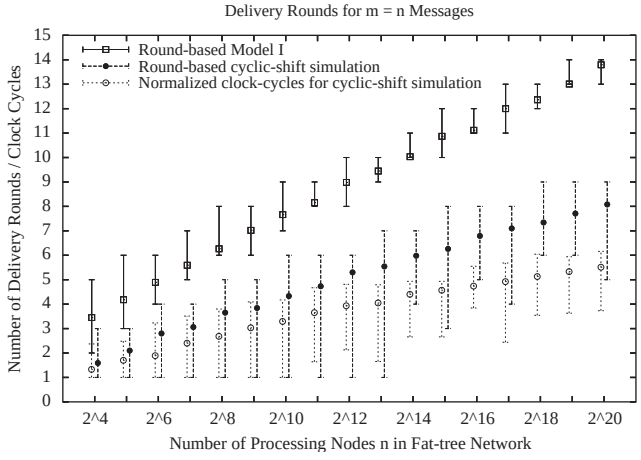

Fig. 7. Comparison of the number of delivery rounds for randomly chosen sources and destinations according to Model I and a cyclic-shift permutation with $m = n$ messages. Our simulations show that the number of delivery rounds for the cyclic-shift permutation with $m = n$ are strictly larger than for smaller numbers of messages $m < n$. The normalized clock-cycle counts show the real behavior of a fat tree with immediate message retry.

sage will be delivered during a round, m is also the upper bound. This extreme case leads us to the following conjecture about the number of delivery rounds of any communication pattern on a fat tree. Assume that $m = m_1 + m_r$, where m_1 is the number of messages to be transmitted or delivered sequentially, and m_r is the number of messages whose sources and destinations can be approximated by a random distribution. Then, the number of delivery rounds is bound by

$$O(m_1 + \lg m_r).$$

In the extreme case where $m = m_1$, the number of delivery rounds is bound by $O(m_1)$. In the other extreme case where $m = m_r$, the number of delivery rounds is bound by $O(\lg m_r)$ as we have proved in Section 6. We may view any case between these extremes as a superposition of a sequential component consisting of m_1 messages and a parallel component consisting of m_r messages with bound $O(m_1 + \lg m_r)$ for the number of delivery rounds.

We also ran experiments to evaluate the performance of the fat-tree for several regular communication patterns that are frequently studied in the routing literature [11]. Let $p_1 \dots p_{\lg n}$ denote the binary representation of node p.

Cyclic-shift permutation: For a given shift k, node p sends one message to node $q = (p + k)\%n$. This pattern arises for example in stencil computations over a grid.

Transpose permutation: The node with binary representation $p_1 \dots p_{(\lg n)/2} p_{(\lg n)/2+1} \dots p_{\lg n}$ sends one message to node $p_{(\lg n)/2+1} \dots p_{\lg n} p_1 \dots p_{(\lg n)/2}$. The primary application using this pattern is a matrix transposition.

Bit-reversal permutation: Node $p_1 \dots p_{\lg n}$ sends one message to node $p_{\lg n} \dots p_1$. This pattern, as well as the transpose permutation, is considered traditionally a worst-case routing problem.

Our simulations indicate that the average number of delivery rounds for each of these communication patterns is bounded by $O(\lg m)$ for $0 < m \leqslant n$. Figs. 6 and 7 show the simulation results for $m = n$ messages. Although we do not present the corresponding graphs, we report that the number of delivery rounds of the permutations for $m < n$ is strictly less than those shown in the figures. For the transpose permutation on the left-hand side of Fig. 6, we generated data points only for those cases where the number of processing nodes n is a square. The normalized clock-cycle counts show

the behavior of a fat-tree network with immediate message retry after a collision. The real performance of a fat tree is on average about a factor of two faster than predicted by the round-based model.

The results of the cyclic-shift permutation in Fig. 7 are more comprehensive than those in Fig. 6. Since the number of delivery rounds depends on the shift parameter k, we present as the average number of delivery rounds the average of the delivery rounds of the average over a range of shift parameters $0 < k < n$. The minimum and maximum number of delivery rounds of each error bar represent the corresponding values for all values of k.

For all three communication patterns, the number of delivery rounds is strictly less than the average number of rounds required when destination nodes are chosen randomly. These simulation results suggest that $O(\lg m)$ is the optimal bound not only for randomly chosen sources and destinations but for many different communication patterns on the fat-tree network.

8. Conclusion

We have shown that $m \leqslant n$ messages with randomly chosen, distinct sources and independently and randomly chosen destinations are delivered in a fat-tree network with n processing nodes within $O(\lg m)$ delivery rounds with high probability. Our proof methodology is based on an approximating collision model of the messages transmitted into the network. This model constitutes a tradeoff between simplicity and accuracy. It facilitates a relatively simple probabilistic analysis and reflects reality with sufficient accuracy at the same time. Whether our proof methodology is applicable to leaner fat trees or even entirely different network architectures remains an open question. We have presented empirical evidence to validate our claim that $O(\lg m)$ is a tight upper bound for the delivery of messages not only under the simplifying assumptions that enable our analysis, but also for practical implementations and communication scenarios on a fat-tree network.

Acknowledgments

We thank Bradley Kuszmaul for sharing his insights into the implementation of fat-tree networks, and Charles Leiserson for valuable discussions about our proof methodology.

References

[1] R. Cole, B.M. Maggs, Friedhelm Meyer auf der Heide, M. Mitzenmacher, A.W. Richa, K. Schröder, R.K. Sitaraman, B. Vöcking, Randomized protocols for low-congestion circuit routing in multistage interconnection networks, in: 30th Annual Symposium on Theory of Computing, ACM, Dallas, TX, 1998, pp. 378–388.

[2] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, The MIT Press, Cambridge, MA, 1990.

[3] A. DeHon, Fat-tree routing on transit, Master's Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1990.

[4] R.I. Greenberg, C.E. Leiserson, Randomized routing on fat-trees, in: 26th Annual IEEE Symposium on Foundations of Computer Science, 1985, pp. 241–249.

[5] V. Gupta, E. Schenfeld, Performance analysis of a synchronous, circuit-switched interconnection cached network, in: Eighth International Conference on Supercomputing, ACM Press, Manchester, England, 1994, pp. 246–255.

[6] R.R. Koch, Increasing the size of a network by a constant factor can increase performance by more than a constant factor, in: 29th Annual Symposium on Foundations of Computer Science, IEEE, White Plains, NY, 1988, pp. 221–230.

[7] T.F. Knight, Jr., P.G. Sobalvarro, Routing statistics for unqueued banyan networks, Tech. Rep. A.I. Memo 1101, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, September 1990.

[8] C.P. Kruskal, M. Snir, The performance of multistage interconnection networks for multiprocessors, IEEE Trans. Comput. C-32 (12) (1983) 1091–1098.

[10] E.D. Lazowska, J. Zahorjan, G.S. Graham, K.C. Sevcik, Quantitative System Performance: Computer System Analysis Using Queueing Network Models, Prentice-Hall, Englewood Cliffs, NJ, 1984.

[11] F.T. Leighton, Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes, Morgan Kaufmann, Los Altos, CA, 1992.

[12] F.T. Leighton, B.M. Maggs, A.G. Ranade, S.B. Rao, Randomized routing and sorting on fixed-connection networks, J. Algorithms 17 (1994) 157–205.

[13] C.E. Leiserson, Fat-trees: universal networks for hardware-efficient supercomputing, IEEE Trans. Comput. C-34 (10) (1985) 892–901.

[14] C.E. Leiserson, Z.S. Abuhamdeh, D.C. Douglas, C.R. Feynman, M.N. Ganmukhi, J.V. Hill, W.D. Hillis, B.C. Kuszmaul, M.A.S. Pierre, D.S. Wells, M.C. Wong-Chan, S.-W. Yang, R. Zak, The network architecture of the connection machine CM-5, J. Parallel Distrib. Comput. 33 (2) (1996) 145–158.

[15] B.M. Maggs, R.K. Sitaraman, Simple algorithms for routing on butterfly networks with bounded queues, in: 24th Annual Symposium on Theory of Computing, ACM, Victoria, BC, Canada, 1992, pp. 150–161.

[16] M. Mitzenmacher, A.W. Richa, R. Sitaraman, The power of two random choices: a survey of the techniques and results, in: S. Rajasekaran, P.M. Pardalos, J.H. Reif, J.D. Rolim (Eds.), Handbook of Randomized Computing, vol. I, Kluwer Press, 2001, pp. 255–305.

[17] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, Cambridge, 1995.

[18] D. Nussbaum, I. Vuong-Adlerberg, A. Agarwal, Modeling a circuit-switched multiprocessor interconnect, ACM SIGMETRICS Performance Evaluation Review 18 (1) (1990) 267–269.

[19] J.H. Patel, Processor-memory interconnections for multiprocessors, in: Sixth Annual Symposium on Computer Architecture, ACM Press, New York, 1979, pp. 168–177.

[20] M. Schwartz, Telecommunication Networks: Protocols, Modeling and Analysis, Addison-Wesley, Reading, MA, 1987.

[21] D.D. Sleator, R.E. Tarjan, Amortized efficiency of list update and paging rules, Commun. ACM 28 (2) (1985) 202–208.

[22] E. Waingold, M. Taylor, D. Srikrishna, V. Sarkar, W. Lee, V. Lee, J. Kim, M. Frank, P. Finch, R. Barua, J. Babb, S. Amarasinghe, A. Agarwal, Baring it all to software: raw machines, IEEE Comput. (1997) 86–93.

[23] C.-L. Wu, M. Lee, Performance analysis of multistage interconnection network configurations and operations, IEEE Trans. Comput. 41 (1) (1992) 18–27.

Volker Strumpen is currently a Research Staff Member at IBM Austin Research Laboratory. This work has been conducted while he was a Research Scientist at the Computer Science and Artificial Intelligence Lab of the Massachusetts Institute of Technology. In the past, he has served in academic positions at the University of Iowa, the Massachusetts Institute of Technology and Yale University. He was also affiliated with Sony in Atsugi, Japan, and helped starting up Akamai Technologies. Strumpen received a Diploma in electrical engineering from RWTH Aachen in Germany and a PhD in Computer Science from ETH Zurich, Switzerland.

Arvind Krishnamurthy is an Assistant Professor at Yale University. His research interests are primarily at the boundary between the theory and practice of distributed systems. His dissertation concerned programming language and compiler support for parallel programs. Since then he has worked in mechanism design/game theory applied to computer networks, techniques to make RAID's low latency devices, distributed storage systems that integrate the numerous ad hoc devices around the home, byzantine routing, automated mechanisms for managing overlay networks and distributed hash tables, and technologies for the third world.