Load-Balancing Performance of Consistent Hashing: Asymptotic Analysis of Random Node Join

Xiaoming Wang, *Student Member, IEEE*, and Dmitri Loguinov, *Member, IEEE*

***Abstract*—Balancing of structured peer-to-peer graphs, including their zone sizes, has recently become an important topic of distributed hash table (DHT) research. To bring analytical understanding into the various peer-join mechanisms based on consistent hashing, we study how zone-balancing decisions made during the initial sampling of the peer space affect the resulting zone sizes and derive several asymptotic bounds for the maximum and minimum zone sizes that hold with high probability. Several of our results contradict those of prior work and shed new light on the theoretical performance limitations of consistent hashing. We use simulations to verify our models and compare the performance of the various methods using the example of recently proposed de Bruijn DHTs.**

***Index Terms*—Asymptotic bounds, balls-into-bins, consistent hashing, load balancing, peer-to-peer (P2P).**

I. Introduction

PEER-TO-PEER networks have become a powerful alternative to the client/server infrastructure in the Internet that provides a distributed platform for such applications as web caching, bulk data dissemination, and even media streaming. The latest peer-to-peer networks organize users into massive (millions of nodes) graphs called *distributed hash tables* (DHTs), which provide a scalable, efficient, and fault-tolerant environment for exchanging information between end-users. Even though static DHTs received significant attention in traditional approaches [34], [36], [37], [41], [43] and more-recent developments [7], [12], [16], [22], [24], [26], [32], [40], one of the most important areas of peer-to-peer research remains the study of evolving DHT graphs as users randomly join and leave the system [2], [3], [5], [17], [18], [23], [30], [35].

In classic DHT systems such as Chord [37] or CAN [34], objects are hashed into a virtual coordinate space $\mathcal{S}$, which is dynamically partitioned between n users in the system using nonoverlapping subsets $\mathcal{F}_i \subseteq \mathcal{S}$ ($\cup_{i=1}^{n}\mathcal{F}_i = \mathcal{S}$). Each user keeps track of the peers whose objects hash into its zone $\mathcal{F}_i$ and serves requests for these objects generated by the remaining users. Many performance metrics in a dynamic graph are determined by the distribution of DHT zone sizes held by each peer. Imbalance in zone sizes may lead to increased diameter, smaller node degree, lower bisection width, and higher local congestion during routing through the graph. In addition, uneven zone distribution results in an unfair allocation of user objects to peers and creates "hotspots" in certain parts of the graph. Even though hotspots can be relieved with more sophisticated *object-hashing* techniques [4]–[6], they have no effect on the weakened structure of the underlying graph.

User arrival in most DHTs can be modeled by a three-step process: 1) generation of d points $X_1, \ldots, X_d$ in the DHT space according to some algorithm; 2) sampling of the existing DHT zones $\mathcal{F}_1, \ldots, \mathcal{F}_d$ that contain these points; and 3) splitting of the largest sampled zone. The join process is called *random sampling* if $X_1, \ldots, X_d$ are generated uniformly randomly within the DHT space and *deterministic sampling* if sample points are based on deterministic properties of the underlying graph. Node-join is further classified as *single-point* if $d = 1$ and *multipoint* otherwise. Finally, the join process is called *random-split* if the existing peer's zone $\mathcal{F}_i$ is partitioned at the corresponding sample point X_i and *center-split* if $\mathcal{F}_i$ is always divided in half.

Assuming a sequential join[1] process of n users into a peer-to-peer network, this paper studies how the load-balancing decisions made during node arrival affect the resulting zone distribution and how these algorithms perform as the size of the graph $n \to \infty$.

A. Main Results and Paper Structure

In a random graph of size n, define f_{max} to be the ratio of the largest zone size to the average zone size and f_{min} to be the ratio of the average zone size to the smallest zone size. Among methods that sample a single point in the DHT space [34], [36], [37], it is well known that f_{max} is $\Theta(\log n)$ with high probability [17], [32], [37]. We improve this result by establishing the constants inside $\Theta(\log n)$ and deriving the exact upper, as well as lower, bounds on f_{max} that hold with probability $1 - n^{-c}$, for arbitrary constants c, under both *random* and *center* splits of existing nodes.

Although the largest zone is usually studied for the purposes of load-balancing user objects/keys and the $\Theta(.)$ bounds are clear [17], [32], [37], the minimum zone has not received as much attention. Naor and Wider [32] state without proof that the minimum zone is smaller than average by the same factor $\Theta(\log n)$. Both Loguinov *et al.* [24] and Fraigniaud and Gauron [12] implicitly assume in their derivations that f_{min} is $o(n)$, while [12] additionally concludes that $f_{min} \leq 2^{O(\log n)}$, which essentially means that the upper bound is any power-function of n. To reconcile these partial results, we show that f_{min} is upper bounded by n^{1+c} with probability $1 - n^{-c}$ under random

Manuscript received December 1, 2004; revised December 6, 2005; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Seshan. This work was supported by the National Science Foundation under Grants CCR-0306246, ANI-0312461, CNS-0434940, and CNS-0519442. An earlier version of this paper appeared in ACM SIGMETRICS 2004.

The authors are with the Department of Computer Science, Texas A&M University, College Station, TX 77843 USA (e-mail: xmwang@cs.tamu.edu; dmitri@cs.tamu.edu).

Digital Object Identifier 10.1109/TNET.2007.893881

[1]Note that under certain assumptions, a mixture of joins and departures can be reduced to the pure join model; however, analysis of such scenarios is beyond the scope of this paper.

splits and by $3.246\sqrt{\log n}$ with probability $1-o(1)$ under center splits, where $\log(.)$ represents the natural logarithm throughout the paper. We further show that splitting existing neighbors in the center is in fact *optimal* among all possible splitting methods and the use of uniform (as opposed to nonuniform) hash indexes provides the best possible performance in terms of both f_{max} and f_{min}.

Among multipoint sampling methods, Naor and Wieder [32] select d random points in the DHT space and choose the largest node to split (i.e., the approach used in the classical balls-into-bins "power of two choices" [4], [19], [29], [38]). They show using Chernoff-type bounds that for $d = 8\log n$, f_{max} is upper-bounded by 2 with probability $1 - n^{-2}$. We analyze the same problem using an approach from "balls-into-bins" [28] and derive asymptotic upper/lower bounds on f_{max} for arbitrary d. Our results show that in large graphs

$$f_{max} \leq 2 + \frac{(1+c)\log n}{d}(1 - o(1)) \tag{1}$$

with probability at least $1 - n^{-c}$. Furthermore, we show that there exists an infinite number of graph sizes n such that (1) is tight (i.e., the bound is violated with probability at least n^{-c}). Specifically, for $d = r\log n$ and all values of n, $f_{max} \leq 2 + \frac{1+c}{r}$ with high probability; however, there exist such n that $f_{max} > 2 + \frac{1+c}{r}$ with probability n^{-c}. This result contradicts the one shown in [32] and demonstrates that r must tend to infinity for f_{max} to converge to 2. Also notice that multipoint sampling does *not* lead to the classical $\Theta(\log\log n/\log d)$ bound on f_{max} as might have been expected from the analysis of various balls-into-bins problems [4], [29].

Another zone-balancing approach is first suggested in CAN [34] and later analyzed by Adler *et al.* [2]. In this method, each new node samples a random peer x in the graph and then queries d direct neighbors of x (the graph is assumed to be d-regular). The paper [2] demonstrates that as long as the degree of each node is $\Omega(\log n)$, both f_{max} and f_{min} are some constants (the exact value of the constants is not shown). We study a similar problem, in which nodes are allowed to sample other parts of the graph based on some *deterministic* function (which, for example, may represent the graph's linking rules), and derive upper bounds on f_{max} under this model. Our analysis shows that when $d = r\log n$, the following bound holds with probability at least $1 - n^{-c}$:

$$f_{max} \leq 2 + \frac{1+c}{r} + \eta \tag{2}$$

where η is $\log(1+\frac{1+c}{r}+\log(1+\frac{1+c}{r}+\cdots$ (the bound is tight for infinitely many values of n). This is in contrast to the random sampling model where the upper bound on f_{max} converges to $2 + \frac{1+c}{r}$ for $d = r\log n$. Using this insight, we find that, for example, for $d \approx \log n$ and $c \approx 1$, the deterministic model requires 2.2 times more samples than the purely random model to achieve the same bounds on f_{max}.

Finally, Loguinov *et al.* [24] use a variation of Adler's approach [2], in which the joining peer walks along the edges of the graph starting in a random location and splitting the largest node found within a certain number of hops from the initial node. At each step, the walk is biased toward the largest neighbor; however, since the location of this neighbor varies during the evolution of the graph, closed-form analysis of this approach is rather complicated. We do not offer a model for this method at this time, but compare its performance with that of the remaining methods in simulations.

Other P2P balancing methods include the *virtual-server* approach originally used in Chord [13], [16], [37], the Messor system [31], proximity-aware balancing [42], cluster-based balancing [30], and several other dynamic algorithms [1], [18], [35], which provide alternative mechanisms for balancing P2P graphs and are orthogonal to our analysis.

This paper is organized as follows. Section II provides the background and motivation. Section III studies the random-split model and derives bounds for both f_{max} and f_{min}. In Section IV, we rederive the same bounds for the single-sample, center-split model. Section V studies the maximum zone of multipoint methods and Section VI shows P2P simulations of de Bruijn DHTs. Section VII concludes the paper.

II. Motivation and Preliminaries

Generic load balancing is a relatively old and very well-researched area [19], [29]. This problem typically assumes the existence of n fixed bins and $m \geq n$ objects, which are placed into the bins using uniform, or possibly nonuniform, random selection. Assuming $m = n$, the largest bin will contain $\Theta(\log n)$ balls with high probability, which can be reduced to $\Theta(\log\log n/\log d)$ by sampling d random bins before placing each object [4]. The main application of these results in P2P systems has been balancing of object keys (which we simply call "objects") between the peers [5].

While balancing the number of keys per P2P node is an important objective, we are also concerned with the structure of the graph since failure of high-degree nodes (i.e., peers with large zones) compromises the strength of the underlying graph, congestion in large zones leads to increased response delay, and the presence of low-degree nodes (i.e., peers with small zones) increases the diameter of the system. The first two problems are common to all graphs, while the third one is most noticeable in de Bruijn DHTs [12], [16], [24], [32].

For example, in a system with $n = 10^6$ peers, the maximum zone is between 12 and 28 times larger than average with probability $1 - 1/n$ (we show this result later in the paper). Given a Chord-like system with the average node degree $\log_2 n = 20$, the in-degree of the largest peer is between 240 and 560 with high probability. Once this peer fails, over 200 links are broken simultaneously, leading to rather adverse effects on the graph. It is also true that the largest peer receives routing traffic in proportion to its degree, which may increase the response delay of all queries passing through this node. Finally, if the system utilizes a variation of de Bruijn graphs [12], [32], peers with the smallest zone will have their out-degree equal to 1, will be susceptible to disconnection from the graph, and will experience a larger routing diameter.[2] As we show later in the paper, almost 6% of all nodes in de Bruijn graphs end up with degree 1 under random node join.

We next briefly describe the model of the DHT space utilized in this paper and then proceed to zone-balancing analysis.

[2]This is a consequence of the routing rules in de Bruijn graphs. For more information, see [24].

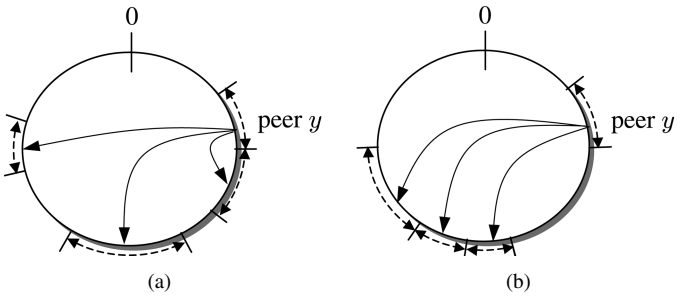


Fig. 1. Peer linkage in DHT graphs. The degree of the graph is 3. (a) Chord. (b) de Bruijn.

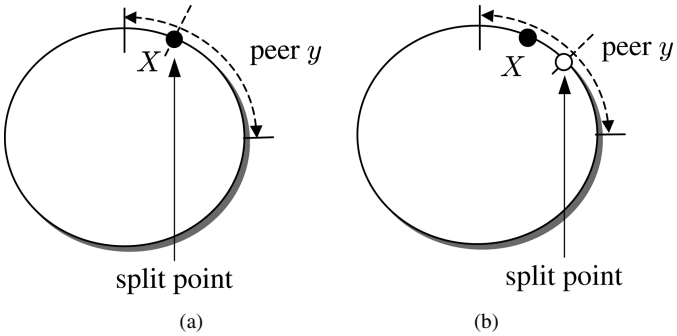


Fig. 2. Illustration of splitting mechanisms. (a) Random split. (b) Center split.

We use the unit-ring model[3] shown in Fig. 1 as the DHT space and dissect how "well" the various distributed join algorithms partition its circumference between the nodes of a P2P system. Fig. 1 shows four peers holding nonoverlapping parts of the ring and three edges originating from peer y. While the linking rules vary between the different types of graphs, they all have the same characteristic – the location of each neighbor is computed based on the zone of peer y and random splitting decisions made during join. Assuming a generic k-regular graph used in most DHTs, Chord links to $k = \log_2 n$ neighbors at exponentially increasing distances Fig. 1(a), while de Bruijn graphs link to k sequential nodes at a certain offset from the original node Fig. 1(b).

The construction of the ring is accomplished through a distributed join process. A new node x selects a random location X in the DHT space based on some hashing function and then attempts to join the peer-to-peer system in or around that random location. In the first approach (e.g., Chord), node x splits the existing peer at exactly X. This is illustrated in Fig. 2(a) where node x splits peer y in the point of x's random hash index X. Notice that this construction leads to the possibility of having very small zones when X lands near one of the boundaries of an existing zone. In the second approach (e.g., CAN), x splits the peer in half as demonstrated in Fig. 2(b). To keep the notation consistent, we call the former method a "random split" and the latter method a "center split."

To further improve fairness in zone sizes, several recent DHTs [12], [32] sample d random locations in the graph and then use a center split of the largest zone they find. Even though these methods perform much better than any of the single-point approaches, sampling d random points in the graph may become costly, especially if d is on the order of $8 \log n$ [32]. This generally leads to $\Theta(d \log n) = \Theta(\log^2 n)$ messages per join, where the constants inside $\Theta(\cdot)$ depend on the diameter of the graph. In Chord with one million nodes (both the diameter and degree are 20), sampling $8 \log n$ peers requires on average 1105 messages and appears excessive.

To decrease the message join overhead, an alternative approach [1], [24] is to deterministically sample the neighbors of the first peer and subsequently walk along the edges of the graph to discover more nodes. This reduces the join overhead by a factor of

$$\frac{dD_{av}}{D_{av} + d/k - 1} = \Theta(kD_{av}) \tag{3}$$

where k is the degree of the graph and D_{av} is the average distance between nodes. For example, in Chord with one million nodes, the deterministic method can sample the same $d = 8 \log n$ peers using 76 times fewer messages than the previous approach (i.e., using only 14.5 messages on average per join). Note, however, that the deterministic method generally must sample more than d points in the graph to provide the same bounds on f_{max} as in the purely random approach.

In the rest of the paper, we address such issues as whether $8 \log n$ is the "correct" value of d for the graph to achieve a desired level of balancing and how many samples in the deterministic method make it equivalent to the purely random approach.

III. Single-Point Random Split

Our treatment of the DHT space assumes a 1-D torus, a purely random and perfectly uniform number generator, and infinite precision of each random hash index (i.e., the probability of collision is zero). We use n to represent the number of peers in the system and focus on deriving the bounds on max/min zone sizes that hold with high probability. Due to limited space, certain proofs have been omitted from this paper and can be found in [39].

Definition 1: An event E_n occurs *with high probability* (w.h.p.) with respect to n if there exists a fixed constant $c > 0$ such that

$$P(E_n) \geq 1 - n^{-c}, \forall n. \tag{4}$$

Typically, (4) ensures stronger bounds on the likelihood of event E_n compared to simply saying that E_n happens "almost surely," or with probability $1 - o(1)$. Although it is customary [1], [17], [32] in this class of problems to derive bounds that hold w.h.p. and study only the asymptotic behavior of the system as $n \to \infty$, we pay special attention to $o(1)$ terms whenever possible and keep our results applicable even to graphs of small size n.

A. Maximum Zone

We next formally define the performance metrics mentioned in the introduction.

Definition 2: Random variable f_{max} is the ratio of the maximum zone size to the average zone size after n points (peers) have joined a random instance of the system.

Definition 3: Random variable f_{min} is the ratio of the average zone size to the minimum zone size after n points have joined the system.

[3]Note that our analysis is not limited to ring topologies and applies to other virtual coordinate spaces.

TABLE I
COMPLIANCE OF f_{max} WITH ITS BOUNDS IN RANDOM SPLITS

n	Range	$1-n^{-c}$	p_l	p_u
$3,000$	$[7.1, 10.4]$	91%	91.6%	91.3%
$30,000$	$[9.1, 13.4]$	95.5%	95.8%	95.7%
$300,000$	$[11.3, 16.4]$	97.7%	97.1%	97.1%

TABLE II
COMPLIANCE OF f_{min} WITH ITS BOUNDS IN RANDOM SPLITS

n	Range	$1-n^{-c}$	p_l	p_u
300	$[131, 2937]$	89.8%	89.8%	89.0%
$3,000$	$[936, 73785]$	96.0%	95.8%	96.3%
$10,000$	$[2714, 398107]$	97.5%	98.1%	98.0%

Both f_{max} and f_{min} are always no less than 1 and provide the main performance metric used throughout the paper. Now suppose that n random points $X_1, X_2, \ldots, X_n$ are independently and uniformly chosen on the unit circle. Define Y_i to be the ith spacing between the points along the circle, M_n to be the largest spacing: $M_n = \max(Y_1, \ldots, Y_n)$, and S_n to be the smallest spacing: $S_n = \min(Y_1, \ldots, Y_n)$.

Theorem 1: Under random splits, each of the following inequalities holds with probability $1 - n^{-c}$:

$$\log n - \log(c \log n) \le f_{max} \le (1+c) \log n. \tag{5}$$

Proof: First, recall the following result due to Darling [9]:

$$\lim_{n\to\infty} P\left(M_n < \frac{\log n + x}{n}\right) = e^{-e^{-x}}. \tag{6}$$

Next, notice that there exists a critical point x at which (6) makes a sharp transition from "almost never" to "almost surely." This percolation effect is common to our problem regardless of how the user joins the graph and is often found in other areas of networking [14]. Recalling that e^z for small z is approximately $1+z$ and substituting $x = -\log(c \log n)$ and $x = c \log n$ into (6), we get both bounds in (5). ∎

Hence, one can conclude that there almost always exists a zone larger than average by a factor of $\log n - \log\log n$, but almost never larger by a factor of $(1+c)\log n$. For example, in a graph with $n = 10^6$ peers, f_{max} is between 12 and 28 with probability $1 - 2/n$. To understand how well these bounds hold for small $n \ll \infty$, we generated 1000 random graphs of three different sizes – 3000, 30 000, and 300 000 nodes. Table I shows in columns p_l and p_u the fraction of graphs in which the actual f_{max} complies with (respectively) the lower and upper bounds of (5) for $c = 0.3$ (ideally, both p_l and p_u should equal $1 - n^{-c}$). As the table shows, f_{max} found in these graphs violates the bounds in (5) with probability very close to the predicted n^{-c}.

B. Minimum Zone

We next examine the behavior of f_{min} in the following theorem and show that these bounds are exponentially worse than those in (5).

Theorem 2: Under random splits, each of the following inequalities holds with probability $1 - n^{-c}$:

$$\frac{n}{c \log n} \le f_{min} \le n^{1+c}. \tag{7}$$

Proof: Recall that all Y_i's are uniformly distributed on the simplex $\{(x_1, \ldots, x_n) : \sum_{i=1}^n x_i = 1\}$ and that [10], [11]

$$P\left(\bigcap_{i=1}^{n} [Y_i > a]\right) = \begin{cases} (1-na)^{n-1}, & na < 1 \\ 0, & na \ge 1. \end{cases} \tag{8}$$

Note that the left side of (8) is the probability that the minimum zone size S_n is at least a. Rewrite (8) in terms of S_n and assume sufficiently large n

$$P(S_n > n^{-\delta}) = \left(1 - \frac{n^{2-\delta}}{n}\right)^{n-1} \approx e^{-n^{2-\delta}}. \tag{9}$$

Substituting $\delta = 2 + c$ into (9), we get the upper bound of (7). Similarly, using $\delta = 2 - \log(c \log n)/\log n$, we get the lower bound of (7). ∎

To illustrate the extent of fluctuation in f_{min}, we again generated 1000 random graphs and examined the number of graphs violating (7) for $c = 0.4$. Table II shows that (7) holds with high accuracy for a variety of graph sizes and that the range to which f_{min} can be confined w.h.p. is substantially larger than traditionally expected [32]. Thus, a 10 000-node graph *almost always* has a peer whose zone size is smaller than average by a factor of 2700. Furthermore, unfairness by a factor of almost 400 000 occurs in $n^{-c} = 2.5\%$ of all random graphs.

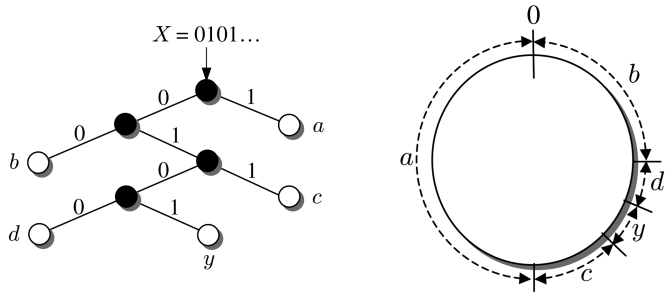
We next show how these bounds can be improved simply by using a different peer-splitting algorithm and derive more pleasant results for f_{min}.

IV. SINGLE-POINT CENTER SPLIT

A. Maximum Zone

Notice that when existing users are split in half by incoming nodes, the DHT space is organized into a dynamic binary trie. The join process of each peer x can be modeled as a ball that drops into the root of the virtual trie and then descends down the tree randomly choosing whether it goes left or right. The leaf at which the ball ends up is the node that x will split. The movement of the ball represents the digits in the binary expansion of x's hash index X (recall that these digits are independent and uniform across all peers according to our assumptions). This model is shown in Fig. 3 where a new incoming node with $X = 0101\ldots$ splits node y, which is the leaf that shares the longest common prefix with x among the existing nodes. Note that a similar tree-based model was independently proposed by Adler *et al.* [2]; however, their analysis is completely different from ours.

Further notice that the zone size of each peer x is a simple exponential function of its depth h_x in the binary trie, i.e., 2^{-h_x}. Thus, the problem of finding f_{max} and f_{min} in "center-split" peer-to-peer DHTs boils down to estimating the probabilistic bounds on the smallest and largest depth of any leaf in the trie. Let h_i be the depth of peer i in a particular (random) instance of the graph, $D = \min_{i=1}^n \{h_i\}$ be the smallest depth, and $H = \max_{i=1}^n \{h_i\}$ be the largest depth of any leaf. Assuming that we can bound both random variables D and H with high probability, what can be said about the resulting bounds on f_{max}



(a) (b)

Fig. 3. (a) Construction of split-trees using random balls. (b) Representation of the same tree on the circle.

and f_{min}? We state the obvious answer to this question in the following lemma without proof.

Lemma 1: Assume a center-split DHT in which $D_l \leq D \leq D_u$ and $H_l \leq H \leq H_u$ hold with high probability. Then, f_{max} and f_{min} are bounded by the following inequalities also with high probability:

$$n2^{-D_u} \leq f_{max} \leq n2^{-D_l} \tag{10}$$

$$n^{-1}2^{H_l} \leq f_{min} \leq n^{-1}2^{H_u}. \tag{11}$$

In what follows, we examine the distribution of D and derive its probabilistic bounds D_l and D_u. The discussion of H is given in the next section.

To begin, we define a sequence of indicator random variables $\{A_i\}$, $i \geq 0$, where $A_i = 1$ if level i of the split-trie is full after n users joined the system and $A_i = 0$ otherwise. We say that a level is *full* if all nodes of that level are present and nonleaf. Notice that level i can be full only if $i < D$ and that $A_i = 1$ implies that $A_k = 1, \forall k < i$. It immediately follows that the smallest leaf depth D is at least $k+1$ if and only if all levels from 0 to k are full

$$P(D \geq k+1) = P\left(\bigcap_{i=0}^{k} [A_i = 1]\right). \tag{12}$$

Using this insight, our next result formulates the distribution of D as a simple recurrence equation.

Lemma 2: In a center-split trie with n leaves, the tail distribution of D for $n \geq 1$ and $k \geq 0$ is given by

$$P(D \geq k+1) = P(D \geq k) P_n(A_k | A_{k-1}) \tag{13}$$

where $P(D \geq 0) = 1$ and $P_n(A_k|A_{k-1})$ is the conditional probability of level k being full given that all previous levels $0, \ldots, k-1$ are full

$$P_n(A_k|A_{k-1}) = P\left(A_k = 1 | D \geq k\right). \tag{14}$$

Notice that recurrence (13) does not limit the number of samples d used in the join process and applies to both single-point and multipoint methods. The only difference between these two approaches is the shape of $P_n(A_k|A_{k-1})$. We show the analysis of single-point split in this section and leave the discussion of multipoint methods for Section V.

Lemma 3: For single-point center-split of the unit-ring, the probability that level k is full given that all previous levels are full is

$$P_n(A_k|A_{k-1}) \approx \exp\left\{-2^k e^{-n2^{-k}+1}\right\}. \tag{15}$$

Proof: First notice that any split-trie built using n peers contains n leaves and $n-1$ nonleaf nodes. Next, examine level k of the trie and observe that all 2^k possible nodes at this level must be nonleaf for level k to be fully split. Assuming that all previous levels are full (i.e., $A_{k-1} = 1$), exactly $2^k - 1$ nonleaf nodes contributed to filling up levels $0, \ldots, k-1$ and the remaining $n - 1 - (2^k - 1) = n - 2^k$ nonleaf nodes had a chance to split level k. After the first $k-1$ levels have been filled up, each node at level k is "hit" by an incoming ball (which splits the node in half) with an equal probability 2^{-k}. Thus, our problem reduces to finding the probability that $u = n - 2^k$ uniformly and randomly placed balls into $m = 2^k$ bins manage to occupy each and every bin with at least one ball. There are many ways to solve this problem, one of which involves the application of well-known results from the coupon collector's problem [33]. We use this approach below.

Define $Z(u)$ to be the random number of *nonempty* bins after u balls are thrown into m bins. Thus, we can write $P_n(A_k|A_{k-1}) = P(Z(u) = m)$. Recall that in the coupon collector's problem, u coupons are drawn uniformly randomly (i.e., each with an equal probability $1/m$) from a total of m different coupons. Then, the probability $Z(u) = m$ to obtain m distinct coupons at the end of the experiment is given by [33]

$$P(Z(u) = m) = \sum_{j=0}^{m} (-1)^j \binom{m}{j} \left(1 - \frac{j}{m}\right)^u. \tag{16}$$

For large u, the term $(1 - j/m)^u$ can be approximated by $e^{-uj/m}$, yielding

$$\begin{aligned} P(Z(u) = m) &\approx \sum_{j=0}^{m} (-1)^j \binom{m}{j} e^{-uj/m} \\ &= \left(1 - e^{-u/m}\right)^m. \end{aligned} \tag{17}$$

Since we are only interested in asymptotically large $m = \Theta(\log n)$, (17) allows a further approximation

$$P(Z(u) = m) \approx e^{-me^{-u/m}} \tag{18}$$

which immediately leads to the result in (15). ∎

The accuracy of (15) is demonstrated in Table III, which shows the distribution of D in simulations for different n. As the table shows, the combined result of (13)–(15) matches simulations very well, especially as n increases.

With the result in Lemma 3, we are now in the place to derive the probabilistic bounds on D.

Theorem 3: Assuming $c \leq 1$, the mass of D concentrates on two values D_l and $D_l + 1 = D_u$ with probability at least $1 - n^{-c}$, where

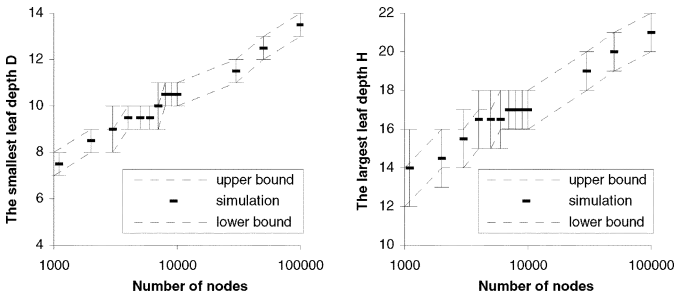
$$D_l = \lfloor \log_2 n - \log_2((1+c)\log n - \rho) \rfloor + 1 \tag{19}$$

and ρ is $\Theta(\log\log n)$.

To verify the correctness of the random-tree model, we generated 1000 random graphs using center splits of the unit circle

TABLE III
SMALLEST DEPTH D OF SPLIT-TREES IN SIMULATION

n	D	Actual	Model	n	D	Actual	Model
3,000	8	0.7%	0.6%	30,000	11	0.3%	0.2%
	9	98.7%	97.6%		12	99.7%	99.7%
	10	0.6%	1.8%		13	0.0%	0.1%
300,000	14	0.1%	0.1%	3,000,000	18	99.9%	99.9%
	15	99.9%	99.9%		19	0.1%	0.1%



(a) (b)

Fig. 4. Unit-ring simulations. Both use center splits and 1000 graphs per value of n. (a) Smallest depth D. (b) Largest depth H.

and examined the smallest depth D in each execution. We used $\varepsilon = \log 1000/\log n$ in (19) to guarantee 99.9% confidence in the bounds. Fig. 4(a) shows that the actual results (whose spread is shown with vertical bars) follow the model very well. Notice that as $n \to \infty$, the mass of D indeed concentrates on two values D_l and $D_l + 1$.

It is also easy to notice that depending on the value of n, the upper bound on f_{max} fluctuates anywhere between $\frac{1+c}{2}\log n - \Theta(\log\log n)$ and $(1+c)\log n - \Theta(\log\log n)$ [the result depends on the floor function in (19)].

B. Minimum Zone

Next, we focus on estimating the largest depth H (i.e., the *height*) of the tree in Fig. 3(a). Even though this problem appears similar to the one just studied, the results are substantially different as can be seen in the next theorem.

Theorem 4: With probability $1 - o(1)$, the mass of H in center-split DHTs concentrates on three values H_l, $H_l + 1$, and $H_l + 2 = H_u$, where

$$H_l = \left\lfloor \log_2 n + \sqrt{2\log_2 n} - 1.5 \right\rfloor. \tag{20}$$

Note that the "$o(1)$" term in the statement of Theorem 4 depends on the decimal expansion of n and simply equals zero in many practical graphs of nontrivial size [20]. Fig. 4(b) shows simulation results (99.9% confidence) from the unit-ring topology for the largest leaf depth H and the corresponding bounds from (20). Together with Table IV, these simulations demonstrate that the mass of H in fact centers on three values as $n \to \infty$.

The result of Theorem 4 is quite interesting since it shows that by constructing a simple split-tree, the bound on f_{min} can be significantly improved from $\Theta(n^{1+c})$ shown in the previous section to $\Theta(2^{\sqrt{2\log_2 n}}) = \Theta(3.246^{\sqrt{\log n}})$. Nevertheless, this bound is still noticeably worse than f_{max}'s $\Theta(\log n)$.

TABLE IV
HEIGHT H OF SPLIT-TREES IN SIMULATION

n	H_l	H_u	Actual height H
3,000	14	16	14: 5.5%, 15: 86.1%, 16: 8.4%
30,000	18	20	18: 3.8%, 19 : 89.7%, 20: 6.5%
300,000	22	24	22: 17.5%, 23: 81.0%, 24: 1.5%
3,000,000	26	28	26: 46.7%, 27: 52.9%, 28: 0.4%

Neglecting the ceiling and floor functions in (19) and (20), consider $n = 10^6$ and $c = 1$. In this case, f_{min} is upper limited by 67, while f_{max} is just below 28. For $n = 10^9$, f_{min} is limited by 274 and f_{max} by 41. Another example of this difference can be observed from the simulations in Tables III and IV. Using the last row in both tables ($n = 300\,000$), notice that $f_{max} \le 18.3$, while $f_{min} \le 55.9$ with probability $1 - o(1)$.

C. Optimality

We conclude this section by observing that splitting an existing neighbor in half is in fact *optimal* among all methods that sample a single peer in the circle.

Theorem 5: For single-point sampling, f_{max} and f_{min} are minimized in expectation by using a uniform hashing function and splitting existing neighbors in the center.

Proof: First notice that any off-center splitting of existing nodes produces *nonuniform* random trees where the probability for the ball to drop toward each child is proportional to the size of the child (this is easy to explain since the probability that hash index X of a new node belongs to a certain zone is simply proportional to that zone's size). This can be modeled as a random p-tree, where p is the probability for the ball to drop left and $1 - p$ to drop right from any given node. Recall that the expected depth of a node in a random p-tree is given by [33]

$$E[h_i] = \frac{\log n}{\mathcal{H}} \tag{21}$$

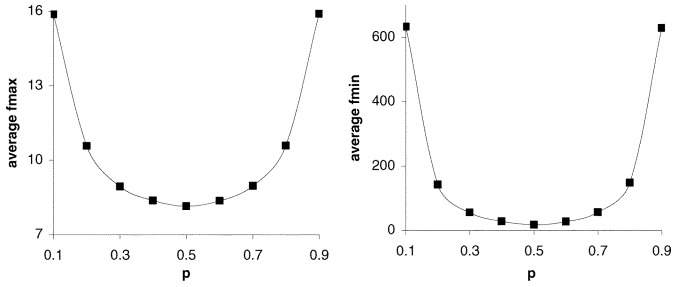
where entropy $\mathcal{H} = -[p\log p + (1-p)\log(1-p)]$. It is easy to verify that (21) has a unique global minimum at $p = 1/2$. Using similar reasoning, nonuniform hashing functions (in which, for example, zeros appear with probability p and ones with probability $1 - p$) also produce unbalanced p-trees and are therefore suboptimal. ■

The result of this theorem is illustrated in Fig. 5, which shows the average values of f_{max} and f_{min} in 1000 random graphs for $n = 30\,000$ and off-center splitting of existing peers (p and $1-p$ are the two fractions into which each peer is split). The same splitting method can be interpreted as peers applying a nonuniform hashing function in which zeros appear with probability p and ones with probability $1 - p$. The figure clearly shows that $p = 0.5$ is optimal in both cases and confirms our prior observation that f_{min} is substantially harder to bound than f_{max}.

V. MAXIMUM ZONE OF MULTIPOINT SAMPLING

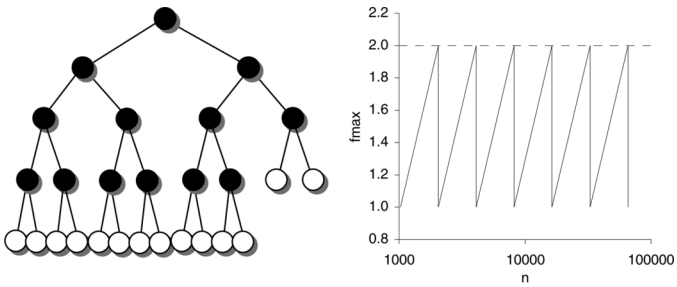
A. Ideal Case

The simplest method of balancing the graph with multipoint sampling is to allow the joining user to examine *all* existing peers in the system and then split the largest found zone. In this setting, the distribution of zone sizes is optimal and f_{max} fluctuates between 1 and 2 as formally stated below.



(a) (b)

Fig. 5. Average f_{max} and f_{min} in off-center split schemes with probability p. Both simulations use 30 000 nodes and 1000 iterations. (a) f_{max}. (b) f_{min}.



(a) (b)

Fig. 6. (a) Instance of the split-tree under the ideal model with $n = 14$. (b) Ideal f_{max} for n between 1024 and 65536.

Lemma 4: If all existing users are sampled during join, f_{max} is given by

$$f_{max} = n2^{-\lfloor \log_2 n \rfloor}. \tag{22}$$

Fig. 6(a) shows an instance of the split-tree generated in the ideal case and Fig. 6(b) plots the ideal f_{max} as a function of n. Notice that the peaks of this curve reach 2 in points where n is one less than an integer power of 2 and the split-tree forms an almost complete binary tree with height $\log_2 n$.

It is clear that ideal load-balancing achieves the best possible result in terms of f_{max}; however, this approach suffers from high traffic overhead for each joining node, especially when n is large. Thus, a question arises of whether there exist solutions that produce small f_{max} and simultaneously incur low join overhead. To explore this problem, we next study two multipoint sampling methods and derive the smallest number of samples d needed to upper-bound f_{max} by 2 with high probability.

B. Random Model

In this section, we examine the behavior of the maximum zone when each incoming peer is allowed to sample d random locations in the ring (as before, the implicit assumption here is that the peer will split the largest discovered node). We again model this problem with split-trees, examine the evolution of the system as we add a new peer into the network during each time step, and derive the conditional probability $P_n(A_k|A_{k-1})$ that level k is fully split given that all previous levels are.

We only model the center-split approach since all proposed multipoint methods split the sampled nodes in the middle. Further note that in multipoint sampling, D_u is equal to either D_l (large d) or $D_l + 1$ (small d). Therefore, in the rest of the paper, we limit our analysis to D_l since its value can be trivially used to obtain D_u and also deduce upper (rather than lower, which are arguably less useful) bounds on f_{max}.

In what follows below, we show how to reduce the derivation of $P_n(A_k|A_{k-1})$ to the classical "power of two choices" problem, which has been studied by Mitzenmacher [27], [28] based on Kurtz's theorem and general theory of density-dependent Markov processes [21].

Lemma 5: For d-point sampling and center-splits of the unit-ring, $P_n(A_k|A_{k-1})$ is given by

$$P_n(A_k|A_{k-1}) = \left(\frac{\mu(n)}{2^k}\right)^{2^k} \tag{23}$$

where $\mu(n)$ is the solution to the following differential equation at time $t = n$:

$$\frac{d\mu(t)}{dt} = 1 - \left(\frac{\mu(t)}{2^k}\right)^d \tag{24}$$

with initial condition $\mu(t) = 0$ for $t \leq 2^k$.

Proof: Again, skipping the first $2^k - 1$ join events that split levels above k, we model the process of the remaining join events with $j = n - 2^k$ balls dropping into $m = 2^k$ bins and examine the evolution of the number of nonempty bins as a function of time t. Before each ball is placed into a bin, we sample d random bins and place the ball into the least-occupied bin. The goal of our analysis is to determine the probability that all m bins are occupied at the end of this process.

We assume that the system starts at time $t_0 = 2^k$, stops at time n, and adds one new node to the DHT at each integer time unit $t \geq t_0$. Let $Z(t)$ be the number of nonempty bins at time t in a given (random) instance of the graph process. Under this notation, $P_n(A_k|A_{k-1}) = P(Z(n) = m)$ and $Z(t) = 0, \forall t \leq t_0$. Further, let $\mu(t) = E[Z(t)]$ be the expectation of $Z(t)$ over all random graphs. Given $m = 2^k$ bins with d options, $\mu(n)$ can be written as the solution to the differential (24) [27], [28].

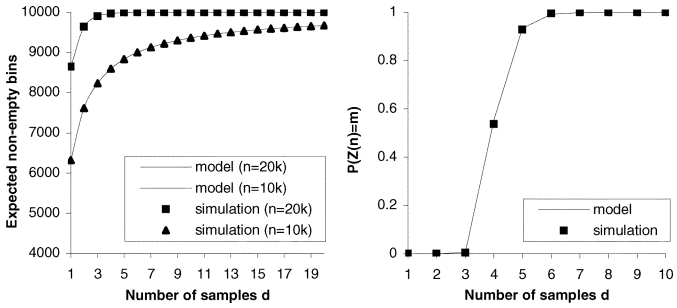
Now assuming that $\mu(t)$ is known, we need to determine the probability that all bins are full at the end of the experiment. To resolve this issue, define W_i to be the number of balls in bin i and $B_i = 1$ if $W_i \geq 1$ and $B_i = 0$ otherwise. Next, since $\{W_i\}$ are only constrained by their summation (i.e., $\sum_{i=1}^{m} W_i = n - m$), it follows that for large n and m, $\{W_i\}$ asymptotically behave as if they were completely independent [10]. A similar observation applies to $\{B_i\}$, which leads to the fact that $Z(n) = \sum_{i=1}^{m} B_i$ can be approximated by a binomial random variable $B(m, q)$, where $m = 2^k$ is the number of bins and $q = P(B_i = 1)$ is the probability that any given bin is nonempty at time n.

To compute $P(Z(n) = m)$, we first determine the value of q. Since the expectation of $B(m, q)$ is $mq = \mu(n)$, q is readily available as $\mu(n)/m$. Then, the probability that a binomial variable equals its maximum value is given by

$$P(B(m,q) = m) = \binom{m}{m} q^m (1-q)^0 = q^m \tag{25}$$

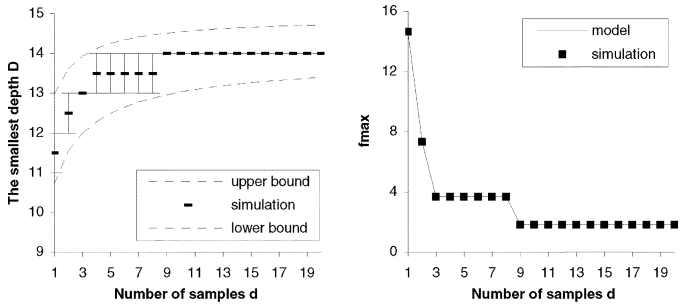
which immediately leads to the result in (23). ■

We conducted numerous balls-into-bins simulations to verify the accuracy of (24). For all values of d and even values of n as small as 500, $\mu(n)$ matched $E[Z(n)]$ remarkably well. Fig. 7(a) shows the quality of the fit between $\mu(n)$ and $E[Z(n)]$ for $m =$



(a) (b)

Fig. 7. (a) Comparison of the expected number of nonempty bins to the solution of (24) in 1000 iterations. (b) Comparison of the actual probability $P(Z(n) = m)$ to the solution of (23).



(a) (b)

Fig. 8. (a) Continuous upper/lower bounds on D from (23) and the actual smallest depth in simulations. (b) The discrete upper bound on f_{max} and that in simulations.

10 000 and two cases of n: 10 000 and 20 000 balls (simulation results are plotted as isolated points and the models are drawn as continuous lines). Fig. 7(b) shows probability $P(Z(n) = m)$ in another simulation for $m = 10\,000$ and $n = 30\,000$ balls. As seen in both figures, both models (23) and (24) follow the actual result seamlessly.

Our next simulation compares the bounds on the smallest depth D obtained from (23) to those observed in simulations of the unit-ring for $n = 30\,000$. We use a binary search to find two values of k in (23) that guarantee $P_n(A_k|A_{k-1}) = 1 - n^{-c} = 0.999$ (lower bound on D) and $P_n(A_k|A_{k-1}) = n^{-c} = 0.001$ (upper bound on D). Note that we call these bounds "continuous" since they generally produce noninteger k. Fig. 8(a) shows the spread of D observed in 1000 simulations (99.9% confidence) and the corresponding continuous upper and lower bounds. After converting noninteger k of the previous example to "discrete" lower bound $D_l = \lfloor k + 1 \rfloor$, we plot in Fig. 8(b) the upper bound on f_{max} in comparison to that in simulations. As seen in both figures, the result of Lemma 5 provides a very accurate estimate of both D and f_{max}.

Since (23) does not generally allow a closed-form solution for large d, one must resort to binary search or similar methods to obtain the probability that f_{max} exceeds a certain threshold. This approach is time consuming and says nothing about how f_{max} behaves as a function of d. Thus, to overcome these limitations, we next derive an asymptotic expansion of (23) for arbitrary d and demonstrate its accuracy in simulations.

C. Asymptotic Expansion of (23)

In this section, we study the behavior of the solution to (24) and obtain a closed-form expression for the bounds on D that are satisfied with high probability.

Theorem 6: Under d-point sampling and center-splits, the minimum tree depth D is bounded from below by D_l with probability at least $1 - n^{-c}$, where

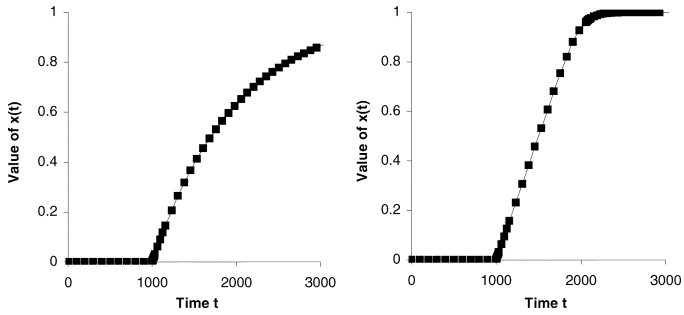
$$D_l = \lfloor \log_2 n + \log_2 d - \log_2((1+c)\log n - \beta) \rfloor + 1 \quad (26)$$

$$\beta = \log(2d + (1+c)\log n) - \log \xi - 2d \quad (27)$$

for some small constant $0.2 < \xi < 0.5$.

Proof: Set $x(t) = \mu(t)/m$ and rewrite (24)

$$m\frac{dx}{dt} = 1 - x^d \quad (28)$$



(a) (b)

Fig. 9. Numerical solution to (28) for $d = 1$ and $d = 10$ with 1000 bins and 3000 balls. (a) $d = 1$. (b) $d = 10$.

with the initial condition $x(t) = 0, \forall t \le t_0$, where $t_0 = 2^k$. The value of $x(t)$ determines the expected fraction of full bins at time t and is always between 0 and 1. Notice that when t is small, the system starts at $x(t_0) = 0$ and linearly increases until x^d becomes nonnegligible compared to the constant 1 in (28). Thus, the solution to this equation consists of two "fundamental" curves, one of which is almost linear and the other one is strictly nonlinear. This is shown in Fig. 9 for $m = 1000$ and $n = 3000$. Notice the substantially longer linear component for $d = 10$ on the right side of the figure.

We say that the system "switches" from linear to nonlinear slope when x^d exceeds a certain threshold ϕ. The exact value of ϕ is not essential and all values in the range [0.1, 0.3] lead to similar results according to our analysis. Under these assumptions, the breaking point t_1 between linear and nonlinear slopes can be computed from $x^d(t_1) = \phi$, which leads to $x(t_1) = (t_1 - t_0)/m = \phi^{1/d}$ (in this expansion, we use the fact that $x(t)$ is a linear function of t before time t_1). The remaining nonlinear curve can be easily estimated from (28) assuming nonnegligible values of x

$$m\frac{dx}{dt} = 1 - x^d = 1 - (1-y)^d \approx yd \quad (29)$$

where $y(t) = 1 - x(t)$ tends to zero for large t and is taken to be small enough to apply Taylor expansion in (29). The solution to (29) is

$$x(t) = 1 - Ce^{-dt/m}, \qquad t \ge t_1 \quad (30)$$

where constant C is selected from the initial condition $x(t_1) = (t_1 - t_0)/m$. Expanding C, we have

$$x(t) = 1 - (1 - \phi^{1/d})e^{-d(t - t_0 - \phi^{1/d}m)/m}, \qquad t \geq t_1. \quad (31)$$

Next, substituting $t_0 = 2^k$ into (31), we get

$$x(n) = 1 - (1 - \phi^{1/d})e^{-d(n - 2^k - \phi^{1/d}m)/m} \quad (32)$$

since $t = n$ always belongs in the nonlinear part of the curve. Substituting (32) and $x(t) = \mu(t)/m$ into (23), we have

$$P_n(A_k|A_{k-1}) = \left(1 - (1 - \phi^{1/d})e^{-d(n - 2^k - \phi^{1/d}m)/m}\right)^m. \quad (33)$$

To understand (33), set level k to the following expression:

$$k = \log_2(dn) - \log_2((1 + c)\log n - \beta) \quad (34)$$

where β is the term we determine below. Expanding $m = 2^k$ and $(1 + z)^y \approx e^{zy}$ for small z and large y, we get

$$P_n(A_k|A_{k-1}) \approx \exp\left\{-n^{-c}\frac{(1 - \phi^{1/d})e^{d\phi^{1/d}}de^{\beta + d}}{d + (1 + c)\log n - \beta}\right\}. \quad (35)$$

Noticing that $d(1 - \phi^{1/d})$ for large d is a constant equal to $-\log\phi$, we extract ξ from (35)

$$\xi = d(1 - \phi^{1/d})e^{-d(1 - \phi^{1/d})} \approx -\phi\log\phi. \quad (36)$$

Since $P(D \geq k)$ is no smaller than $P(D \geq k+1) \geq 1 - n^{-\varepsilon}$, we can approximate $P(D \geq k + 1)$ with $P_n(A_k|A_{k-1})$ to get

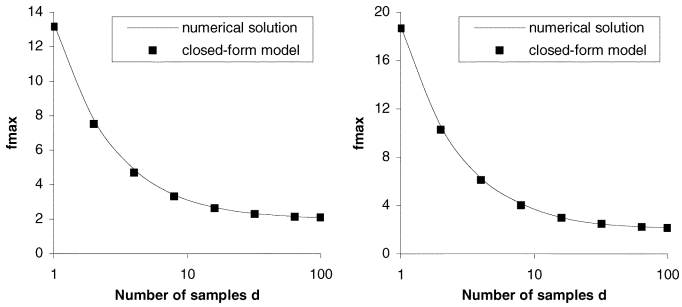
$$P(D \geq k + 1) \approx \exp\left\{-n^{-c}\frac{\xi e^{\beta + 2d}}{d + (1 + c)\log n - \beta}\right\}. \quad (37)$$

With the help of Lambert's function W (i.e., a multivalued solution to $e^{W(z)}W(z) = z$) [8], we can ensure that the term next to n^{-c} is exactly 1 using the following β:

$$\begin{aligned}\beta &= (1 + c)\log n - W(e^{2d}n^{1+c}\xi) \\ &\approx \log(\log\xi + 2d + (1 + c)\log n) - \log\xi - 2d \\ &\approx \log(2d + (1 + c)\log n) - \log\xi - 2d \end{aligned} \quad (38)$$

where the last approximation holds since $\log\xi$ is negligible compared to the other terms. Recalling that $D_l = \lfloor k + 1 \rfloor$ and combining (38) with (34), we have (26). ■

We verify the result of this theorem by again solving (23) for D_l using a binary search to achieve 99.9% confidence. We test these numerical bounds against the model (26) using two examples with 3000 and one million nodes n. In the former case, $c = 0.86$ and in the latter case, $c = 0.5$. We use these values of c in (26) and directly obtain D_l, which leads to the corresponding upper bound on f_{max}. Fig. 10 shows the result of this process and confirms that (26) is very accurate. As both parts of the figure show, the value of f_{max} first drops almost linearly, but then the slope flattens out and f_{max} converges to 2.1 and 2.2, respectively, at $d = 100$. We also examined the effect of varying ϕ between 0.1 and 0.5 and observed no significant impact on the outcome. We use $\phi = 0.2$ throughout the paper for convenience since it results in ξ being close to $1/e$ regardless of d.



(a) (b)

Fig. 10. Comparison of the continuous bounds on f_{max} from a numerical solution to (23) (99.9% confidence) to those from the closed-form model (26) for 3000 nodes and one million nodes. (a) $n = 3000$. (b) $n = 10^6$.

We can now rewrite the main result (26) in terms of f_{max}.

Corollary 1: For all sufficiently large n, f_{max} is bounded by the following with probability at least $1 - n^{-c}$:

$$f_{max} \leq 2 + \frac{(1 + c)\log n}{d} - \frac{\Theta(\log(d + \log n))}{d}. \quad (39)$$

Furthermore, for each $N > 0$ there exists $n > N$ such that the bound in (39) is tight (i.e., violated with probability at least n^{-c}).

Note that (39) is an upper bound that holds for *all* large n. It is possible to carefully select n such that the term inside the floor function in (26) is an integer, in which case f_{max} can be bounded by half of what is shown in (39). For other choices of n, f_{max} will fluctuate between $1 + \frac{(1+c)\log n}{2d}(1 - o(1))$ and $2 + \frac{(1+c)\log n}{d}(1 - o(1))$.

Further analysis of (39) for $d = r\log n$ yields

$$f_{max} \leq 2 + \frac{1 + c}{r} - o(1). \quad (40)$$

Assuming sufficiently large graphs and neglecting the $o(1)$ term, $d = \log n$ samples can bound f_{max} by $3 + c$ and $d = 8\log n$ samples by $2.125 + c/8$ with high probability.

D. Deterministic Model

In this section, we use a different model of sampling d points along the circle, which relies on one random and $d-1$ deterministic choices. This method arises when the new node samples direct neighbors of a randomly chosen peer, where the neighbors are predetermined by some fixed rules (a similar model is studied in [2] as discussed in the introduction). To model this situation, we organize the nodes at level k of the split-trie into nonoverlapping groups of size d. If the first random point (ball) lands into group j, the peer is allowed to sample the remaining $d - 1$ points of the group. Hence, grouping is symmetric and deterministically leads to the same result regardless of where within the group any given point (ball) lands.

One example of this framework is shown in Fig. 11 for $d = 4$. In the figure, zone A always samples three other (known) locations of the circle. This can be implemented by adding 1/4, 1/2, and 3/4 of the circle's circumference to the location of the first point and then sampling the peers holding these additional

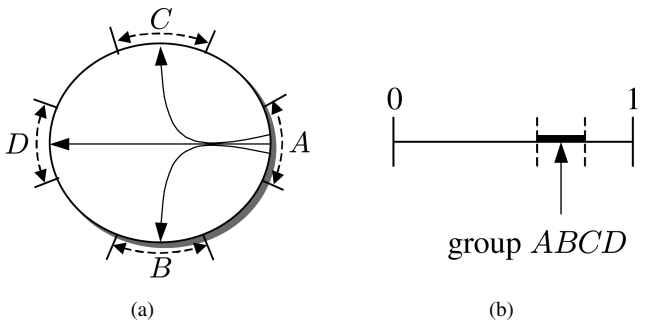


Fig. 11. (a) Model of deterministic peer sampling. (b) Its representation in terms of groups.

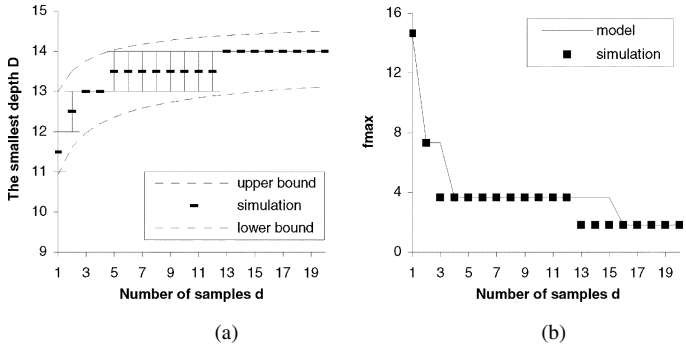


Fig. 12. (a) Continuous upper/lower bounds from model (41) and the actual D (99.9% confidence) in simulations of the unit circle for 30 000 nodes. (b) The discrete upper bound on f_{max} computed from (41) and that in simulations.

points. If these locations happen to be A's neighbors, then sampling comes with no additional message overhead. This model is simple to generalize to any value of d as long as the individual zones do not overlap.

Finally, note that this deterministic model does *not* directly correspond to the linking rules of any particular P2P network since it isolates the nodes in each group from the rest of the graph. Nevertheless, the above model leads to very interesting results and provides a baseline comparison with the purely random approach.

Lemma 6: Assuming deterministic sampling of d bins in each group and nonoverlapping groups, $P_n(A_k|A_{k-1})$ is given by

$$P_n(A_k|A_{k-1}) \approx \left(\frac{B_{d/2^k}(d, n-2^k-d+1)}{B(d, n-2^k-d+1)}\right)^{2^k/d} \quad (41)$$

where $B(a,b)$ is the beta function and $B_x(a,b)$ is the incomplete beta function [25]

$$B_x(a,b) = \int_0^x t^{a-1}(1-t)^{b-1}dt. \quad (42)$$

Proof: We apply the same approach as in previous sections and study the probability that $u = n - 2^k$ balls placed into $m = 2^k$ bins are able to "split" each of the m bins. First notice that every bin within a given group j is split as long as at least d balls land into group j. Therefore, we need to compute the probability that *each* group receives at least d random balls out of u. The number of balls N_j that are thrown into group j is given by a binomial distribution $B(u, d/m)$, where u is the number of balls and d/m is the probability that a new ball is randomly placed into group j. Ignoring the mild dependency between $\{N_j\}$ (which asymptotically makes no difference), the probability that all groups receive at least d points is

$$P\left(\bigcap_{j=1}^{m/d}[N_j \geq d]\right) \approx P\left(B(u, d/m) > d\right)^{m/d}. \quad (43)$$

Next, recall that the upper tails of a binomial random variable can be expressed using the regularized beta function [25]

$$P\left(B(u, d/m) \geq d\right) = \frac{B_{d/m}(d, u-d+1)}{B(d, u-d+1)} \quad (44)$$

where $B(a,b)$ is the beta function and $B_x(a,b)$ is the incomplete beta function in (42). From (43) and (44), the result (41) follows immediately. ■

As expected, for $d = 1$, (41) simplifies to become (15); however, for larger values of d, we need to use numerical methods to compute (41). An alternative method is to derive an estimate for the upper tails of the binomial distribution and simplify (41) to a more workable form. We carry out this task in the next section and in the meantime, check the accuracy of the beta-function model in simulations.

In all balls-into-bins experiments, (41) was perfectly accurate. We skip these results for brevity and instead focus on the accuracy of the model in bounding the value of the smallest depth D. Using binary-search and $n = 30\,000$, we solved (41) to obtain continuous upper/lower bounds D_u and D_l. In Fig. 12(a), we compare these bounds to the actual value of D (99.9% confidence) observed in simulations of the unit circle. During simulations, each joining peer deterministically sampled $d-1$ additional locations in the ring by adding $i/d, i = 1, \ldots, d-1$, to its original hash index X. As seen in the figure, the beta-function model accurately tracks the evolution of D.

In Fig. 12(b), we show the *discrete* version of f_{max} (i.e., after applying the corresponding floor function) from model (41) and compare it to that obtained in simulations. As the figure shows, the fixed-bin structure of the model is too "conservative" for the unit-ring in cases when the number of groups $2^k/d$ is not an integer and overestimates the real f_{max} in points when D makes a jump. This issue notwithstanding, we find that (41) provides a good approximation to our class of deterministic sampling methods.

E. Asymptotic Expansion of (41)

We next study how (41) behaves for different values of d.

Lemma 7: The result in (41) can be converted to a more "digestible" form as follows:

$$P_n(A_k|A_{k-1}) \approx \exp\left\{-\frac{n^{-c}e^{-\beta+d}}{(d-1)!} \times \frac{[(1+c)\log n + \beta - d]^d}{[(1+c)\log n + \beta - d]^2 - d^2}\right\} \quad (45)$$

where

$$\beta = dn2^{-k} - (1+c)\log n. \quad (46)$$

Proof: We first use the well-known Poisson approximation to the binomial distribution in (44) and then apply tail expansion to the resulting Poisson distribution [15]

$$P(B(n-2^k,p)\le d-1)\approx\sum_{k=0}^{d-1}\frac{e^{-\lambda}\lambda^k}{k!}\approx\frac{e^{-\lambda}\lambda^{d-1}}{(d-1)!}\frac{1}{1-(d-1)/\lambda}\quad(47)$$

where $\lambda=(n-2^k)p=(n-2^k)d/m$ is the mean of both distributions. Substituting the complement of the probability in (47) into (41), we have

$$P_n(A_k|A_{k-1})\approx\left(1-\frac{e^{-(n-2^k)\frac{d}{m}}}{(d-1)!}\frac{\left[(n-2^k)\frac{d}{m}\right]^{d-1}}{1-\frac{(d-1)m}{(n-2^k)d}}\right)^{\frac{m}{d}}.\quad(48)$$

After basic arithmetic manipulations, (48) becomes (45). ■

The approximation in (45) was almost identical to the original beta function in (41) in all comparisons that we performed. A typical fit between the two models for one case of $n=30\,000$ is shown in Fig. 13(a). However, since (45) by itself is not very useful and requires a binary search just like (41), our next step is to derive the exact bound on the smallest depth D that holds with probability $1-n^{-c}$.

Theorem 7: In deterministic sampling, the minimum split-tree depth D is bounded from below by $D_l=\lfloor-\log_2 M_n\rfloor+1$ with probability at least $1-n^{-c}$, where M_n is the largest zone size

$$M_n=\begin{cases}\frac{W(en^{1+c})}{n}, & d=1\\ \frac{(1+c)\log n+2}{2n}, & d=2\\ \frac{(d-2)Q(n,d,c)+d}{dn}, & d\ge 3\end{cases}\quad(49)$$

where $W(z)$ is Lambert's function as before, $Q(n,d,c)$ is given by

$$Q(n,d,c)=-W_{-1}\left(-\frac{\left[(d-1)!n^{-(1+c)}\right]^{\frac{1}{d-2}}}{d-2}\right)\quad(50)$$

and $W_{-1}(z)$, for negative z, is the secondary branch of multi-valued Lambert's function W [8].

Proof: Simplifying (45) and omitting insignificant constants

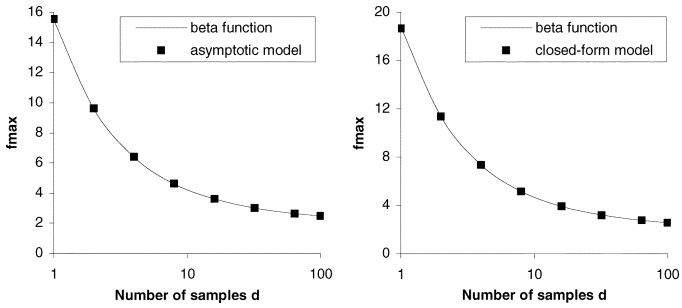
$$P_n(A_k|A_{k-1})\approx\exp\left\{-n^{-c}e^{-\beta+d}\times\frac{((1+c)\log n+\beta-d)^{d-2}}{(d-1)!}\right\}.\quad(51)$$

To obtain the desired result, we must solve

$$e^{-\beta+d}\frac{[(1+c)\log n+\beta-d]^{d-2}}{(d-1)!}=1\quad(52)$$

for β. For trivial values of d equal to 1 and 2, the solution to (52) is elementary and is shown in (49) (we omit the derivations). For $d\ge 3$, a sequence of straightforward, but rather technical manipulations brings (52) into the canonical form $xe^x=z$, which allows the application of Lambert's function W. Unfolding the value of z in a separate set of steps, we arrive at the result in (49). We omit these details for brevity. ■

The result in (49) is a major improvement over (41) since it requires no binary search to compute the upper bounds on f_{max}. For any given d, n, and c, (49) directly produces the result,



(a) (b)

Fig. 13. (a) Beta-function solution (41) and asymptotic model (45) for 30 000 nodes. (b) Verification of the closed-form model (49) against the beta-function (41) for one million nodes.

where $W_{-1}(z)$ can be easily computed in many software packages (such as Matlab or Mathematica). We verified the bounds on f_{max} derived from (49) in numerous tests. One example for $1-n^{-c}=0.999$ and one million nodes is shown in Fig. 13(b).

While the result of the last theorem allows an easy computation of the bounds on f_{max}, it is still not clear how this metric in the deterministic method compares to that in the random approach. To address this question, we present a much simpler shape of (50) assuming $d=r\log n$.

Theorem 8: For $d=r\log n$, (49)–(50) simplify to the following:

$$f_{max}\le 2+\frac{1+c}{r}+\eta-\frac{\Theta(\log\log n)}{r\log n}\quad(53)$$

where η is

$$\eta=\log\left(1+\frac{1+c}{r}+\log\left(1+\frac{1+c}{r}+\cdots\right.\right.\right.\quad(54)$$

Proof: The derivations are again straightforward and we omit certain trivial steps. Substituting $d=r\log n$ and $\beta=(\eta+1)d$ into (45), after some manipulations and application of Stirling's formula to $(d-1)!$, we have

$$P_n(A_k|A_{k-1})\approx\exp\left\{\frac{-n^{-c}e^{-\beta+d}\left(1+\eta+\frac{1+c}{r}\right)^{r\log n}}{\Theta(\log n)}\right\}.\quad(55)$$

As before, we need to solve the following recurrence:

$$\frac{e^{-\beta+d}\left(1+\eta+\frac{1+c}{r}\right)^{r\log n}}{\Theta(\log n)}=1.\quad(56)$$

Expanding β and taking the logarithm of both sides we have

$$-\eta r\log n+\log\left(1+\eta+\frac{1+c}{r}\right)r\log n=\Theta(\log\log n)\quad(57)$$

from which

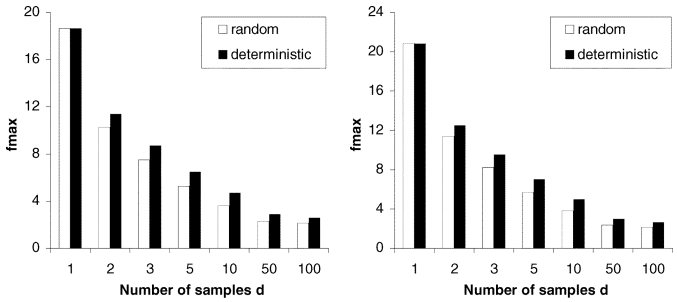
$$\eta r\log n=\log\left(1+\eta+\frac{1+c}{r}\right)r\log n-\Theta(\log\log n).\quad(58)$$

For the solution to exist, both sides of (58) must have the same asymptotics, or in other words, η must satisfy $\eta=\log(1+\eta+\frac{1+c}{r})$, which leads to the result in (54). Substituting $\beta=(\eta+1)d-\Theta(\log\log n)$ into (46), we have the bound in (53). ■

The result in (53) is very interesting as it shows that for example, for $d=\log n$ and $c=0.1$, η is approximately 1.1913

TABLE V
f_{max} COMPUTED BY MODELS (41) AND (49) FOR VERY LARGE n

n	10^6	10^9	10^{12}	10^{32}	10^{72}	10^{305}
Model (41)	3.72	3.93	4.05	–	–	–
Model (49)	3.71	3.91	4.02	4.27	4.37	4.43



(a) (b)

Fig. 14. Comparison of f_{max} in random and deterministic sampling. (a) $n = 10^6$. (b) $n = 10^7$.

and f_{max} converges to $3 + c + \eta \approx 4.2913$ for sufficiently large n. This is in contrast to random sampling, where f_{max} converges to 3.1. We next verify the asymptotics of (53) for growing n and $c = 0.22$. For this c, the value of η is 1.2418 and the asymptotic bound in (53) is 4.4618. Table V shows the convergence process for f_{max} computed using both the beta function in (41) and Lambert's function in (49). Matlab's ability to compute the incomplete beta function (41) stops at approximately $n = 10^{12}$, while (49) provides results up to $n = 10^{305}$. The table shows that the $o(1)$ term in (53) slowly decays to zero and that f_{max} converges to a value very close to the one predicted by the model.

F. Discussion

The deterministic model clearly provides worse performance than the random model studied earlier; however, the difference in terms of f_{max} between the models is not as significant as one might have expected. This is shown in Fig. 14 for two values of n, where the deterministic model obtains f_{max} larger than that in the random model by a small additive constant.

Several remaining issues are whether random or deterministic sampling can achieve optimal (i.e., best possible) load balancing of P2P zones using *logarithmic* d and how many samples make the deterministic model equal to the random one. We first define "optimality" and then discuss which models can actually achieve it.

Theorem 9: For any $N > 0$, there always exists such $n > N$ that under arbitrary splitting mechanisms and for any number of samples d, the actual f_{max} in every random graph of size n is at least 2.

With the aid of this theorem, it becomes apparent that the random sampling mechanism in (40) achieves *optimal* load balancing with $d = r \log n$ only when $r \to \infty$. All such functions (e.g., $\log\log n \cdot \log n$ and $\log^2 n$) are super-logarithmic and thus provide a negative answer to our question above. This result is illustrated in Fig. 15, which plots the upper bound on f_{max} obtained from model (26) and (27) and actual simulations[4] for $d = 8\log n$ and $d = \log^3 n/10$. As expected, the former case does not achieve optimal zone-balancing (f_{max} converges to 2.125), while the latter case does. In deterministic sampling, it can be noticed in (53) that f_{max} also converges to 2 (i.e., $(1 + c)/r + \eta \to 0$) if and only if $r \to \infty$ (simulations not shown for brevity).

We next study the issue of making the random and deterministic models exhibit similar performance.

Corollary 2: Assuming that the random method samples $r_1 \log n$ nodes and the deterministic method samples $r_2 \log n$ nodes, the corresponding upper bounds on f_{max} are equal if

$$r_2 = \frac{(1+c)r_1}{1 + c - r_1 \log\left(1 + \frac{1+c}{r_1}\right)}. \tag{59}$$

Assuming that $d \approx \log n$ and $c \approx 1$, the two methods are equivalent in terms of f_{max} if the deterministic model uses approximately 2.2 times more samples than the random model. Notice, however, that as $r_1 \to \infty$, the deterministic factor r_2 in (59) asymptotically grows as $2r_1^2/(1 + c)$, which increases quite aggressively and quickly voids any benefits (such as the reduced message overhead) obtained by the deterministic method. Therefore, one must conclude that if an application desires bounds on f_{max} very close to 2 (i.e., large r_1), it will typically find the random model more appealing. In other cases when the application needs a "quick and dirty" bound (i.e., small r_1), the deterministic method provides a viable alternative. For example, to achieve $f_{max} \leq 3$ with probability $1 - 1/n$ (i.e., $r_1 = 2$), the deterministic model requires only 3.3 times more samples than the random model.

We apply this analysis in the next section to study the performance of multipoint methods in actual peer-to-peer systems.

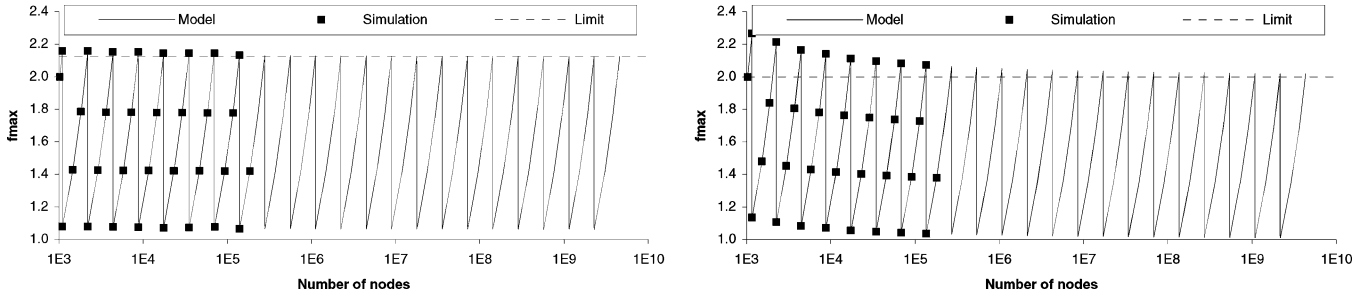
VI. P2P SIMULATIONS

In this section, we briefly analyze the performance of multipoint sampling methods in actual DHTs. We selected de Bruijn graphs for the underlying model since multipoint methods have been proposed mostly in this context and also because the linking rules of this graph provide an interesting platform for observing the effect of large/small zone sizes on node degree.

We implemented a variation of k-regular de Bruijn DHTs borrowing design ideas from [12], [16], [24], and [32]. In all simulations, we use $n = 30\,000$, $k = 8$ (diameter of the graph is 5), and examine the following sampling methods: 1) random sampling of $d = \log n$ points in the DHT [12], [32]; 2) deterministic sampling of approximately $2.2d$ points using a *random walk* along the out-going edges of the graph [1]; and 3) deterministic sampling of the same $2.2d$ points using a *biased walk* [24].

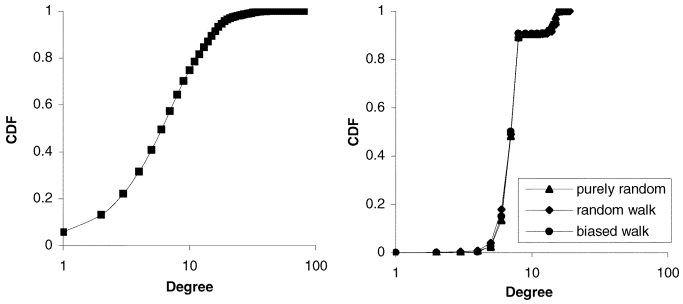
Our main performance metric is the degree distribution of the nodes in the graph after all peers have joined the system. We average our results over 100 simulations and show the resulting distribution of degree below. A baseline example is shown in Fig. 16(a) for the single-point, center-split method. Although the maximum degree 81 is quite rare, there are 5.7% of nodes

[4]Due to the enormous processing capacity required to simulate f_{max} for large n, the simulations in the figure stop at $n = 10^5$.



(a) (b)

Fig. 15. Upper bound on f_{max} with 99.9% confidence in random sampling. The "model" curve refers to the upper bound on f_{max} obtained from (26)and (27) and the "limit" line is the limit of (39) as $n \to \infty$. (a) $d = 8 \log n$. (b) $d = \log^3 n/10$.



(a) (b)

Fig. 16. (a) CDF of degree in de Bruijn DHTs under single-point sampling and center-splits. (b) The same degree distribution in the various multipoint sampling methods.

TABLE VI
FRACTION OF NODES IN THE FINAL GRAPH WITH A CERTAIN DEGREE k

Model	$1 \leq k \leq 3$	$k = 8$	$k > 16$
Random Sampling	0.23%	40.9%	0.13%
Random Walk	0.41%	40.0%	0.22%
Biased Walk	0.001%	40.6%	0.001%

with degree 1, 13% with degree 2 or less, and 22% with degree 3 or less.

In Fig. 16(b), we show the CDF of the degree distribution for the three multipoint methods. We sample $d = 11$ random points in the first method and 24 points in the two deterministic methods. The *random walk* method examines eight neighbors of the original peer and then randomly walks for two hops recording zone sizes of the neighbors of each visited node (i.e., an extension of [2]). The *biased* method does the same, except it always chooses the largest neighbor to walk toward to [24]. After the walk is finished, the largest discovered node is split by the joining peer. As shown in Fig. 16(b), sampling $2.2d$ nodes in the deterministic method approximates the purely random model rather well. Additional results in Table VI confirm this observation and also show that the biased walk performs better than the other two methods at removing the extreme values (i.e., below 4 and above 16) of degree k from the graph.

We finally analyze the message overhead involved in the three methods. The join overhead of the purely random method is approximately 55 messages, while the same metric in the other two approaches is only 7 as long as each peer maintains a list of zone sizes held by its current neighbors.

VII. CONCLUSION

We examined the distribution of the maximum and minimum zone sizes in peer-to-peer networks and derived tight bounds for these metrics. We found that deterministic sampling performed worse than purely random sampling and that both methods could reach $f_{max} = 2$ using a super-logarithmic sampling size. Future work involves analysis of the height of split-trees under multipoint sampling and design of greedy algorithms for the random walk that can improve the balancing performance of existing deterministic methods.

REFERENCES

[1] K. Aberer, A. Datta, and M. Hauswirth, The quest for balancing peer load in structured peer-to-peer systems EPFL, Tech. Rep., 2003.
[2] M. Adler, E. Halperin, R. M. Karp, and V. V. Vazirani, "A stochastic process on the hypercube with applications to peer-to-peer networks," in *Prooc. ACM STOC*, Jun. 2003, pp. 575–584.
[3] J. Aspnes, Z. Diamadi, and G. Shah, "Fault-tolerant routing in peer-to-peer systems," in *ACM PODC*, Jul. 2002, pp. 223–232.
[4] Y. Azar, A. Broder, A. Karlin, and E. Upfal, "Balanced allocations," *SIAM J. Comput.*, vol. 29, pp. 180–200, Jul. 1999.
[5] J. Byers, J. Considine, and M. Mitzenmacher, "Geometric generalizations of the power of two choices," in *ACM SPAA*, Jun. 2004, pp. 54–63.
[6] J. Byers, J. Considine, and M. Mitzenmacher, "Simple load balancing for distributed hash tables," in *IPTPS*, Feb. 2003, pp. 80–87.
[7] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making gnutella-like P2P systems scalable," in *ACM SIGCOMM*, Aug. 2003, pp. 407–418.
[8] R. Corless, G. Gonnet, D. Hare, D. Jeffrey, and D. Knuth, "On the Lambert W function," *Adv. Comput. Math.*, vol. 5, pp. 329–359, 1996.
[9] D. A. Darling, "On a class of problems related to the random division of an interval," *Ann. Math. Stat.*, vol. 24, pp. 239–253, Jun. 1953.
[10] L. Devroye, "Law of the iterated logarithm for order statistics of uniform spacings," *Ann. Prob.*, vol. 9, no. 5, pp. 860–867, 1981.
[11] L. Devroye, "A log-log law for maximal uniform spacings," *Ann. Prob.*, vol. 10, no. 35, pp. 863–868, 1982.
[12] P. Fraigniaud and P. Gauron, The content-addressable network D2B CNRS Univ. Paris Sud, Paris, France, Tech. Rep. 1349, 2003.
[13] B. Godfrey, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica, "Load balancing in dynamic structured P2P systems," in *Proc. IEEE INFOCOM*, Mar. 2004, pp. 2253–2262.
[14] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.
[15] H.-K. Hwang, "Asymptotic estimates of elementary probability distributions," *Stud. Appl. Math.*, vol. 99, pp. 393–417, Nov. 1997.
[16] F. Kaashoek and D. Karger, "Koorde: A simple degree-optimal hash table," in *Proc. IPTPS*, Feb. 2003, pp. 98–107.
[17] D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web," in *ACM STOC*, May 1997, pp. 654–663.
[18] D. Karger and M. Ruhl, "New algorithms for load balancing in peer-to-peer systems," in *IRIS Student Workshop*, Aug. 2003.

[19] R. M. Karp, M. Luby, and F. Meyer auf der Heide, "Efficient PRAM simulation on a distributed memory machine," in *ACM STOC*, May 1992, pp. 318–326.
[20] C. Knessl and W. Szpankowski, "Limit laws for heights in generalized tries and Patricia tries," *J. Alg.*, vol. 44, pp. 63–97, 2002.
[21] T. G. Kurtz, "Solutions of ordinary differential equations as limits of pure jump Markov processes," *J. Appl. Prob.*, vol. 7, pp. 49–58, 1970.
[22] C. Law and K.-Y. Siu, "Distributed construction of random expander graphs," in *Proc. IEEE INFOCOM*, Mar. 2003, pp. 2133–2143.
[23] D. Liben-Nowell, H. Balakrishnan, and D. Karger, "Analysis of the evolution of peer-to-peer networks," in *ACM PODC*, Jul. 2002, pp. 233–242.
[24] D. Loguinov, A. Kumar, V. Rai, and S. Ganesh, "Graph-theoretic analysis of structured peer-to-peer systems: Routing distances and fault resilience," in *ACM SIGCOMM*, Aug. 2003, pp. 395–406.
[25] W. Magnus, F. Oberhettinger, and R. P. Soni, *Formulas and Theorems for the Special Functions of Mathematical Physics*. Berlin, Germany: Springer-Verlag, 1966.
[26] D. Malkhi, M. Naor, and D. Ratajczak, "Viceroy: A scalable and dynamic emulation of the butterfly," in *ACM PODC*, Jul. 2002, pp. 183–192.
[27] M. Mitzenmacher, "Load balancing and density dependent jump Markov processes," in *IEEE FOCS*, Oct. 1996, pp. 213–222.
[28] M. Mitzenmacher, "Studying balanced allocations with differential equations," *Combin., Prob., Comput.*, vol. 8, pp. 473–482, Sep. 1999.
[29] M. Mitzenmacher, "The power of two choices in randomized load balancing," Ph.D. dissertation, Univ. California, Berkeley, 1996.
[30] A. Mondal, K. Goda, and M. Kitsuregawa, "Effective load-balancing of peer-to-peer systems," in *Data Engineering Workshop*, Mar. 2003.
[31] A. Montresor, H. Meling, and O. Babaoglu, "Messor: Load-balancing through a swarm of autonomous agents," in *Int. Workshop on Agents and Peer-to-Peer Computing*, Jul. 2002.
[32] M. Naor and U. Wieder, "Novel architectures for P2P applications: The continuous-discrete approach," in *ACM SPAA*, Jun. 2003, pp. 50–59.
[33] B. Rais, P. Jacquet, and W. Szpankowski, "Limiting distribution for the depth in patricia tries," *SIAM J. Discrete Math.*, vol. 6, no. 2, pp. 197–213, May 1993.
[34] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in *ACM SIGCOMM*, Aug. 2001, pp. 161–172.
[35] M. Roussopoulos and M. Baker, "Practical load balancing for content requests in peer-to-peer networks," *Distrib. Comput.*, vol. 18, no. 6, pp. 421–434, Jun. 2006.
[36] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems," in *IFIP/ACM Int. Conf. Distributed Systems Platforms*, Nov. 2001, pp. 329–350.
[37] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balarkishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *ACM SIGCOMM*, Aug. 2001, pp. 149–160.
[38] B. Vocking, "How asymmetry helps load balancing," in *IEEE Symp. Foundations of Computer Science*, Oct. 1999, pp. 131–141.
[39] X. Wang, Y. Zhang, X. Li, and D. Loguinov, "On zone-balancing of peer-to-peer networks: Analysis of random node join," in *ACM SIGMETRICS*, Jun. 2004, pp. 211–222.
[40] J. Xu, A. Kumar, and X. Yu, "On the fundamental tradeoffs between routing table size and network diameter in peer-to-peer networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 1, pp. 151–163, Jan. 2004.
[41] B. Y. Zhao, J. D. Kubiatowicz, and A. Joseph, Tapestry: An infrastructure for fault-tolerant wide-area location and routing Univ. California, Berkeley, Tech. Rep., 2001.
[42] Y. Zhu and Y. Hu, "Efficient, proximity-aware load balancing for DHT-based P2P systems," *IEEE Trans Parallel Distrib. Syst.*, vol. 16, no. 4, pp. 349–361, Apr. 2005.
[43] S. Q. Zhuang, B. Y. Zhao, and A. D. Joseph, "Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination," in *ACM NOSSDAV*, Jun. 2001, pp. 11–20.

Xiaoming Wang (S'04) received the B.S. degree in computer science and the M.S. degree in electronic engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 1999 and 2002, respectively. He is currently working toward the Ph.D. degree at Texas A&M University, College Station.

During 2002–2003, he worked for the Samsung Advanced Institute of Technology, Korea. His research interests include peer-to-peer systems, probabilistic analysis of computer networks, and topology modeling.

Dmitri Loguinov (S'99–M'03) received the B.S. degree (with honors) in computer science from Moscow State University, Moscow, Russia, in 1995, and the Ph.D. degree in computer science from the City University of New York, New York, in 2002.

Since 2002, he has been an Assistant Professor of computer science with Texas A&M University, College Station. His research interests include peer-to-peer networks, video streaming, congestion control, Internet measurement, and modeling.