Popularity-Biased Random Walks for Peer-to-Peer Search under the Square-Root Principle

Ming Zhong
Department of Computer Science
University of Rochester
zhong@cs.rochester.edu

Kai Shen
Department of Computer Science
University of Rochester
kshen@cs.rochester.edu

ABSTRACT

The *square-root principle* is known to achieve low search time for peer-to-peer search techniques that do not utilize query routing indices (*e.g.*, query flooding or random walk searches). Under this principle, each object is probed with probability proportional to the square root of its query popularity. Existing search methods realize the square-root principle by using either object replication or topology reconstruction, which may not be desirable for those applications with large, dynamic datasets and limited network bandwidth. We propose a new approach that uses popularity-biased random walks to achieve the square-root principle. With the guidance of the Metropolis algorithm, each step of the random walks is determined based on the content popularities of current neighbors. Compared to previous methods, our approach has comparable search performance at no cost of data movement or topology changes.

1. INTRODUCTION

Distributed hash tables (DHT) such as Chord [14], CAN [11], Pastry [12], and Tapestry [15] can support efficient search over peer-to-peer (p2p) networks. These DHTs require sophisticated query routing indices that may be too expensive to maintain for some applications. Index-free search methods like query flooding [8] and random walks are easier to deploy and maintain. However, they are inefficient in that they often have to probe a large proportion of the network nodes in order to satisfy an object search.

To address such inefficiency, Cohen and Shenker proposed square-root replication strategy to reduce the search time of query flooding [3]. Under square-root replication, the number of replication copies of each data object is proportional to the square root of its popularity. A subsequent work by Lv *et al.* uses the same strategy to achieve low search time for random walks [9]. In addition to replication, topology reconstruction can also realize the square-root principle by making the degree of each node proportional to the square-root of its content popularity [4]. Compared to the original topology, the new topology exploits the heterogeneity among peer popularities and reduces search time by making those peers with high query-answering capabilities more accessible to the community.

However, for those applications with large, dynamic datasets and limited network bandwidth, object replication or topology reconstruction may not be feasible since the maintenance of up-to-date topologies or data replication copies requires considerable communication overhead.

Motivated by this, we propose a new random walk search method that use popularity-biased random walks to achieve the square-root principle at no cost of data movement or topology changes. Intuitively, if one can adjust the number of links to satisfy the square-root principle, the same objective can be achieved by visiting different links with appropriate weights. The theoretical foundation of our algorithm is the Metropolis algorithm [10], which can guide the configuration of biased random walks in a way that they uniquely converge to the square-root popularity distribution — a random walker visits a node with probability proportional to the square-root of its content popularity. Our analytical and simulation results show that the proposed approach can achieve comparable performance with previous search methods.

The rest of this paper describes our popularity-biased random walk search method that achieves the square-root principle. We present the simulation results that compare the performance of our search method and previous approaches. We also evaluate the impact of system parameters (*e.g.*, network topologies and query distributions) on the random walk search performance.

2. BACKGROUND

2.1 The Square-Root Principle

We describe the background on the square-root principle for random walk searches [3, 9]. Consider a network with n peers. Let p_i denote the content popularity of peer i, *i.e.*, p_i is the proportion of the queries into the system that can succeed at peer i. If we assume q_i is the probability that the search process visits peer i at each step, then the probability that the search process hits i at the xth step is $q_i \cdot (1 - q_i)^{x-1}$ ($x \leq T$ due to possible TTLs). Hence the expected number of steps for the search process to hit i is:

$$E_i = \sum_{x=1}^{T} [x \cdot q_i \cdot (1 - q_i)^{x-1}] \approx \frac{1}{q_i}, \quad \text{(for large } T\text{'s)} \qquad (1)$$

To simplify the problem, we assume that each query is satisfied only at a unique peer. Then according to the definition of p_i, we have the expected number of steps for an arbitrary query to succeed:

$$\sum_{i=1}^{n} p_i \cdot \frac{1}{q_i} \qquad (2)$$

Hence the problem is to find $q_1, q_2, ..., q_n$ that minimize Equation (2) subject to $\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} q_i = 1$. By using the classical method of Lagrange multipliers [2], the optimal solution is found:

$$q_i \propto \sqrt{p_i} \Leftrightarrow q_i = \frac{\sqrt{p_i}}{\sum_{j=1}^{n} \sqrt{p_j}}, i = 1, 2, ..., n \qquad (3)$$

For the above optimal solution of q, the expected number of steps for an arbitrary query to succeed after the convergence of random walks is:

$$\sum_{i=1}^{n} p_i \cdot \frac{1}{q_i} = \left(\sum_{i=1}^{n} \sqrt{p_i}\right)^2 \qquad (4)$$

Specifically, for Zipf-like popularity distributions with $p_i \propto (\frac{1}{i})^{\alpha}, \alpha > 0, 1 \leq i \leq n$, the expected search time for the square-root principle is:

$$\left(\sum_{i=1}^{n} \sqrt{p_i}\right)^2 \approx \begin{cases} n \cdot \frac{1-\alpha}{(1-\frac{\alpha}{2})^2} = \Theta(n) & \text{if } 0 < \alpha < 1; \\ 4 \cdot \frac{n}{\log n} = \Theta(\frac{n}{\log n}) & \text{if } \alpha = 1; \\ \frac{n^{2-\alpha}}{C(\alpha)\cdot(1-\frac{\alpha}{2})^2} = \Theta(n^{2-\alpha}) & \text{if } 1 < \alpha < 2; \\ \frac{\log^2 n}{C(\alpha)} = \Theta(\log^2 n) & \text{if } \alpha = 2; \\ \frac{C(\frac{\alpha}{2})^2}{C(\alpha)} = \Theta(1) & \text{if } \alpha > 2\ . \end{cases} \qquad (5)$$

where $C(\alpha)$ ($\alpha > 1$) is the power summation constant for large n's, *i.e.*, $C(\alpha) \approx \sum_{i=1}^{\infty}(\frac{1}{i})^{\alpha}$. For example, $C(2) \approx 1.6449$, $C(3) \approx 1.2021$.

2.2 Biased Random Walks and the Metropolis Algorithm

Let $G = (V, E)$ be an undirected connected graph. A random walk on G starts at node v_0, which is either fixed or drawn from some initial distribution π_0. If the random walk is at node v_t at time step t, then it moves to a neighbor v_{t+1} of node v_t at step $t+1$, chosen randomly with certain probability distribution.

Let π_t denote the distribution of node v_t so that $\pi_t(i) = Prob(v_t = i)$, $i \in V$. Let $P = (P_{i,j}), i, j \in V$, denote the transition matrix of the random walk—$P_{i,j}$ is the probability that the random walk moves from node i to node j in one step. $P_{i,j} = 0$ if nodes i, j are not adjacent. The dynamics of the random walk follows $\pi_{t+1} = \pi_t P = \pi_0 P^{t+1}$.

The following theorem by Doeblin [5] gives sufficient conditions for the unique convergence of random walks.

THEOREM 1. *[5] If P is irreducible and aperiodic, then π_t converges to a unique stationary distribution π such that $\pi P = \pi$ independent of the initial distribution π_0.*

Here P is *irreducible* if and only if for any i, j, there exists a t such that $(P^t)_{i,j} > 0$. P is *aperiodic* if and only if for any i, j the greatest common divisor of the set $\{t : (P^t)_{i,j} > 0\}$ is 1. Intuitively *irreducibility* means that any two nodes are mutually reachable by random walks. *Aperiodicity* means that the graph G is non-bipartite. Aperiodicity can be achieved by introducing self-loop transitions of some positive probability on each node of the graph.

Given the guarantee on the unique convergence of self-loop enabled random walks on undirected connected graphs, the next question is this: *How to define the transition matrix P such that the random walk converges to the desired probability distribution?* The Metropolis algorithm [1, 10] was designed as a standard approach to assign transition probabilities to Markov chains so that they converge to any specified probability distributions.

THEOREM 2. *[1, 10] Let $G = (V, E)$ be an undirected connected graph, and let π be the desired probability distribution. Let d_i denote the degree of node i. For each neighbor j of node i, let*

$$P_{i,j} = \begin{cases} \frac{1}{2} \cdot \frac{1}{d_i} & \text{if } \frac{\pi(i)}{d_i} \leq \frac{\pi(j)}{d_j}; \\ \frac{1}{2} \cdot \frac{1}{d_j} \cdot \frac{\pi(j)}{\pi(i)} & \text{if } \frac{\pi(i)}{d_i} > \frac{\pi(j)}{d_j}. \end{cases}$$

and $P_{i,i} = 1 - \sum_{j \in\ neighbors(i)} P_{i,j}$. Then π is the unique converged stationary probability distribution of the random walk with transition matrix P.

It is easy to prove that π is a converged distribution of the above random walk by verifying $\pi P = \pi$. As explained before, the laziness factor $\frac{1}{2}$, which could be any constant between 0 and 1, ensures that each node has a self-loop and hence the above random walk also uniquely converges to π. Note that the Metropolis algorithm only requires the ratio $\frac{\pi(j)}{\pi(i)}$ without the need for knowing each individual $\pi(i)$ or the normalization factor $\sum_i \pi(i)$.

Our previous work has employed biased random walks to support non-uniform random membership management in peer-to-peer networks [16].

3. BIASED RANDOM WALKS TO REALIZE THE SQUARE-ROOT PRINCIPLE

We propose a new random walk search method that realizes the square-root principle as achieved by object replication [3, 9] and topology reconstruction [4], but without the overhead of adjusting replication copies or network topologies upon dataset changes. Under our method, each query issues a popularity-biased random walker, which travels around the network until the requested object is discovered or the TTL has expired. At each step of the walk, the next hop is chosen from the neighbors of the current node with probabilities biased towards their content popularities.

The Metropolis algorithm [10] can help us configure the bias probabilities at each random walk step such that the random walk converges to the desired node visitation probability distribution. Specifically, we determine our probability-biased random walk as follows.

Let p_i, the peer content popularity, be defined the same as in Section 2.1. Let d_i denote the number of network neighbors for peer i. If a random walker is at peer i at a certain time step, then for each neighbor j of i it moves to j with probability $P_{i,j}$ after next step, where:

$$P_{i,j} = \begin{cases} \frac{1}{2} \cdot \frac{1}{d_i} & \text{if } \frac{\sqrt{p_i}}{d_i} \leq \frac{\sqrt{p_j}}{d_j}; \\ \frac{1}{2} \cdot \frac{1}{d_j} \cdot \frac{\sqrt{p_j}}{\sqrt{p_i}} & \text{if } \frac{\sqrt{p_i}}{d_i} > \frac{\sqrt{p_j}}{d_j}. \end{cases} \qquad (6)$$

and the probability for the random walk does not move at the step $P_{i,i} = 1 - \sum_{k \in\ neighbor(i)} P_{i,k}$. The peer content popularity p_i can be estimated as the number of queries satisfied at peer i divided by the total number of queries received by i [4]. Hence $P_{i,j}$ is locally computable (it only requires the local information of peer i and its adjacent nodes).

The node visitation distribution of the random walks will converge with the increasing number of random walk steps. Let π denote the converged distribution of our random walks (*i.e.*, a random walker visits peer i with probability $\pi(i)$ at any step after the convergence). Then π must satisfy:

$$\pi(i) \propto \sqrt{p_i} \Leftrightarrow \pi(i) = \frac{\sqrt{p_i}}{\sum_{j=1}^{n} \sqrt{p_j}} \qquad (7)$$

This can be easily proved by verifying $\pi \cdot P = \pi$. Moreover, The laziness factor $\frac{1}{2}$ in Equation (6) ensures that each peer has a self-loop and hence the converged distribution is unique (Theorem 1).

According to Equation (3), the converged distribution of our popularity-biased random walks satisfies the square-root principle and hence minimizes the search time. We expect random walks to converge fast on typical p2p network topologies with good expansion properties ($e.g.$, random graphs and random power-law graphs). We will provide quantitative measurement results in Section 4.

Current random walk search methods [9] typically use multiple independent random walkers with the expectation that k independent random walkers after T steps should cover nearly equal number of nodes as one random walker after $k \cdot T$ steps. Hence the search time can be reduced by roughly k times with no extra communication overhead. We will also use multiple independent random walks as defined in Equation (6) and examine their performance variations in our simulation.

4. SIMULATION

We compare the performance of the three different search methods that realize the square-root principle: our popularity-biased random walks, object replication, and topology reconstruction. Although object replication and topology reconstruction can be used to support both query flooding searches and random walk searches, our simulation study focuses on the performance of the three methods supporting random walk searches.

It is important to keep in mind that object replication and topology reconstruction must maintain up-to-date replication copies and network topologies to achieve the search performance presented in this section. Our popularity-biased random walks do not incur such costs. These costs can be very substantial in large networks with dynamic searchable datasets.

In addition to the performance comparison, another objective of the simulation is to evaluate the impact of system parameters ($e.g.$, network topologies and query distributions) on the random walk search performance.

4.1 Simulation Setup

The three search methods are set up as follows.

1. *Square-root replication.* Each object is replicated randomly over the network in a way that the number of replication copies is proportional to the square-root of its popularity. One uniform random walker is used for searching the network while we set the average number of replication copies as the number of random walkers used in *square-root topology* and *square-root biased walks*. This is intended to make a fair comparison since the expected search time for square-root replication is inversely proportional to the average number of replication copies.

2. *Square-root topology.* Uniform random walkers are used to search the network. The degree of each node is proportional to the square root of its content popularity. To transform the original topology into this square-root topology, we compute the node degree sequence and use the PLRG algorithm [7] to generate the new randomized topology with the desired node degree sequence.

3. *Square-root biased walks.* Without adjusting topologies or replicating data, each query issues a number of random walkers that travel the network according to Equation (6). Multiple random walkers coordinate with each other by periodically calling back the source to learn whether any other walker has found the target. If so, the remaining walkers will terminate upon next call.

System parameters	Value
Network size	20,000
Number of objects	1,000,000
Number of queries	100,000
Average per-link latency	50 ms

Table 1: Default simulation parameters.

In our simulation, the query popularity follows Zipf-like distribution (the frequency of the ith most popular query is proportional to $\frac{1}{i^\alpha}$). Specifically, we choose $\alpha = 0.6$ and $\alpha = 1.2$ based on Sripanidkulchai's measurement results on Gnutella traces [13].

We use random graphs and random power-law graphs as network topologies in our simulation. Random graphs represent those p2p topologies where new links are made independent of existing node degrees. Random power-law graphs represent those networks where new links are more likely attached to nodes with large degrees. In our simulation, the random power-law graphs are generated by using the PLRG algorithm [7]. We use the random power-law graphs with $\alpha = 0.8$, following Lv *et al.*'s simulation setup [9]. We generate random graphs by connecting each new nodes to some nodes selected uniformly at random from all existing nodes.

Some default simulation parameters are listed in Table 1.

4.2 Simulation Results

Figures 1 and 2 present the search time and communication overhead on different network topologies (random graphs and random power-law graphs), query popularity distributions, and a variety of settings on the random walker count (k). We observe that the three random walk search methods have similar performance with small variations (average 14% difference for random graphs and 19% for random powerlaw graphs). The performance difference is mainly due to the different speeds at which the three methods converge to the targeted square-root object probe distribution.

We perform measurements to specifically measure the random walk convergence speeds. Figures 3 and 4 show that the square-root replication has better convergence speed than the square-root biased walks which in turn has better convergence speed than the square-root topology. Due to its non-uniformity, our popularity-biased random walks has slower convergence and hence lower search performance than square-root replication. The square-root topology has the slowest convergence speed because the square-root network topologies tend to have worse expansion properties than typical topologies such as random graphs and random power-law graphs.

Based on the results in Figures 1 and 2, we also examine the impact of system parameters on the random walk search performance. In terms of the query popularity distribution, we find that the search performance for high-skewness popularity distributions ($\alpha = 1.2$) is higher than that for low-

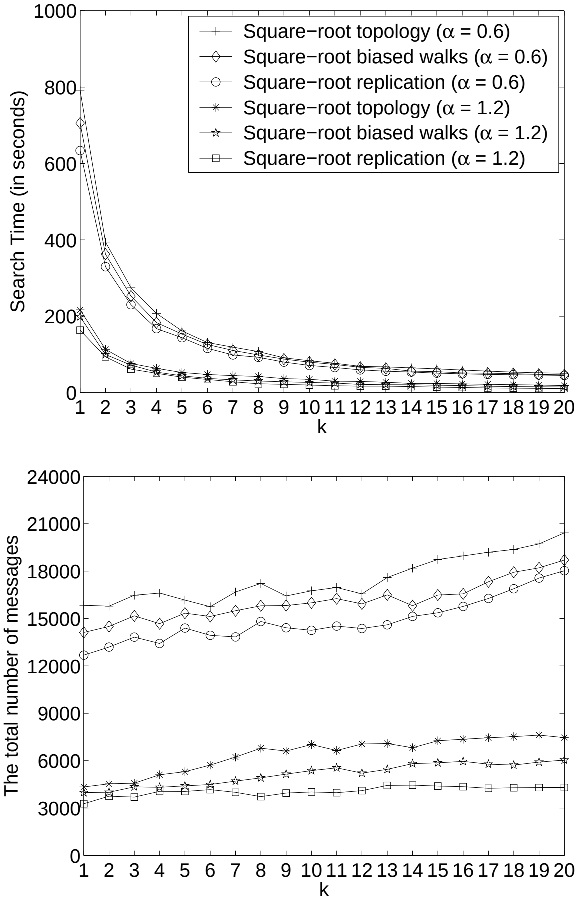

Figure 1: The search time and communication overhead on random graphs, where k is the number of random walkers (the average number of replication copies for square-root replication).

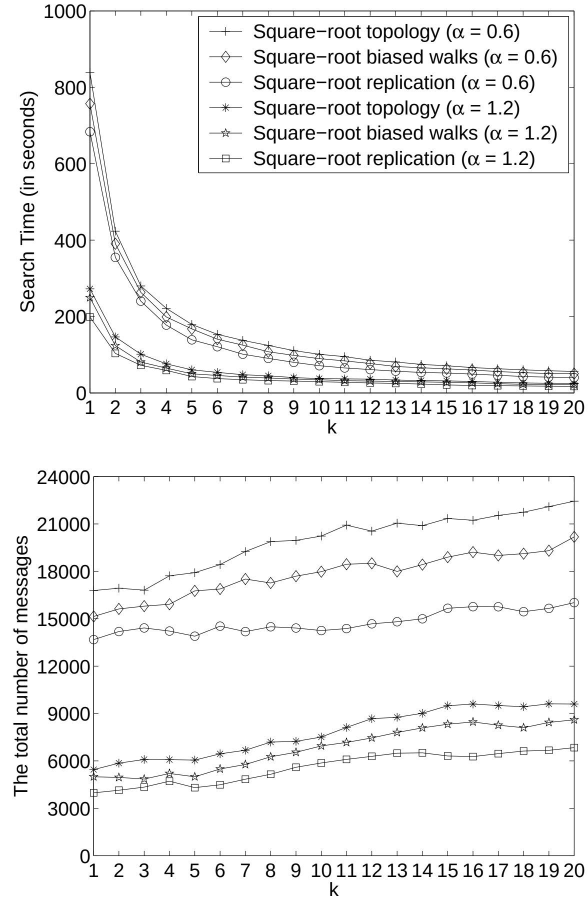

Figure 2: The search time and communication overhead on random power-law graphs, where k is the number of random walkers (the average number of replication copies for square-root replication).

skewness distributions ($\alpha = 0.6$). This can be explained by results in Equation (5), which show that the search time for $\alpha = 1.2$ is $\Theta(n^{0.8})$ and the time for $\alpha = 0.6$ is $\Theta(n)$.

Slightly higher time and communication overhead are observed for random power-law graphs than those for random graphs, which is mainly because random graphs have slightly better expansion properties than random power-law graphs [6]. Consequently random walks tend to converge faster on random graphs.

The simulation results also suggest that increasing the number of random walkers can significantly reduce the search time with slight increase in the communication overhead. Such increase is due to the convergence overhead associated with each random walker (so more walkers would incur more overhead).

5. CONCLUSION

This paper proposes a new index-free p2p search technique that uses popularity-biased random walks to realize the square-root principle. Compared to previous approaches that realize this principle, our method does not incur any cost of object replication or topology adjustments. Our analytical and simulation results show that the new technique can achieve search performance comparable to previous techniques using object replication and topology reconstruction.

Acknowledgments

We would like to thank the anonymous referees for their valuable comments. This work was supported in part by the National Science Foundation grants CCR-0306473, ITR/IIS-0312925, and NSF CAREER Award CCF-0448413.

REFERENCES

[1] Y. Azar, A. Broder, A. Karlin, N. Linial, and S. Phillips. Biased Random Walks. In *Proc. of the 24th ACM Symposium on the Theory of Computing*, pages 1–9, 1992.

[2] D. Ballard. *An Introduction to Natural Computation*. MIT Press, 1997.

[3] E. Cohen and S. Shenker. Replication Strategies in Unstructured Peer-to-Peer Networks. In *Proc. of ACM SIGCOMM*, Pittsburgh, PA, August 2002.

[4] B. F. Cooper. Quickly Routing Searches Without Having to Move Content. In *Proc. of IPTPS*, 2005.

[5] W. Doeblin. Exposé de la théorie des chaînes simples constantes de Markov á un nombre fini d'états. *Mathématique de l'Union Interbalkanique*, 2:77–105,

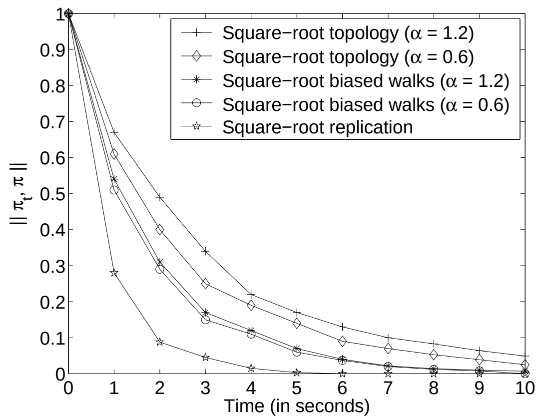

Figure 3: The convergence speed on random regular topologies. π_t denotes the distribution of a random walk at time t. π is the converged distribution of a random walk. $||\pi_t, \pi|| = \frac{1}{2} \cdot \sum_i |\pi_t(i) - \pi(i)|$ measures the difference between π_t and π. Note that $0 \leq ||\pi_t, \pi|| \leq 1$.

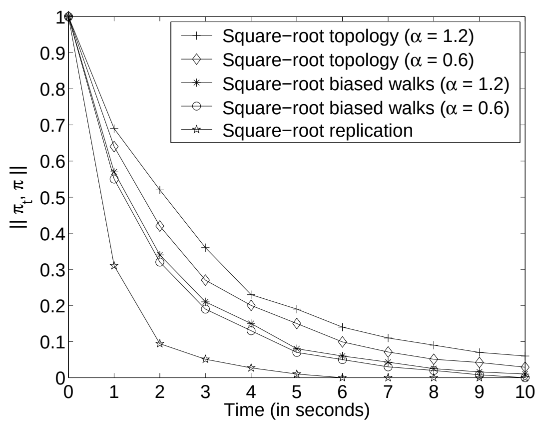

Figure 4: The convergence speed on random power-law topologies.

1938.

[6] C. Gkantsidis, M. Mihail, and A. Saberi. Conductance and congestion in power law graphs. In *Proc. of the ACM SIGMETRICS*, 2003.

[7] C. Gkantsidis, M. Mihail, and E. Zegura. The markov chain simulation method for generating connected power law random graphs. In *Proc. 5th Workshop on Algorithm Engineering and Experiments (ALENEX)*, 2003.

[8] Gnutella. http://www.gnutella.com.

[9] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and Replication in Unstructured Peer-to-Peer Networks. In *Proc. of ICS'02*, 2002.

[10] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[11] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-Addressable Network. In *Proc. of ACM SIGCOMM*, pages 161–172, San Diego, CA, August 2001.

[12] A. Rowstron and P. Druschel. Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems. In *Proc. of IFIP/ACM Middleware Conf.*, pages 329–350, Heidelberg, Germany, November 2001.

[13] K. Sripanidkulchai. The popularity of Gnutella queries and its implications on scalability. In *The O'Reilly Peer-to-Peer and Web Services Conference*, 2001.

[14] I. Stoica, R. Morris, D. Karger, M. Frans Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. In *Proc. of ACM SIGCOMM*, pages 149–160, San Diego, CA, August 2001.

[15] B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing. Technical Report UCB/CSD-01-1141, Computer Science Division, U.C. Berkeley, April 2001.

[16] M. Zhong, K. Shen, and J. Seiferas. Non-uniform Random Membership Management in Peer-to-Peer Networks. In *Proc. of IEEE INFOCOM*, Miami, FL, March 2005.