LABORATORY FOR COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

A Collision Model For Randomized Routing In Fat-Tree Networks

Technical Memo
MIT-LCS-TM-629
July 15, 2002

Volker Strumpen
strumpen@lcs.mit.edu

Arvind Krishnamurthy
arvind@cs.yale.edu

This manuscript describes research pursued at the Laboratory for Computer Science of the Massachusetts Institute of Technology and Yale University.

545 TECHNOLOGY SQUARE, CAMBRIDGE, MASSACHUSETTS 02139

A Collision Model For Randomized Routing In Fat-Tree Networks

Volker Strumpen
Laboratory for Computer Science
Massachusetts Institute of Technology
strumpen@lcs.mit.edu

Arvind Krishnamurthy
Department of Computer Science
Yale University
arvind@cs.yale.edu

July 15, 2002

Abstract

We present a proof that in a fat-tree network with n processing nodes $m \leq n$ 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 $O(\lg m + \ln 1/\epsilon)$ delivery rounds with probability $1 - \epsilon$ for any small $\epsilon > 0$. 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.

1 Introduction

Fat-tree networks are established as area-universal communication networks due to the seminal work of Charles E. Leiserson [8, 3], culminating in the implementation of the Connection Machine CM-5 at Thinking Machines Corporation [9]. Today, advances in semiconductor technology enable us to integrate multiprocessor machines on a single chip, as explored in the Raw project [12], for example. As the number of processors on a chip increases, employing one or more fat-tree networks as interconnection medium is 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 $m \leq n$ messages with randomly chosen 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 [8] derived a bound using the *load factor* λ of a set of messages. He 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 [3] 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 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 [3, 1, 4, 5, 6, 7, 10], suggests that an exact analysis requires significant theoretical armory. While most of this work tackles more general routing problems, we are not aware of any approaches with a goal similar to ours.

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. 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 $m \leq b$ 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 $O(\lg m + \ln 1/\epsilon)$ delivery rounds with probability $1 - \epsilon$ for any small $\epsilon > 0$.

3 Fat-Tree Architecture

Our proof is restricted to the architecture of the fat-tree network shown in Figure 1 with the router design described below.[1] Whether our proof methodology is applicable to leaner trees or even entirely different network architectures remains an open question.

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 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

[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. The fat-tree network is relatively easy to implement and realizable with today's and future micro-technology, which offers six or more levels of metal.

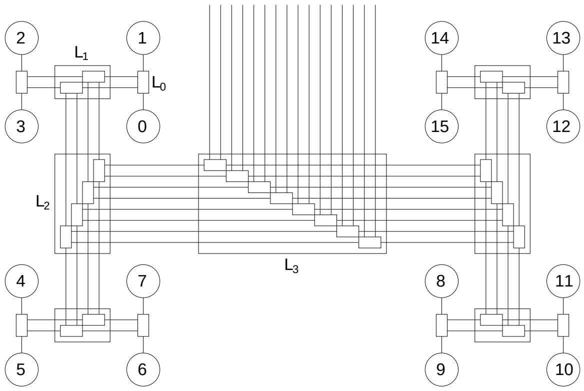


Figure 1: Structure of fat-tree network with 16 processing nodes.

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 Figure 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 Figure 2. We call ports a and b the ***downstream ports***, and ports c and d the ***upstream ports*** as obvious from Figure 1.

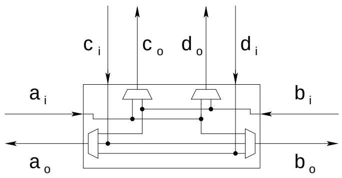


Figure 2: Router design of a full-duplex fat tree.

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 randomness in the routing process. 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 theses 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 a leaf node of the tree, connecting two processing nodes. A ***router node*** at level l in the fat tree consists of 2^l individual L_l-routers. In Figure 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.

4 Structural Analysis

We analyze the collision behavior of a fat-tree network to compute the probability of a collision between two messages.

Lemma 1 (2-Message Collision Probability) *Two messages with randomly chosen, distinct sources and independently and randomly chosen destinations collide on average with probability*

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

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

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

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 ν-subtrees with 2^ν 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 ν-subtree have the same dilation $2(\nu + 1)$, counted in number of links, from the respective reference node.

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. Figure 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 $(i = \nu)$-subtree is the set of nodes $\{2^\nu, \ldots, 2^{\nu+1} - 1\}$.

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 $\Pr[C_2]$:

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

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 $0 \le i \le \lg n - 1$, and for $0 \le j \le \lg n - 1$:

$$\Pr[k,i,j] \quad = \quad \begin{array}{cc} & \begin{array}{ccccc} 0 & \cdots & j=k & \cdots & \lg\frac{n}{2} \end{array} \\ \begin{array}{c} 0 \\ \\ \vdots \\ i=k \\ \\ \vdots \\ \lg\frac{n}{2} \end{array} & \left(\begin{array}{ccccc} & & \vdots & & \\ & 0 & \frac{i+2}{2^{k+1}} & & \\ & & \vdots & 0 & \\ \cdots & \frac{j+2}{2^{k+1}} \ \cdots & 0 & & \\ & & & \ddots & \\ & & 0 & \frac{k+2}{2^{i+1}} & \\ & & & & \ddots \end{array} \right) \end{array} \tag{2}$$

We can derive Equation 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 Figure 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 Figure 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 non-zero elements in row $i = k$ and column $j = k$ of the matrix in Equation 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-subtree 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 $0 \leq r < j$. 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_ν-routers for $0 \leq \nu \leq r$. 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 $0 \leq r < j$. Therefore, we obtain:

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

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 Equation 2 for row $i = k$. The column elements for $i < j = k$ follow by symmetry.

$i = j > k$: This case corresponds to the non-zero elements on the main diagonal of the matrix in Equation 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 $0 \leq r < j$. 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 Figure 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 $2^2 = 4$ 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 $0 \le r < k$. 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

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

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

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

$$\begin{aligned}
\sum_{i=0}^{\lg n-1} 2^i \sum_{j=0}^{\lg n-1} 2^j \Pr[k, i, j] &= \sum_{i=0}^{k-1} 2^{i+k} \frac{i+2}{2^{k+1}} + \sum_{j=0}^{k-1} 2^{j+k} \frac{j+2}{2^{k+1}} + \sum_{\nu=k+1}^{\lg n-1} 2^{2\nu} \frac{k+2}{2^{\nu+1}} \\
&= \sum_{i=0}^{k-1} (i+2) \cdot 2^i + \sum_{\nu=k+1}^{\lg n-1} \frac{k+2}{2} \cdot 2^{\nu}
\end{aligned}$$

$$= \quad k \cdot 2^k + \frac{k+2}{2}\left(n - 2^{k+1}\right).$$

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

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

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. Since each source can transmit only one message at a time, n is an upper bound for m, and we have $2 < m \leq n$. 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.

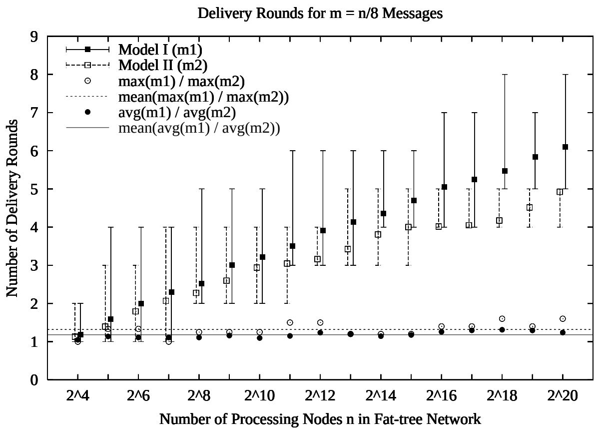
Corollary 1 (Bin Calibration) *The number of bins b of the balls-and-bins game modeling the collision probability in the full-duplex fat tree is $2n/\lg n$.*

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

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

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



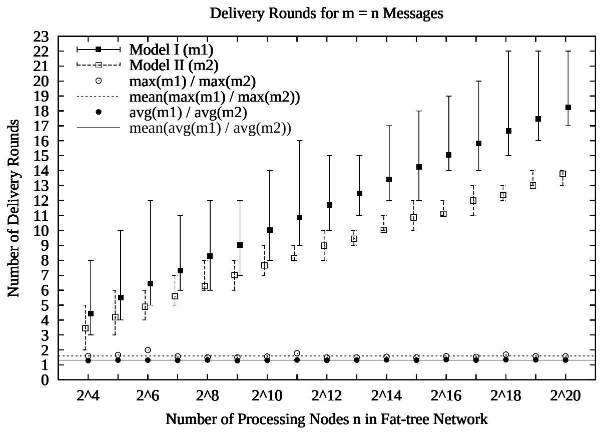


Figure 3: Simulations of Model I including a destination toss and Model II without the destination toss for $m = n/8$ (left) and $m = n$ (right). The number of delivery rounds due to the two models differ by a small constant factor only.

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 collision 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. 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 the model with the destination toss ***Model I*** and our original, simpler balls-and-bins game ***Model II***. Simulations show that the number of delivery rounds due to these two models differ by a constant factor only. Figure 3 shows the number of delivery rounds for $2^4 \leq n \leq 2^{20}$ processing nodes and for the number of messages $m = n/8$ and $m = n$. 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 [2], we can construct a proof that considers the dependencies of the destination distribution expressed by Model I. 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 II 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 II only.

We now turn our attention to results from basic probability theory about the balls-and-bins game underlying Model II. 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

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

since $1 + x \leq e^x$ 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

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

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

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

and the expected number of rejected messages is

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

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 II developed in Section 5. Each of the $m \leq n$ 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 II of a fat-tree network with n processing nodes, $m \leq n$ messages with randomly chosen, distinct sources and independently and randomly chosen destinations are delivered within $O(\lg m + \ln 1/\epsilon)$ delivery rounds with probability $1 - \epsilon$ for any $\epsilon > 0$.*

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 $O(\lg m + \ln 1/\epsilon)$ rounds with probability $1 - \epsilon$. In phase II we prove that $m \leq 2n/\lg n$ messages are delivered within $O(\lg m + \ln 1/\epsilon)$ rounds with probability $1 - \epsilon$. 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 result in more than one ball landing in one or more bins. For $m \leq n$, the number of balls landing in each bin will 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*** [11]. It states that for a non-negative random variable X and any positive real t we have

$$\Pr[X \geq t] \leq \frac{E[X]}{t}.$$

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 Equation 3

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

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

$$\Pr[b_e \geq 2b/e] \;\leq\; \frac{E[b_e]}{2b/e} \;\leq\; \frac{1}{2}.$$

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*:

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

Solving for S, we obtain:

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

since $m/\lg m \leq n/\lg n$ for $2 \leq m \leq n$. 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*** [11]: 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 $0 < \delta \leq 1$, 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

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

for a small value ϵ. This magic 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

$$\mu_s = R \cdot p_s = 2S - 4\ln\epsilon,$$

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

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

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

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

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

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

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

$$\begin{array}{rcl}
\Pr[X_s < \mu_s - \alpha p_s] & < & e^{-\frac{(\alpha p_s)^2}{2\mu_s}} \\
\Leftrightarrow \Pr[X_s < 2S - 4\ln\epsilon - S + 4\ln\epsilon] & < & e^{-\frac{(\alpha p_s)^2}{4\alpha p_s + 8\ln\epsilon}} \\
\Rightarrow \qquad \Pr[X_s < S] & \leq & e^{-\frac{\alpha p_s}{4}} \\
 & = & e^{-S/4+\ln\epsilon} \\
 & \leq & \epsilon \\
\Leftrightarrow \qquad \Pr[X_s > S] & \geq & 1-\epsilon.
\end{array}$$

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

6.2 Analysis of Phase II

During phase II we inject $m \leq 2n/\lg n$ messages into the network. Correspondingly, in our balls-and-bins model, we toss $m \leq 2n/\lg n$ 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 Equation 5, the expected number of rejected messages is $E[R] = m - E[D] \leq m - b\left(1 - e^{-\frac{m}{b}}\right)$. 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 $2 \leq m \leq b$:

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

Note that $f(x) = x(1-e^{-1/x}) \geq 1 - e^{-1}$ for $x \geq 1$, 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:

$$R \geq (1-\alpha)m \quad \Leftrightarrow \quad D \leq m - (1-\alpha)m = \alpha m.$$

We substitute this term in the Markov Inequality to obtain:

$$\begin{aligned}
\Pr[R \geq (1-\alpha)m] &\leq \frac{1}{(1-\alpha)e} \\
\Leftrightarrow \qquad \Pr[D \leq \alpha m] &\leq \frac{1}{(1-\alpha)e} \\
\Leftrightarrow \qquad \Pr[D \geq \alpha m] &\geq 1 - \frac{1}{(1-\alpha)e}.
\end{aligned}$$

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

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

The expected number of successful delivery rounds is then at least

$$\mu_s = R \cdot p_s = 2\lg m - 4\ln\epsilon.$$

We choose

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

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

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

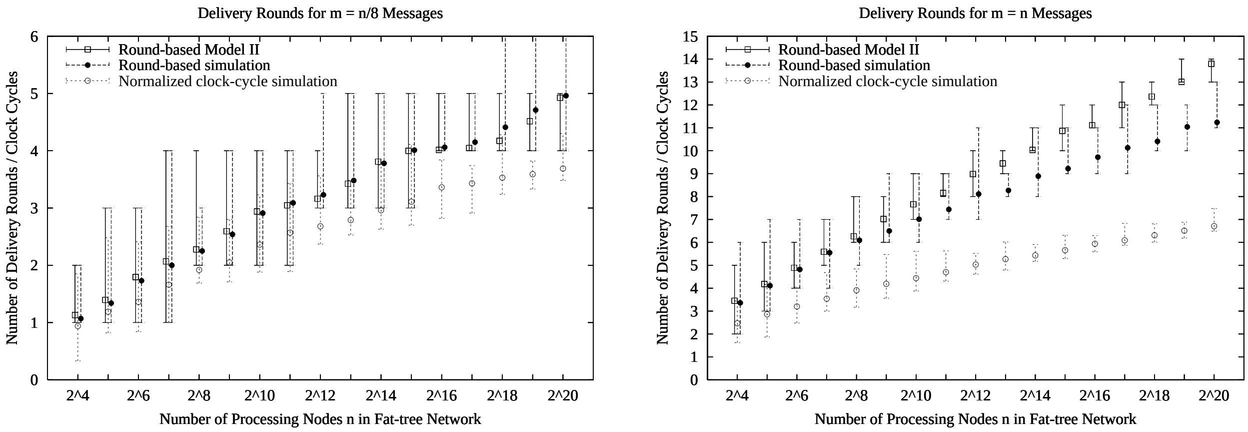


Figure 4: Comparison of balls-and-bins model II with round-based fat-tree simulations for $m = n/8$ and $m = n$. These graphs are representative for other values of $0 < m \leq n$. Model II differs from the round-based fat-tree simulations by a small constant factor only. The normalized performance of 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.

We have established that the number of successful delivery rounds X_s within $R = O(\lg m + \ln(1/\epsilon))$ delivery rounds is larger than $\lg m$ with probability at least $1 - \epsilon$. Because it takes at most $\lg m$ successful delivery rounds to deliver m messages, the number of delivery rounds needed to deliver $m \leq 2n/\lg n$ messages is $O(\lg m + \ln(1/\epsilon))$ with probability at least $1 - \epsilon$ for any $\epsilon > 0$. 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 II does capture the collision behavior with sufficient accuracy, indeed.

Figure 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 II with those for round-based fat-tree simulations demonstrates the validity of Model II. For the maximum number of messages $m = n$ that can be in transit during a single round, Model II shows the largest deviation from the fat-tree simulations. However, the number of delivery rounds predicted by Model II and the round-based fat-tree simulation differ by a small constant factor only, analogous to our observation in Figure 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 Figure 4. These results represent the true performance of our fat-tree design under the assumption that the transmission of $m = n/8$ or $m = n$ messages

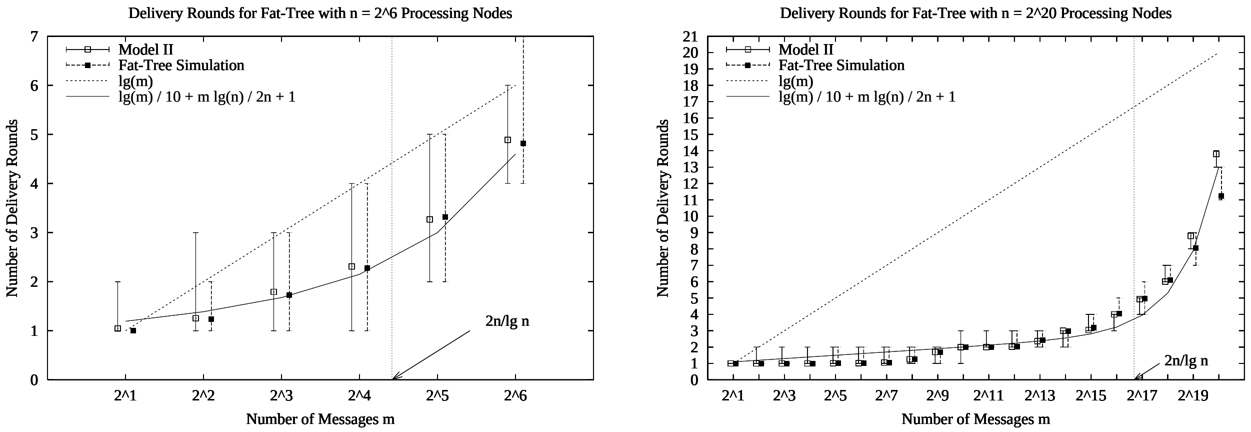


Figure 5: Comparison of balls-and-bins model II with round-based fat-tree simulations for $n = 64$ and $n = 2^{20}$. These graphs are representative for other values of $n \leq 2^{20}$.

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 Figure 4 shows the number of delivery rounds as a function of n for fixed m, Figure 5 provides a view on the number of delivery rounds as a function of m for fixed n. Like the graphs in Figure 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 Figure 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 Figure 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 $O(\lg m) = c_1 \cdot \lg m + c_2$ appears as a straight line in the semi-logarithmic plots of Figure 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 Figure 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 $m \leq 2n/\lg n$. 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 $m/\lg m \leq n/\lg n$ for $2 \leq m \leq n$ to bound the number of successful delivery rounds in Equations 6 and 7 of Section 6.1. In fact, both Model II and the fat-tree simulations exhibit the behavior of the tighter bound $m \lg n/n \leq \lg m$ for the number of delivery rounds, as we observe to the right of the vertical line at $m = 2n/\lg n$.

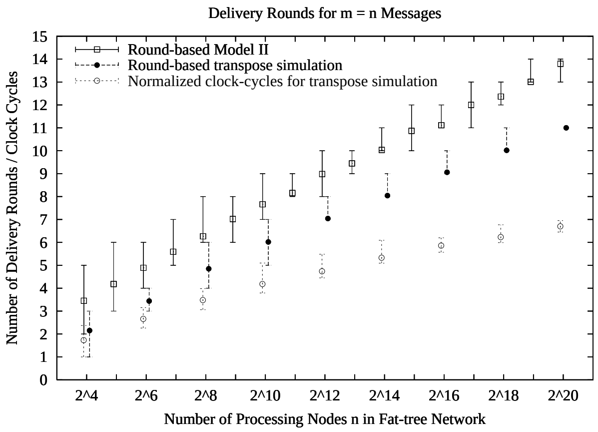
7. Figure 5 includes an ad-hoc curve fit of the number of delivery rounds as a superposition of the models for phase I and phase II. For phase I, we use $m \lg n/2n$, and $\lg m/10 + 1$ for phase II. The sum of both phases yields the curve displayed in Figure 5: $\lg m/10 + m \lg n/2n + 1$.

The simulation results in Figure 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 II 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 message 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



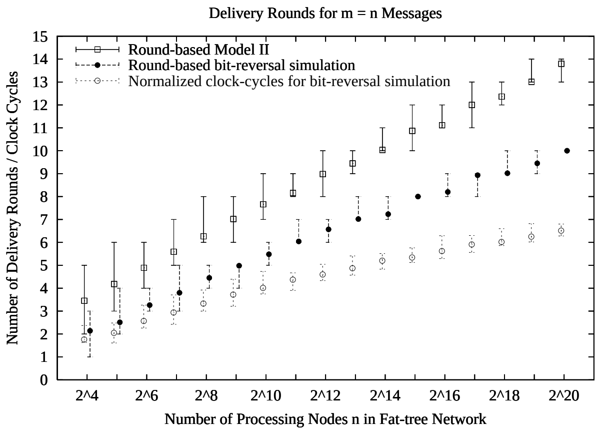


Figure 6: Comparison of the number of delivery rounds for randomly chosen sources and destinations according to Model II (cf. Figure 4), a transpose permutation (left), and a bit-reversal permutation (right) with $m = n$ messages. Our simulations show that the number of delivery rounds for the permutations 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.

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 [6]. Let $p_1 \ldots p_{\lg n}$ 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 $p_1 \ldots p_{(\lg n)/2} p_{(\lg n)/2+1} \ldots p_{\lg n}$ sends one message to node $p_{(\lg n)/2+1} \ldots p_{\lg n} p_1 \ldots p_{(\lg n)/2}$. The primary application using this pattern is a matrix transposition.

Bit-reversal permutation: Node $p_1 \ldots p_{\lg n}$ sends one message to node $p_{\lg n} \ldots p_1$. 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 $0 < m \leq n$. Figures 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 Figure 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 Figure 7 are more comprehensive than those in Figure 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

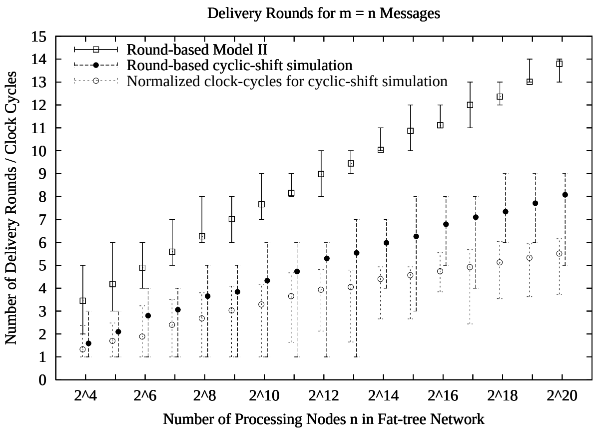


Figure 7: Comparison of the number of delivery rounds for randomly chosen sources and destinations according to Model II and a cyclic-shift permutation with $m = n$ messages. Our simulations show that the number of delivery rounds for the cyclic-shift permutation with $m = n$ are strictly larger than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

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 $m \leq n$ 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. 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.

9 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] Richard Cole, Bruce M. Maggs, Friedhelm Meyer auf der Heide, Michael Mitzenmacher, Andréa W. Richa, Klaus Schröder, Ramesh K. Sitaraman, and Berthold Vöcking. Randomized Protocols for Low-Congestion Circuit Routing in Multistage Interconnection Networks. In *30th Annual Symposium on Theory of Computing*, pages 378–388, Dallas, TX, May 1998. ACM.

[2] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms.* The MIT Press, Cambridge, Massachusetts, 1990.

[3] Ronald I. Greenberg and Charles E. Leiserson. Randomized Routing on Fat-Trees. In *26th Annual IEEE Symposium on Foundations of Computer Science*, pages 241–249, October 1985.

[4] Richard 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*, pages 221–230, White Plains, NY, October 1988. IEEE.

[5] Clyde P. Kruskal and Marc Snir. The Performance of Multistage Interconnection Networks for Multiprocessors. *IEEE Transactions on Computers*, C-32(12):1091–1098, December 1983.

[6] F. Thomson Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes.* Morgan Kaufmann, 1992.

[7] F. Thomson Leighton, Bruce M. Maggs, Abhiram G. Ranade, and Satish B. Rao. Randomized Routing and Sorting on Fixed-Connection Networks. *Journal of Algorithms*, 17:157–205, 1994.

[8] Charles E. Leiserson. Fat-Trees: Universal Networks For Hardware-Efficient Supercomputing. *IEEE Transactions on Computers*, C-34(10):892–901, October 1985.

[9] Charles E. Leiserson, Zahi S. Abuhamdeh, David C. Douglas, Carl R. Feynman, Mahesh N. Ganmukhi, Jeffrey V. Hill, W. Daniel Hillis, Bradley C. Kuszmaul, Margaret A. St. Pierre, David S. Wells, Monica C. Wong-Chan, Shaw-Wen Yang, and Robert Zak. The Network Architecture of the Connection Machine CM-5. *Journal of Parallel and Distributed Computing*, 33(2):145–158, March 1996.

[10] Bruce M. Maggs and Ramesh K. Sitaraman. Simple Algorithms for Routing on Butterfly Networks with Bounded Queues. In *24th Annual Symposium on Theory of Computing*, pages 150–161, Victoria, BC, Canada, May 1992. ACM.

[11] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[12] Elliot Waingold, Michael Taylor, Devabhaktuni Srikrishna, Vivek Sarkar, Walter Lee, Victor Lee, Jang Kim, Matthew Frank, Peter Finch, Rajeev Barua, Jonathan Babb, Saman Amarasinghe, and Anant Agarwal. Baring It All to Software: Raw Machines. *IEEE Computer*, pages 86–93, September 1997.