Non-uniform Random Membership Management in Peer-to-Peer Networks

Ming Zhong Kai Shen Joel Seiferas
Department of Computer Science, University of Rochester
Rochester, NY 14627-0226, USA
Email: {zhong, kshen, joel}@cs.rochester.edu

***Abstract*— Existing random membership management algorithms provide each node with a small, uniformly random subset of global participants. However, many applications would benefit more from non-uniform random member subsets. For instance, non-uniform gossip algorithms can provide distance-based propagation bounds and thus information can reach nearby nodes sooner. In another example, Kleinberg shows that networks with random long-links following distance-based non-uniform distributions exhibit better routing performance than those with uniformly randomized topologies.**

In this paper, we propose a scalable non-uniform random membership management algorithm, which provides each node with a random membership subset with application-specified probability distributions—*e.g.*, with probability inversely proportional to distances. Our algorithm is the first non-uniform random membership management algorithm with proved convergence and bounded convergence time. Moreover, our algorithm does not put specific restrictions on the network topologies and thus have wide applicability.

I. Introduction

A membership management algorithm, which provides each node with run-time peer sampling service, is essential for many peer-to-peer (p2p) network applications, such as gossip-based broadcast algorithms [1], [2], distributed hash tables [3], [4], dynamic load balancing [5], random sampling [6], and network topology construction [7]. Full membership management maintains the complete list of all network members at each node. The storage and communication requirements of such full membership management algorithms grow linearly with the network size, which is prohibitive for large-scale applications. Motivated by this, a number of membership subset management algorithms [6], [7], [8], [9] have been proposed recently. The key common feature of these algorithms is that each node maintains a small, dynamically changing, random membership subset with uniform representation of network members. The scalability is achieved in that per-node random membership subsets grow much more slowly than the full network. For many applications, making decisions based on random membership subsets has comparable performance with knowing the complete membership list.

These earlier studies focused on the maintenance of uniform random membership subsets. However, *non-uniform* random membership subsets are more desirable for some applications. Examples include the following:

This work was supported in part by the National Science Foundation grants CCR-0306473 and ITR/IIS-0312925.

- *Gossips with distance-based propagation time bounds*. Gossip-based broadcast algorithms [1], [2] provide a robust and scalable mechanism for distributed information dissemination. In many network environments, new information is more "interesting" to nodes that are nearby. Kempe *et al.* [10] show that with a carefully chosen probability distribution, their non-uniform gossip algorithms can provide distance-based propagation time bounds and thus information can reach nearby nodes sooner. Their algorithms assume the existence of a scalable mechanism to identify random nodes with a given distance-based probability distribution.
- *Randomized distributed hash table topologies*. Recently, several randomized distributed hash table topologies [3], [4] have been proposed to achieve good trade-offs between the space overhead and the lookup latency. These algorithms have been motivated by Kleinberg's result [11] that by having $O(1)$ non-uniform random links per node, it is possible to route lookups with an average latency of $O(\log^2 n)$ hops by greedy algorithms. Non-uniform random membership subsets are instrumental for easy creation and maintenance of non-uniform random links at each node.
- *Dynamic load balancing*. One of the key issues for dynamic load balancing in p2p networks is to find *hotspots* (highly overloaded nodes) and reassign their loads to others. Random membership subsets with load-based probability distributions (*e.g.*, choose a node with probability proportional to its load) are more likely to find hotspots than uniform sampling used in current p2p load balancing algorithms [5], especially when there is a small number of very highly loaded nodes in the network—*e.g.*, when the load distribution is power-law.
- *Non-uniform random data sampling and collection*. In many applications such as resource discovery, network failure detection, environmental data measurement and military sensor networks, the importance of the collected data is related to the distance between a data producer and a data collector [12]. For instance, a nearby copy of a requested resource item may be more favored in resource discovery. As another example, an application may care more about node or link failures in the neighborhood than failures occurring far away. Non-uniform random membership subsets can serve as the base set for sampling

nodes with non-uniform distributions.

A naive approach for constructing non-uniform random membership subsets is to probabilistically select nodes from uniform random subsets. However, the uniform random subsets may not contain sufficient candidates for further probabilistic selection. Part of the reason is that in order to achieve scalability, uniform random subsets [6], [7], [8], [9] are usually small, of a constant size or of size $O(\log n)$, where n is the network size. For example, if a node's desired non-uniform membership subset specifies that it should only contain nodes among its $\frac{n}{\log n}$ closest neighbors and it attempts to choose this subset from its uniform random membership subset with size $O(\log n)$, then with probability $(1 - \frac{1}{\log n})^{\log n} \approx \frac{1}{e}$ for large n's, it will have none of the $\frac{n}{\log n}$ closest neighbors in its uniform random membership subset. This problem becomes more severe for more skewed target distributions and for smaller per-node uniform subsets.

Thus we view direct non-uniform random membership management as necessary. We list a number of desired properties for a non-uniform random membership management algorithm:

1) *Scalability*. The per-node storage and communication overhead of non-uniform random membership management algorithms should grow slowly with the network size.
2) *Customization*. Non-uniform random membership management algorithms should be able to generate per-node random subsets with application-specified non-uniform probability distributions.
3) *Correctness*. It is desirable to have proved consistency between the application-specified non-uniform probability distribution and the probability distribution of the random subsets generated by an algorithm.
4) *Bounded convergence time*. After network structure changes, the random membership subsets should converge quickly to adapt to the new network structure. This property also indicates an algorithm's resilience to network failures.
5) *Topology independence*. Non-uniform random membership management algorithms should not put specific restrictions on applicable network topologies.

A. Overview of Our Results

In this paper, we propose a non-uniform random membership management algorithm with the desirable properties described above. Our algorithm provides each node with a random membership subset satisfying application specified probability distributions (*e.g.*, choose a node with probability inversely proportional to its distance to the current node). In our algorithm, global knowledge of network size or node ID distribution is not necessary. The sizes of per-node random membership subsets are independently decided by each node and can be adjusted at runtime. The communication overhead of our algorithm is moderate compared with the current uniform random membership management algorithms [6], [7], [8], [9]. The message complexity of our algorithm is $\Theta(n)$ for each time step, where n is the network size.

In our algorithm, random membership subsets are generated by biased random walks. Guided by the Metropolis algorithm [13], [14], our biased random walks can generate random membership subsets with arbitrary probability distributions. Our algorithm does not put specific restrictions on network topologies and it can be applied to all peer-to-peer networks, such as rings, tori, random regular graphs and power law graphs. For these topologies, we provide provable upper bounds for the convergence time of our algorithm for distance-based probability distributions. Our proof techniques can be applied to other probability distributions and network topologies. Along with asymptotic bounds, we also provide simulation results to quantitatively assess the algorithm convergence time at typical settings.

B. Related Work

To the best of our knowledge, the only direct approach for choosing non-uniform random peers in p2p networks is due to Manku *et al.* [3], [4]. Their approach is specifically designed for supporting Chord [15]-like distributed hash tables, and therefore it has limited applicability. In particular, this approach only works with ring topologies and it relies on the assumption that node IDs are evenly distributed in the network address space.

A number of previous studies proposed scalable random membership management algorithms with *uniform* representation of network members, such as Saxons [7], lpbcast [8], and SCAMP [9]. They have low communication overhead for large networks and they work for arbitrary network topologies. Some [8], [9] also have analytical results on the membership information propagation speed. However, it is not clear how these algorithms can be adapted for supporting *non-uniform* random membership management.

Kostić *et al.* proposed a random membership subset service for tree-shaped network topologies [6]. It employs an epoch-based gather-scatter algorithm to distribute membership information with uniform randomness. However, this algorithm can not be applied to more general mesh-like network structures.

King and Saia proposed a distributed algorithm which, with high probability, always chooses a node uniformly at random from the set of nodes in distributed hash tables [16]. However, their algorithm only works for ring topologies.

C. Organization of the Paper

The remainder of this paper is organized as follows. In Section II, we introduce the theoretical background of our algorithm. Section III proposes our algorithm for generating random membership subsets with distance-based probability distributions. We prove the correctness of our algorithm, and analyze the asymptotic bound for its convergence time over several common peer-to-peer topologies. In section IV, we present the simulation results for our random membership subset algorithm. We conclude and identify open problems in Section V.

II. Background: Random Walks and the Metropolis Algorithm

This section introduces the theoretical background of our 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 [17] gives sufficient conditions for the convergence of random walks.

Theorem 1: 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.

A. The Metropolis Algorithm

Given the guarantee on the 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 will converge to the desired probability distribution?* The Metropolis algorithm was designed as a standard approach to assign transition probabilities to Markov chains so that they converge to any specified probability distributions.

Theorem 2: [13], [14] 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.

Theorem 2 can be proved by verifying $\pi P = \pi$. The Metropolis algorithm only requires knowing the ratio $\frac{\pi(j)}{\pi(i)}$. The normalization factor $\sum_i \pi(i)$ is unnecessary. The laziness factor $\frac{1}{2}$ ensures that each node has a self-loop and thus P is *aperiodic*. A random walk configured by the Metropolis algorithm is time-reversible in the sense that $\forall i, j, \pi(i)P_{i,j} = \pi(j)P_{j,i}$.

B. The Convergence Time of the Metropolis Algorithm

The Metropolis algorithm guarantees that an appropriately configured random walk converges to the desired probability distribution. The next question to ask is how quickly π_t converges to π.

Definition 1: The *total variation difference* between π_t and π is $\|\pi_t, \pi\| = \frac{1}{2} \max_{v_0} \sum_i |\pi_t(i) - \pi(i)|$.

The total variation difference measures the difference between two probability distributions. It is maximized over all possible starting nodes $v_0 \in V$. The total variation difference is at most 1.

Definition 2: For $\epsilon > 0$, the *mixing time* is defined as $\tau(\epsilon) = \min\{t : \forall t' \geq t, \|\pi_{t'}, \pi\| \leq \epsilon\}$.

The mixing time measures the time for π_t to converge to π. Diaconis and Stroock [18] proved the following mixing time bound:

Theorem 3: Let $\pi_{\min} = \min_i \pi(i)$, then $\tau(\epsilon) \leq \Delta_P^{-1} \log((\pi_{\min}\epsilon)^{-1})$. Here Δ_P is the eigengap of the transition matrix P.

It is known that P has $|V|$ eigenvalues $\lambda_1, \lambda_2, ..., \lambda_{|V|}$ such that $1 = \lambda_1 > |\lambda_2| \geq ... \geq |\lambda_{|V|}|$. The eigengap of P is defined as $\Delta_P = 1 - |\lambda_2|$, which provides a bound for the mixing time. A larger eigengap means shorter convergence time. However, for large-scale p2p network applications, the sizes of transition matrices are so large that it is very difficult to compute exact eigenvalues and eigengaps. Several approaches [18], [19], [20] have been proposed for establishing bounds for eigengaps of transition matrices. In this paper, we compute the eigengap bounds by using the *canonical path* approach [19].

The main idea of the canonical path approach is this: slow (exponential time) mixing is characterized by a bad (exponentially small) cut in the graph, since it takes exponential time for the probability flow to move from one side of the cut to the other, to reach the equilibrium. Thus the minimum cut (max-flow) in the probability transition graph provides a bound for the mixing time.

Let π be the unique converged distribution. P is the transition matrix of the random walk. Let the edge capacity $Q(e) = \pi(x)P_{x,y} = \pi(y)P_{y,x}$. For distinct nodes x, y in the graph $G = (V, E)$, a *canonical path* γ_{xy} refers to a path between x, y. Γ, a family of canonical paths, includes exactly one path for each pair of distinct nodes x, y: $\Gamma = \{\gamma_{xy} : x, y \in V, x \neq y\}$. The *congestion* of Γ is defined as:

$$\rho(\Gamma) = \max_e \frac{1}{Q(e)} \sum_{\gamma_{xy} \ni e} \pi(x)\pi(y).$$

Intuitively, the path γ_{xy} carries flow $\pi(x)\pi(y)$. $Q(e)$ represents the capacity of the edge e. A canonical path family Γ represents a flow scheme for the pairs of distinct nodes in the network. $\rho(\Gamma)$ is the maximum flow/capacity ratio of the canonical path family Γ. A canonical path family with low congestion means that the graph lacks small cuts and the random walks mix quickly.

Let $\bar{\rho} = \min_\Gamma \rho(\Gamma)l(\Gamma)$, where $l(\Gamma)$ is the maximum length of a path in Γ. $\bar{\rho}$ chooses the canonical path family with the minimum congestion, which provides a lower bound for the eigengap of the transition matrix P:

Theorem 4: [19] $\Delta_P \geq \bar{\rho}^{-1}$.

The bound for the mixing time can be achieved by combining Theorem 3 and Theorem 4:

Theorem 5: [19] $\tau(\epsilon) \leq \bar{\rho} \log((\pi_{\min}\epsilon)^{-1})$.

In summary, this approach aims to find a canonical path family Γ with low congestion, which provides bounds for the mixing time of the random walks.

III. Non-uniform Random Membership Management

Our non-uniform random membership management algorithm aims to provide each node in p2p networks with a random membership subset satisfying application-specified probability distributions. Let π_i denote the desired probability distribution for node i where $\pi_i(j)$ is the probability that a uniformly chosen member from i's membership subset is node j. The basic framework of our algorithm is as follows:

Suppose each node i of the p2p network maintains a membership subset with size k_i, determined independently by i based on the available network bandwidth and space. Node i initiates k_i independent random walks $R_{i,1}, R_{i,2}, ..., R_{i,k_i}$ configured by the Metropolis algorithm such that the random walks converge to π_i. Whenever visited by a random walk $R_{i,l}$, node j will contact node i such that node i updates the lth member of i's membership subset with j. After the random walks converge, the probability that a uniformly chosen member from node i's membership subset is node j is $\pi_i(j)$, which satisfies the desired distribution.

Intuitively speaking, a node constructs its membership subset by sampling nodes with the desired distribution. If node i has not been contacted for a long time with regard to a random walk $R_{i,l}$, then it decides that $R_{i,l}$ is lost and re-initiates $R_{i,l}$. At any time, there are $\sum_{i \in V} k_i = \Theta(|V|)$ in transit random walk messages in the network, which is moderate compared with the current uniform membership management algorithms [6], [7], [8], [9].

The key point for the above framework is to ensure that the random walks are configured correctly and have traveled for large enough number of steps for convergence. We use the Metropolis algorithm to configure our random walks such that they converge to the desired distributions. The mixing time (*i.e.*, convergence time) of the configured random walks varies with different network topologies and membership subset distributions. Due to the hardness of eigenvalue computation, few results have been achieved on bounding the mixing time of non-uniform random walks. Thus it is non-trivial work to analyze the mixing time of non-uniform random walks for specific network topologies and membership subset distributions.

In the remainder of this section, we present the implementation of the above algorithm framework for distance-based probability distributions such as those in the contexts of gossip-based broadcast algorithms [10] and randomized distributed hash tables [4], [11]. We also give the analytical results on the mixing time of our random walks in some common p2p network topologies.

A. Random Membership Subsets with Distance-based Distributions

The distance-based probability distributions as specified in [4], [10], [11] require that node i chooses node $j \neq i$ with probability proportional to $d(i,j)^{-\alpha}$. In other words, $\pi_i(j) \propto d(i,j)^{-\alpha}$ or $\pi_i(j) = \frac{d(i,j)^{-\alpha}}{\sum_{x \neq i} d(i,x)^{-\alpha}}$, where $d(i,j)$ can be defined as either the hop distance from i to j or the Euclidean distance between them. Note that $\pi_i(i) = 0$. The constant α usually refers to the dimensionality of the network topology [11], which is a small natural number.

Based on the Metropolis algorithm, a random walk initiated from a node $i \in V$ is defined as follows:

For each neighbor v_u of the initiating node i, we have $P_{i,v_u} = \frac{1}{Deg(i)}$.

If the random walk is at node v_t at time step t, then for each neighbor v_u of v_t, moves to v_u with probability P_{v_t,v_u}, where

$$P_{v_t,v_u} = \begin{cases} 0 & \text{if } v_u = i; \\ \frac{1}{2} \cdot \frac{1}{d_t} & \text{if } \frac{d(i,v_t)^{-\alpha}}{d_t} \leq \frac{d(i,v_u)^{-\alpha}}{d_u}; \\ \frac{1}{2} \cdot \frac{1}{d_u} \cdot \frac{d(i,v_u)^{-\alpha}}{d(i,v_t)^{-\alpha}} & \text{if } \frac{d(i,v_t)^{-\alpha}}{d_t} > \frac{d(i,v_u)^{-\alpha}}{d_u}. \end{cases}$$

and $P_{v_t,v_t} = 1 - \sum_{v_u \in \ neighbor(v_t)} P_{v_t,v_u}$.

Here d_x denotes the number of neighbors of node v_x, where the neighbors are counted by viewing each link as bidirectional. The random walk views the graph as undirected and is able to make backward steps across directed links. The random walk is self-avoiding, *i.e.*, never returns to i, since $\pi_i(i) = 0$.

Based on the above random walk, we present our basic membership management protocol with the following components:

- *Node Joining*. In most p2p network applications, a node i joins the network by connecting to some initial network neighbors. After the neighbors are determined, node i initiates k_i independent random walks $R_{i,1}, R_{i,2}, ..., R_{i,k_i}$ as defined above, where k_i is the size limit of node i's membership subset. Each random walk also has a TTL threshold after which the random walk message expires.
- *Membership Subset Maintenance*. Whenever node j receives a random walk $R_{i,l}$ initiated from node i, node j sends its own identity to node i. Upon receiving j's identity referred by $R_{i,l}$, node i updates the lth member of its membership subset with j.
- *Node Departure*. In our protocol, a departing node simply leaves the network without doing anything. Its membership in the subsets of other nodes will be purged out eventually following our failure processing mechanism described below.
- *Failure Processing*. A random walk may be lost due to TTL expiration, link failure, node failure, or node departure described above. If node i has not received any membership information referred by $R_{i,l}$ (a random walk

initiated from node i) for a long time, then node i decides that $R_{i,l}$ is lost and re-initiates $R_{i,l}$.

Theorem 6: [**Correctness**] For each node $i \in V$, after the random walks initiated from i converge, the probability that a uniformly chosen member from node i's membership subset is j is proportional to $d(i,j)^{-\alpha}$.

Theorem 6 directly follows the correctness of the Metropolis algorithm, which ensures that each random walk initiated from node i selects node j for i's membership subset with desired probability after convergence.

Given the assurance that the generated membership subsets converge to the desired distribution, the next question is how fast do they converge? In subsequent subsections, we provide analytical results on the bounds of mixing time for major peer-to-peer network topologies.

B. The Mixing Time in Structured P2P Topologies

We present the analytical results on the mixing time of our random walks in unidirectional rings (used in Chord [15]) and unidirectional d-dimensional tori (used in CAN [21]). For structured p2p topologies like rings and tori, the node distance $d(i,j)$ is often measured by hop distance, *i.e.*, the minimum number of hops to go from node i to node j.

Our algorithm on a unidirectional ring of $n+1$ nodes can be viewed as a random walk, starting from the initiating node 0, on a path with $n+1$ nodes labeled $0, 1, ..., n$, where node i is the predecessor of node $i+1$ in the original ring. Thus the hop distance from node 0 to node i in the original ring is $d(0,i) = i$. Fig. 1 shows the transition probabilities between nodes i and its neighbors, which are determined based on the random walk definitions in Section III-A.

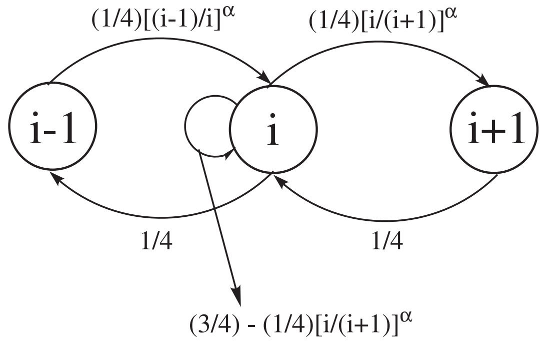

Fig. 1. The transition probabilities in the neighborhood of node i. The hop distance from node 0 to node i is i. Note that i is decided by the random walk at runtime since i is the number of the forward steps (from a node to its successor) taken by the random walk. Thus the transition probabilities can be decided at runtime without the global knowledge of n.

We use the canonical path approach [19] (explained in Section II-B) to analyze the mixing time of the random walk. We first define a canonical path family Γ such that γ_{xy} is the path between x, y without passing by the initiating node 0.

Let π be the converged distribution of the random walk, then $\pi(i)$ is proportional to $d(0,i)^{-\alpha} = i^{-\alpha}$. Hence for large n's:

$$\pi(i) = \frac{i^{-\alpha}}{\sum_{j=1}^{n} j^{-\alpha}} \approx \begin{cases} \frac{1}{(i \ln n)} & \text{if } \alpha = 1; \\ \frac{1}{c(\alpha) i^{\alpha}} & \text{if } \alpha \geq 2. \end{cases}$$

$c(\alpha)$ is the power summation $\sum_{j=1}^{\infty} j^{-\alpha}$, *e.g.*, $c(2) \approx 1.6449$, $c(3) \approx 1.2021$.

To compute Γ's congestion, $\rho(\Gamma)$, we consider an arbitrary edge $e = (i, i+1)$, $i \in \{1, 2, ..., n-1\}$. We need to know the node pairs x, y that can be routed through the edge $e = (i, i+1)$. These include all $x \in \{1, 2, ..., i\}$ and $y \in \{i+1, i+2, ..., n\}$.

Hence,

$$\sum_{\gamma_{xy} \ni e} \pi(x)\pi(y) = \begin{cases} \frac{1}{\ln^2 n} \sum_{x=1}^{i} \sum_{y=i+1}^{n} \frac{1}{xy} & \text{if } \alpha = 1; \\ \frac{1}{c(\alpha)^2} \sum_{x=1}^{i} \sum_{y=i+1}^{n} \frac{1}{(xy)^{\alpha}} & \text{if } \alpha \geq 2. \end{cases}$$

$$= \begin{cases} \frac{1}{\ln^2 n} \sum_{x=1}^{i} \frac{1}{x} \sum_{y=i+1}^{n} \frac{1}{y} & \text{if } \alpha = 1; \\ \frac{1}{c(\alpha)^2} \sum_{x=1}^{i} \frac{1}{x^{\alpha}} \sum_{y=i+1}^{n} \frac{1}{y^{\alpha}} & \text{if } \alpha \geq 2. \end{cases}$$

$$\leq \begin{cases} \frac{1}{\ln^2 n} \sum_{x=1}^{i} \frac{1}{x} \cdot (n-i) \cdot \frac{1}{i+1} & \text{if } \alpha = 1; \\ \frac{1}{c(\alpha)^2} \sum_{x=1}^{i} \frac{1}{x^{\alpha}} \cdot (n-i) \cdot \frac{1}{(i+1)^{\alpha}} & \text{if } \alpha \geq 2. \end{cases}$$

$$\leq \begin{cases} \frac{n-i}{(i+1)\ln^2 n} \sum_{x=1}^{n} \frac{1}{x} \approx \frac{n-i}{(i+1)\ln n} & \text{if } \alpha = 1; \\ \frac{n-i}{(i+1)^{\alpha} c(\alpha)^2} \sum_{x=1}^{n} \frac{1}{x^{\alpha}} \approx \frac{n-i}{(i+1)^{\alpha} c(\alpha)} & \text{if } \alpha \geq 2. \end{cases}$$

The edge capacity $Q(e) = \pi(i+1) P_{i+1,i}$

$$= \frac{1}{4}\pi(i+1) \approx \begin{cases} \frac{1}{4} \cdot \frac{1}{(i+1)\ln n} & \text{if } \alpha = 1; \\ \frac{1}{4} \cdot \frac{1}{(i+1)^{\alpha} c(\alpha)} & \text{if } \alpha \geq 2. \end{cases}$$

Hence,

$\rho(\Gamma) = \max_e \frac{1}{Q(e)} \sum_{\gamma_{xy} \ni e} \pi(x)\pi(y)$

$$\leq \begin{cases} \max_i 4(i+1)\ln n \cdot \frac{n-i}{(i+1)\ln n} & \text{if } \alpha = 1; \\ \max_i 4(i+1)^{\alpha} c(\alpha) \cdot \frac{n-i}{(i+1)^{\alpha} c(\alpha)} & \text{if } \alpha \geq 2. \end{cases}$$

= $4(n-1)$ for both cases.

Since each path in Γ has length at most n, by the definition of $\bar{\rho}$ we have $\bar{\rho} \leq \rho(\Gamma) \cdot n \leq 4n^2$. We also have

$$\pi_{\min} = \begin{cases} \frac{1}{n \ln n} & \text{if } \alpha = 1; \\ \frac{1}{c(\alpha) n^{\alpha}} & \text{if } \alpha \geq 2. \end{cases}$$

According to Theorem 5, we have the mixing time $\tau(\epsilon) \leq \bar{\rho} \log((\pi_{\min}\epsilon)^{-1}) = O(n^2(\log n + \log \epsilon^{-1}))$.

By choosing the desired variation difference ϵ as a small constant or asymptotically smaller than $\pi_{\min}$, *e.g.*, $\Theta(\frac{1}{n^{\alpha+1}})$, we have the final mixing time result:

Theorem 7: [**Mixing in unidirectional rings**] The mixing time of our random walk for distance-based distributions in a unidirectional ring with n nodes is $O(n^2 \log n)$.

Our algorithm on a unidirectional 2-dimensional torus with n^2 nodes can be viewed as a random walk, starting from node $(0,0)$, on a grid with n^2 nodes labeled from $(0,0)$ through $(n-1, n-1)$, where node (i,j) has $(i-1,j)$, $(i,j-1)$ as its predecessors and $(i+1,j)$, and $(i,j+1)$ as its successors

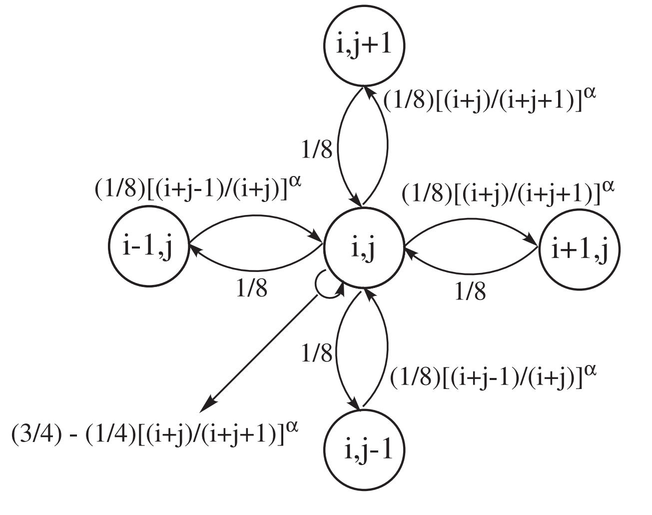

Fig. 2. The transition probabilities in the neighborhood of (i, j), $1 \le i, j \le n-2$. The hop distance from node $(0,0)$ to node (i,j) is $i+j$, which is decided by the random walk at runtime since i, j are the numbers of the horizontal forward steps and vertical forward steps taken by the random walk, respectively. The transition probabilities are decided at runtime without the global knowledge of n.

in the original torus. The hop distance from $(0,0)$ to (i,j) is $d((0,0),(i,j)) = i+j$. The transition probabilities between node (i,j) and its neighbors are illustrated in Fig. 2.

Let π be the converged distribution of the random walk, then for $x = (x_1, x_2)$ we know that $\pi(x)$ is proportionally to $d((0,0),(x_1,x_2))^{-\alpha} = (x_1+x_2)^{-\alpha}$. Hence for large n's,

$$\pi(x) = \frac{(x_1+x_2)^{-\alpha}}{\sum_{k,l \neq (0,0)} (k+l)^{-\alpha}} \approx \begin{cases} \frac{1}{(x_1+x_2) n \ln 4} & \text{if } \alpha = 1; \\ \frac{1}{(x_1+x_2)^2 \ln n} & \text{if } \alpha = 2; \\ \frac{1}{(x_1+x_2)^\alpha c(\alpha-1)} & \text{if } \alpha \ge 3. \end{cases}$$

Here $c(\alpha)$ is the power summation constant as described earlier.

We define Γ, a family of canonical paths γ_{xy}, in a way such that γ_{xy} is the path between x, y decided by the routing mechanism of CAN [21]. In other words, the next hop at each step would get closer to the target. To compute $\rho(\Gamma)$, we consider an arbitrary edge $e =< (i,j),(i+1,j) >$. We first need to consider $\sum_{\gamma_{xy} \ni e} \pi(x)\pi(y)$. We need to know the node pairs x, y that can be routed through edge e. These include all $x \in \{(0,0),(0,1),...,(i,j)\}$ and $y \in \{(i+1,j),(i+1,j+1),...,(n,n)\}$.

Let $x = (x_1,x_2)$, $y = (y_1,y_2)$. Then:
$\sum_{\gamma_{xy} \ni e} \pi(x)\pi(y)$

$$= \begin{cases} \frac{1}{n^2 \ln^2 4} \sum_{x=(1,0)}^{(i,j)} \sum_{y=(i+1,j)}^{(n,n)} \frac{1}{(x_1+x_2)(y_1+y_2)}; \\ \frac{1}{\ln^2 n} \sum_{x=(1,0)}^{(i,j)} \sum_{y=(i+1,j)}^{(n,n)} \frac{1}{(x_1+x_2)^2(y_1+y_2)^2}; \\ \frac{1}{c(\alpha-1)^2} \sum_{x=(1,0)}^{(i,j)} \sum_{y=(i+1,j)}^{(n,n)} \frac{1}{(x_1+x_2)^\alpha(y_1+y_2)^\alpha}. \end{cases}$$

$$= \begin{cases} \frac{1}{n^2 \ln^2 4} \sum_{x=(1,0)}^{(i,j)} \frac{1}{(x_1+x_2)} \sum_{y=(i+1,j)}^{(n,n)} \frac{1}{(y_1+y_2)}; \\ \frac{1}{\ln^2 n} \sum_{x=(1,0)}^{(i,j)} \frac{1}{(x_1+x_2)^2} \sum_{y=(i+1,j)}^{(n,n)} \frac{1}{(y_1+y_2)^2}; \\ \frac{1}{c(\alpha-1)^2} \sum_{x=(1,0)}^{(i,j)} \frac{1}{(x_1+x_2)^\alpha} \sum_{y=(i+1,j)}^{(n,n)} \frac{1}{(y_1+y_2)^\alpha}. \end{cases}$$

$$\le \begin{cases} \frac{1}{n^2 \ln^2 4} \sum_{x=(1,0)}^{(i,j)} \frac{1}{(x_1+x_2)} (n-i)(n-j) \frac{1}{i+j}; \\ \frac{1}{\ln^2 n} \sum_{x=(1,0)}^{(i,j)} \frac{1}{(x_1+x_2)^2} (n-i)(n-j) \frac{1}{(i+j)^2}; \\ \frac{1}{c(\alpha-1)^2} \sum_{x=(1,0)}^{(i,j)} \frac{1}{(x_1+x_2)^\alpha} (n-i)(n-j) \frac{1}{(i+j)^\alpha}. \end{cases}$$

$$\le \begin{cases} \frac{(n-i)(n-j)}{(i+j) n^2 \ln^2 4} \sum_{x=(1,0)}^{(n,n)} \frac{1}{(x_1+x_2)} & \text{if } \alpha = 1; \\ \frac{(n-i)(n-j)}{(i+j)^2 \ln^2 n} \sum_{x=(1,0)}^{(n,n)} \frac{1}{(x_1+x_2)^2} & \text{if } \alpha = 2; \\ \frac{(n-i)(n-j)}{(i+j)^\alpha c(\alpha-1)^2} \sum_{x=(1,0)}^{(n,n)} \frac{1}{(x_1+x_2)^\alpha} & \text{if } \alpha \ge 3. \end{cases}$$

$$\approx \begin{cases} \frac{(n-i)(n-j)}{(i+j) n \ln 4} & \text{if } \alpha = 1; \\ \frac{(n-i)(n-j)}{(i+j)^2 \ln n} & \text{if } \alpha = 2; \\ \frac{(n-i)(n-j)}{(i+j)^\alpha c(\alpha-1)} & \text{if } \alpha \ge 3. \end{cases}$$

The edge capacity
$Q(e) = P_{(i+1,j),(i,j)} \pi((i+1,j))$

$$\approx \begin{cases} \frac{1}{8} \cdot \frac{1}{(i+j) n \ln 4} & \text{if } \alpha = 1; \\ \frac{1}{8} \cdot \frac{1}{(i+j)^2 \ln n} & \text{if } \alpha = 2; \\ \frac{1}{8} \cdot \frac{1}{(i+j)^\alpha c(\alpha-1)} & \text{if } \alpha \ge 3. \end{cases}$$

Hence,
$\rho(\Gamma) = \max_e \frac{1}{Q(e)} \sum_{\gamma_{xy} \ni e} \pi(x)\pi(y)$

$$\le \begin{cases} \max_{i,j} 8 \cdot (i+j) n \ln 4 \cdot \frac{(n-i)(n-j)}{(i+j) n \ln 4} & \text{if } \alpha = 1; \\ \max_{i,j} 8 \cdot (i+j)^2 \ln n \cdot \frac{(n-i)(n-j)}{(i+j)^2 \ln n} & \text{if } \alpha = 2; \\ \max_{i,j} 8 \cdot (i+j)^\alpha c(\alpha-1) \cdot \frac{(n-i)(n-j)}{(i+j)^\alpha c(\alpha-1)} & \text{if } \alpha \ge 3. \end{cases}$$

= $8n^2$ for all three cases.

Since each path in Γ has length at most $2n$, by the definition of $\bar{\rho}$ we have $\bar{\rho} \le \rho(\Gamma) \cdot 2n \le 16n^3$. We also have

$$\pi_{\min} \approx \begin{cases} \frac{1}{2n^2 \ln 4} & \text{if } \alpha = 1; \\ \frac{1}{4n^2 \ln n} & \text{if } \alpha = 2; \\ \frac{1}{c(\alpha-1)(2n)^\alpha} & \text{if } \alpha \ge 3. \end{cases}$$

According to Theorem 5, we have the mixing time $\tau(\epsilon) \le \bar{\rho} \log((\pi_{\min}\epsilon)^{-1}) = O(n^3(\log n + \log \epsilon^{-1}))$.

Since the above bound is for a unidirectional 2-dimensional torus with size n^2. We have $\tau(\epsilon) = O(n^{1.5}(\log n + \log \epsilon^{-1}))$ for a unidirectional 2-dimensional torus with size n. By choosing the desired variation difference ϵ as a small constant or asymptotically smaller than $\pi_{\min}$, *e.g.*, $\Theta(\frac{1}{n^{\alpha+2}})$, we have the final mixing time result:

Theorem 8: **[Mixing in unidirectional 2-dimensional tori]** The mixing time of our random walks for distance-based distributions in a unidirectional 2-dimensional torus with n nodes is $O(n^{1.5} \log n)$.

We can see that our random walks are mixing faster in tori than in rings. This is because tori have better connectivity than rings.

By extending the above analytical process to unidirectional d-dimensional tori, we achieve the following analytical results:

Theorem 9: [**Mixing in unidirectional d-dimensional tori**] The mixing time of our random walks for distance-based distributions in a d-dimensional torus with n nodes is $O(n^{1+\frac{1}{d}}\log n)$.

Note that the above results for unidirectional tori can be unified with the results for unidirectional rings since unidirectional rings can be viewed as unidirectional one-dimensional tori.

C. The Mixing Time in Unstructured P2P Topologies

The underlying topologies of unstructured peer-to-peer networks (*e.g.*, Gnutella [22] and Freenet [23]) are usually characterized by *random regular graphs* [24] or *small-world power-law graphs* [25], which can help to maintain unstructured p2p topologies with desirable graph properties such as low diameters and good expansions.

In unstructured p2p topologies, hop distance is a much less accurate distance measure compared with the Internet distance, or the actual round trip transmission time. Thus we choose to use Internet distance-based target probability distribution in this study on unstructured p2p topologies. Unfortunately, it is too costly to measure the accurate Internet distance on-demand. Ng and Zhang proposed the global network positioning technique [26] to predict the Internet distance with moderate cost. Global network positioning maps network nodes to points in a Euclidean space, where the Internet distance between node i, j is approximated by $d(i,j)$, the Euclidean distance between the points corresponding to node i, j.

Our analysis is based on the above-mentioned Euclidean space model for unstructured peer-to-peer topologies. We present the analytical results on the mixing time of our distance-based random walks (initiated from an arbitrary node i) on *random regular graphs* and *power-law graphs* in Euclidean space. In our analysis, we define L as the longest Euclidean distance between nodes in the studied graph and l as the shortest Euclidean distance between nodes in the graph.

Let us consider a random d-regular graph ($d \geq 3$) with n nodes distributed in a multidimensional Euclidean space, where d is the degree of nodes. It is known that the graph diameter is $O(\log n)$ with high probability [27] and there exists a family of canonical paths, Γ, such that the number of paths containing an arbitrary edge $e =< j, k >$ is $O(n \log n)$ [28]. Hence

$$\sum_{\gamma_{xy} \ni e} \pi(x)\pi(y) \leq \frac{1}{l^\alpha} \cdot \frac{1}{C} \cdot \frac{1}{l^\alpha} \cdot \frac{1}{C} \cdot O(n \log n)$$

where the constant $C = \sum_{x \neq i} d(i,x)^{-\alpha}$ is the normalization factor since for each node $x \neq i$, $\pi(x)$ is proportionally to $d(i,x)^{-\alpha}$.

Without loss of generality, we assume that $d(i,j) \geq d(i,k)$. Then the edge capacity

$$Q(e) = P_{j,k}\pi(j) = \frac{1}{2d} \cdot \frac{1}{d(i,j)^\alpha} \cdot \frac{1}{C}$$

Then

$$\rho(\Gamma) = \max_e \frac{1}{Q(e)} \sum_{\gamma_{xy} \ni e} \pi(x)\pi(y)$$

$$\leq \max_j 2d \cdot d(i,j)^\alpha \cdot C \cdot \frac{1}{l^{2\alpha}} \cdot \frac{1}{C^2} \cdot O(n \log n)$$

$$= 2d \cdot L^\alpha \cdot \frac{1}{l^{2\alpha}} \cdot \frac{1}{C} \cdot O(n \log n)$$

$$\leq 2d \cdot L^\alpha \cdot \frac{1}{l^{2\alpha}} \cdot \frac{1}{n \cdot L^{-\alpha}} \cdot O(n \log n) = 2d \cdot (\frac{L}{l})^{2\alpha} \cdot O(\log n)$$

Each canonical path has length at most $O(\log n)$, the graph diameter. By the definition of $\bar{\rho}$ we have

$$\bar{\rho} = 2d \cdot (\frac{L}{l})^{2\alpha} \cdot O(\log n) \cdot O(\log n) = O(\log^2 n),$$

where we consider L, l, d as constant parameters independent of the network size. We know that $\pi_{\min} \geq \frac{1}{L^\alpha C} \geq \frac{1}{n} \cdot (\frac{l}{L})^\alpha$. Thus according to Theorem 5, we have the mixing time

$$\tau(\epsilon) \leq \bar{\rho} \log((\pi_{\min}\epsilon)^{-1}) = O(\log^2 n(\log n + \log \epsilon^{-1})).$$

By choosing the desired variation difference ϵ as a small constant or asymptotically smaller than $\pi_{\min}$, *e.g.*, $O(\frac{1}{n^2})$, we have the final mixing time result:

Theorem 10: [**Mixing in random d-regular graphs**] For $d \geq 3$, the mixing time of our random walks for distance-based distributions in a random d-regular graph with n nodes is $O(\log^3 n)$ with high probability.

Compared with rings and tori, random regular graphs have lower mixing time bounds due to their better expansion properties.

Unstructured p2p topologies are observed to possess *small-world* properties and *power-law* degree distributions [25]. For power-law graphs with the degree distribution $P(k) \propto k^{-\beta}$, the maximum node degree is $O(n^{\frac{1}{\beta}})$ with high probability for large n's. It is also known that the graph diameter is $O(\log n)$ [29] and there exists a family of canonical paths, Γ, such that the number of paths containing an arbitrary edge $e =< j, k >$ is $O(n \log^2 n)$ [28]. Based on these results, we can derive the convergence time bounds for power-law graphs as follows.

The edge congestion for an arbitrary edge e is

$$\sum_{\gamma_{xy} \ni e} \pi(x)\pi(y) \leq \frac{1}{l^\alpha} \cdot \frac{1}{C} \cdot \frac{1}{l^\alpha} \cdot \frac{1}{C} \cdot O(n \log^2 n)$$

Assuming $d(i,j) \geq d(i,k)$, the edge capacity

$$Q(e) = P_{j,k}\pi(j) = \frac{1}{2 \cdot Deg(j)} \cdot \frac{1}{d(i,j)^\alpha} \cdot \frac{1}{C},$$

where $Deg(j)$ is the degree of node j and the constant $C = \sum_{x \neq i} d(i,x)^{-\alpha}$ is the normalization factor. Hence,

$$\rho(\Gamma) = \max_e \frac{1}{Q(e)} \sum_{\gamma_{xy} \ni e} \pi(x)\pi(y)$$

$$\leq \max_j 2 \cdot Deg(j) \cdot d(i,j)^\alpha \cdot C \cdot \frac{1}{l^{2\alpha}} \cdot \frac{1}{C^2} \cdot O(n \log^2 n)$$

Network topologies	Asymptotic bounds
Rings	$O(n^2 \log n)$
d-dimensional tori	$O(n^{1+\frac{1}{d}} \log n)$
Random regular graphs	$O(\log^3 n)$ w.h.p.
Power-law graphs	$O(n^{\frac{1}{\beta}} \log^4 n)$ w.h.p.

TABLE I

ASYMPTOTIC BOUNDS ON THE CONVERGENCE TIME.

$$= 2 \cdot n^{\frac{1}{\beta}} \cdot L^{\alpha} \cdot \frac{1}{l^{2\alpha}} \cdot \frac{1}{C} \cdot O(n \log^2 n)$$

$$\leq 2 \cdot n^{\frac{1}{\beta}} \cdot L^{\alpha} \cdot \frac{1}{l^{2\alpha}} \cdot \frac{1}{n \cdot L^{-\alpha}} \cdot O(n \log^2 n) = 2 \cdot n^{\frac{1}{\beta}} \cdot (\frac{L}{l})^{2\alpha} \cdot O(\log^2 n)$$

Each canonical path has length at most $O(\log n)$, the graph diameter. By the definition of $\bar{\rho}$ we have

$$\bar{\rho} = 2 \cdot n^{\frac{1}{\beta}} \cdot (\frac{L}{l})^{2\alpha} \cdot O(\log^2 n) \cdot O(\log n) = O(n^{\frac{1}{\beta}} \cdot \log^3 n),$$

where we consider L, l as constant parameters independent of the network size. We know that $\pi_{\min} \geq \frac{1}{L^{\alpha} C} \geq \frac{1}{n} \cdot (\frac{l}{L})^{\alpha}$. Thus according to Theorem 5, we have the mixing time

$$\tau(\epsilon) \leq \bar{\rho} \log((\pi_{\min}\epsilon)^{-1}) = O(n^{\frac{1}{\beta}} \cdot \log^3 n(\log n + \log \epsilon^{-1})).$$

By choosing the desired variation difference ϵ as a small constant or asymptotically smaller than $\pi_{\min}$, *e.g.*, $O(\frac{1}{n^2})$, we have the final mixing time result:

Theorem 11: [**Mixing in power-law graphs**] With high probability, the mixing time of our random walks for distance-based distributions in a power-law graph the degree distribution $P(k) \propto k^{-\beta}$ is $O(n^{\frac{1}{\beta}} \cdot \log^4 n)$, where n is the network size.

Table I summarizes the asymptotic bounds on the mixing (or convergence) time of our random walks for the four topologies we studied.

IV. SIMULATION RESULTS

The mixing time bound results in Section III are achieved by using the canonical path approach. Though the canonical path approach is a popular technique for bounding the mixing time, for many applications it only provides weak bounds. Considering this, our mixing time bounds derived by this approach may not be tight. It remains as our future work to explore new techniques to give improved mixing time bounds for our random walks. As a complement to the asymptotic mixing time bounds, we present the simulation results of our random walks in terms of the convergence time for major p2p topologies.

In this section, we study the mixing time of our distance-based random walks on four kinds of networks with size ranging from 2^6 to 2^{13} nodes: unidirectional rings, unidirectional 2-dimensional tori, random regular graphs, and Barabási-Albert power-law graphs [30][1]. For each kind of networks, we not only study the mixing time of our random walks in static networks but also consider dynamic networks with uniformly random node arrivals and departures. Here the mixing time is measured by the number of steps for a random walk (initiated from an arbitrary node i) to satisfy the *total variation difference* $||\pi_t, \pi|| \leq 0.001$ (Definition 1, Section II-B) or satisfy the *maximum relative error* $M(\pi_t, \pi) \leq 1\%$. Here π is the ideal distance-based distribution as defined in Section III-A ($\forall j \neq i, \pi(j) \propto d(i,j)^{-\alpha}$). π_t is the distribution achieved by our distance-based random walks at step t. $M(\pi_t, \pi) = \max_i \frac{|\pi_t(i) - \pi(i)|}{\pi(i)}$.

A. Unidirectional Rings

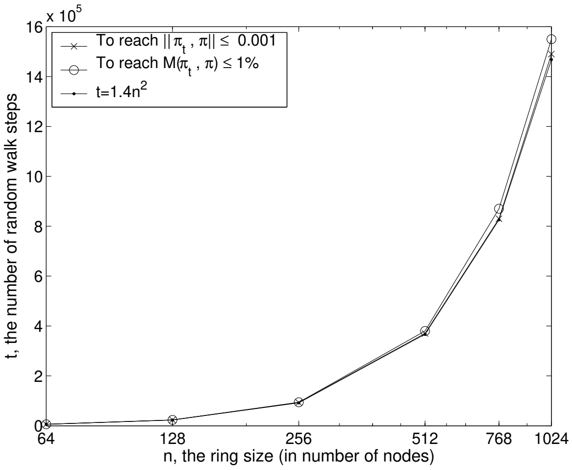

Fig. 3. **Convergence for rings**. The number of steps for a random walk initiated from an arbitrary node to reach small *total variation difference* or small *maximum relative error*, *i.e.*, $||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$, in static unidirectional rings.

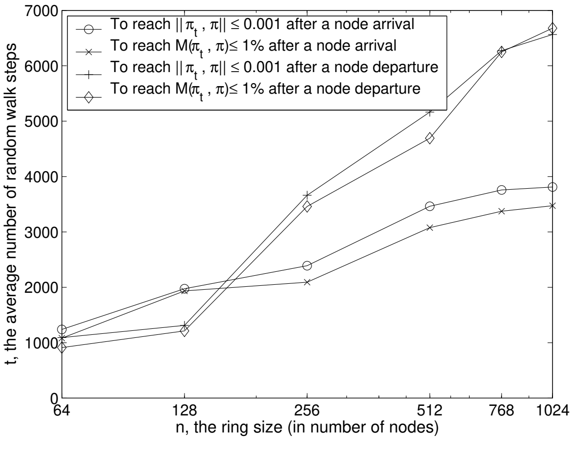

Fig. 4. **Convergence for rings after network changes**. The average number of steps for previously converged random walks (with $||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$) to converge again ($||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$) after network changes in unidirectional rings. A network change is either a uniformly random node arrival or the departure of an existing node chosen uniformly at random.

Fig. 3 illustrates the number of steps for our random walks ($\alpha = 1$ for rings) to reach $||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$

[1]The node degree distribution of Barabási-Albert power-law graphs can be approximated by $P(k) \propto k^{-3}$ [31].

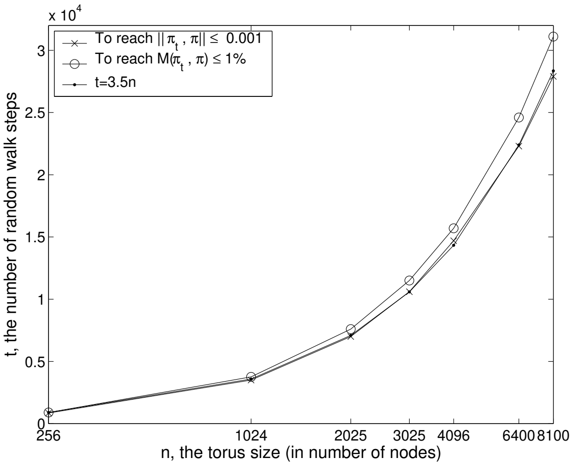

Fig. 5. **Convergence for tori**. The number of steps for a random walk initiated from an arbitrary node to reach $||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$ in static unidirectional 2-dimensional tori.

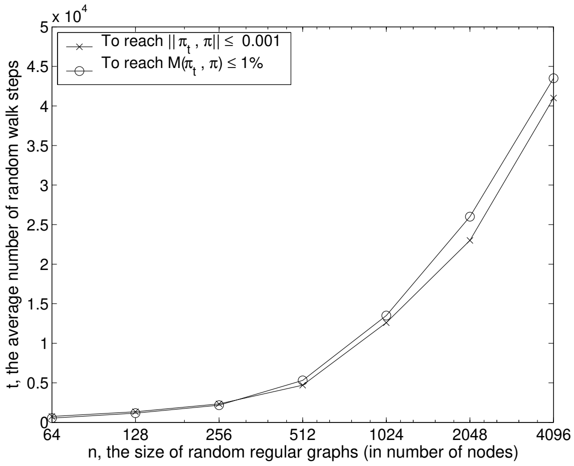

Fig. 7. **Convergence for random regular graphs**. The number of steps for a random walk initiated from an arbitrary node to reach $||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$ in static random regular graphs.

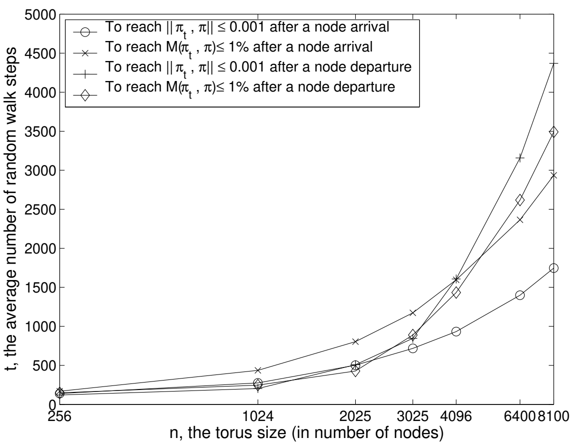

Fig. 6. **Convergence for tori after network changes**. The average number of steps for previously converged random walks (with $||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$) to converge again ($||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$) after network changes in unidirectional 2-dimensional tori. A network change is either a uniformly random node arrival or the departure of an existing node chosen uniformly at random.

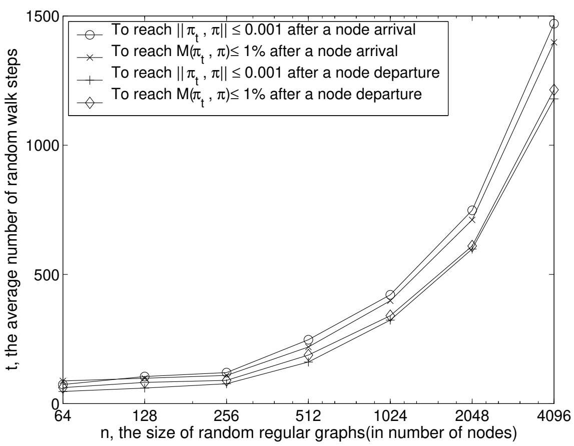

Fig. 8. **Convergence for random regular graphs after network changes**. The average number of steps for previously converged random walks (with $||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$) to converge again ($||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$) after network changes in random regular graphs. A network change is either a uniformly random node arrival or the departure of an existing node chosen uniformly at random.

in unidirectional rings of different sizes. The relationship between the number of steps, t, and the ring size n is very close to $t = 1.4n^2$, which suggests a tighter mixing time bound, $O(n^2)$, than the bound in Theorem 7, $O(n^2 \log n)$. Fig. 3 shows that it takes a large number of steps for a random walk initiated from an arbitrary node to converge in a static network. However, once a random walk converges, it will converge relatively faster to future dynamic network changes as shown in Fig. 4.

Specifically, if we assign a 40 ms average latency to all links, then our random walks in a 1024-node unidirectional ring take about 17 hours for initial convergence and thereafter take averagely about 4 minutes to converge again after a uniformly random node arrival or departure. The slow convergence is due to the low connectivity of rings, which could be compensated by introducing more per-node links or long links as in Chord [15]. Due to their better expansion properties, other p2p topologies (results shown in later subsections) exhibit much faster convergence speed than rings.

B. Unidirectional 2-dimensional Tori

Fig. 5 illustrates the number of steps for our random walks ($\alpha = 2$ for 2-dimensional tori) to reach $||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$ in unidirectional 2-dimensional tori of different sizes. The relationship between the number of steps, t, and the torus size n is close to $t = 3.5n$, which suggests a tighter mixing time bound, $O(n)$, than the bound in Theorem 8, $O(n^{1.5} \log n)$. Fig. 6 gives the average number of steps for previously converged random walks to converge again after network changes.

Based on the 40 ms link latency estimate, our random walks in a 1024-node unidirectional 2-dimensional torus take

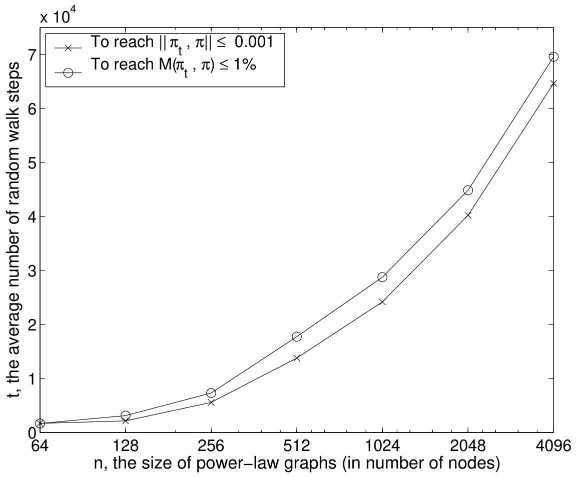

Fig. 9. **Convergence for power law graphs**. The number of steps for a random walk initiated from an arbitrary node to reach $||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$ in static Barabási-Albert power-law graphs.

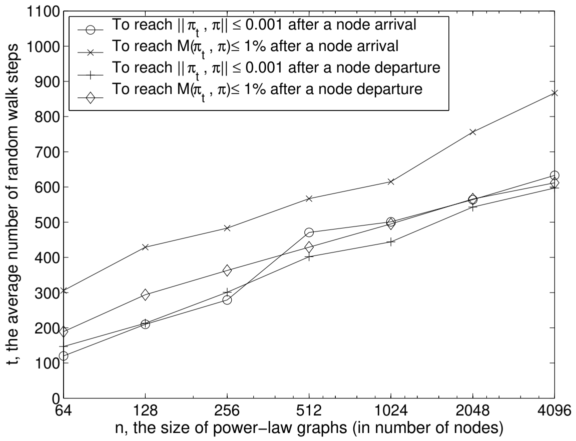

Fig. 10. **Convergence for power law graphs after network changes**. The average number of steps for previously converged random walks (with $||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$) to converge again ($||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$) after network changes in Barabási-Albert power-law graphs. A network change is either a uniformly random node arrival or the departure of an existing node chosen uniformly at random.

about 2.5 minutes for initial convergence and thereafter take averagely about 17 seconds to converge again after a uniformly random node arrival or departure.

C. Random regular graphs

Here we consider random regular graphs with nodes generated uniformly at random in a unit cube, $(0, 1)^3$. The node degree is 6. Fig. 7 illustrates the number of steps for our random walks ($\alpha = 3$) to reach $||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$ in random regular graphs of different sizes. Fig. 8 gives the average number of steps for previously converged random walks to converge again after network changes.

Based on the 40 ms link latency estimate, our random walks in a 1024-node random 6-regular graph take about 9 minutes for initial convergence and thereafter take averagely about 16 seconds to converge again after a uniformly random node arrival or departure.

Network topologies	Initial convergence	Convergence after a network change
Rings	17 hours	4 minutes
2-dimensional tori	2.5 minutes	17 seconds
Random regular graphs	9 minutes	16 seconds
Power-law graphs	19 minutes	25 seconds

TABLE II

CONVERGENCE TIME FOR 1024-NODE NETWORKS WITH 40 MS LATENCY FOR ALL LINKS.

D. Barabási-Albert Power-law Graphs

We consider power-law graphs generated based on the Barabási-Albert model [30]. The nodes are generated uniformly at random in a unit cube, $(0, 1)^3$. During the growth of the graph, a node joins the graph by linking to 6 existing nodes chosen randomly with probability proportional to their degrees. Fig. 9 illustrates the number of steps for our random walks ($\alpha = 3$) to reach $||\pi_t, \pi|| \leq 0.001$ or $M(\pi_t, \pi) \leq 1\%$ in Barabási-Albert graphs of different sizes. Fig. 10 gives the average number of steps for previously converged random walks to converge again after network changes.

Based on the 40 ms link latency estimate, our random walks in a 1024-node Barabási-Albert graph take about 19 minutes for initial convergence and thereafter take averagely about 25 seconds to converge again after a uniformly random node arrival or departure.

Table II summarizes the simulation results on the convergence time of our random walks. The results are for 1024-node networks with 40 ms latency for all links. Note that the results for rings and tori are not directly comparable to those of random regular graphs and power-law graphs because they use different distance metrics.

V. CONCLUSIONS

In this paper, we present a non-uniform random membership management algorithm satisfying distance-based distributions for peer-to-peer networks. To the best of our knowledge, our algorithm is the first to support non-uniform random membership management with proved convergence and analytical bounds on the convergence time. Along with asymptotic bounds, we also provide simulation results to quantitatively assess the algorithm convergence time at typical settings.

Our algorithm does not put restrictions on network topologies and can be applied to many p2p topologies, such as rings, tori, random regular graphs, and power law graphs. The framework of our algorithm can also be used to generate random membership subsets with other non-uniform distributions.

It remains to explore new techniques to achieve tighter mixing time bounds for our distance-based random walks. We will also extend our algorithm to other peer-to-peer topologies, such as de Bruijn graphs, butterflies, and skip nets.

ACKNOWLEDGMENT

We would like to thank the anonymous referees for their valuable comments.

REFERENCES

[1] R. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking, "Randomized Rumor Spreading," in *Proc. of the 41st IEEE Sympo. on Foundations of Computer Science*, Redondo Beach, CA, Nov. 2000, pp. 565–574.

[2] B. Pittel, "On Spreading A Rumor," *SIAM J. Applied Math.*, vol. 47, no. 1, pp. 213–223, Feb. 1987.

[3] G. S. Manku, "Routing Networks for Distributed Hash Tables," in *Proc. of the 22nd ACM Sympo. on Principles of Distributed Computing*, June 2003, pp. 133–142.

[4] G. S. Manku, M. Bawa, and P. Raghavan, "Symphony: Distributed Hashing in A Small World," in *Proc. of the 4th USENIX Sympo. on Internet Technologies and Systems*, Seattle, WA, Mar. 2003.

[5] D. Karger and M. Ruhl, "Simple Efficient Load Balancing Algorithms for Peer-to-Peer Systems," in *Proc. of ACM SPAA*, Barcelona, Spain, June 2004, pp. 36–43.

[6] D. Kostić, A. Rodriguez, J. Albrecht, A. Bhirud, and A. Vahdat, "Using Random Subsets to Build Scalable Network Services," in *Proc. of the 4th USENIX Sympo. on Internet Technologies and Systems*, Seattle, WA, Mar. 2003.

[7] K. Shen, "Structure Management for Scalable Overlay Service Construction," in *Proc. of the First USENIX/ACM Sympo. on Networked Systems Design and Implementation*, San Francisco, CA, Mar. 2004, pp. 281–294.

[8] P. Th. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec, "Lightweight Probabilistic Broadcast," *ACM Trans. on Computer Systems*, vol. 21, no. 4, pp. 341–374, Nov. 2003.

[9] A. J. Ganesh, A. Kermarrec, and L. Massoulié, "SCAMP: Peer-to-peer Lightweight Membership Service for Large-scale Group Communication," in *Proc. of the 3rd International Workshop on Networked Group Communicatio*, London, UK, Nov. 2001, pp. 44–55.

[10] D. Kempe, J. Kleinberg, and Alan Demers, "Spatial Gossip and Resource Location Protocols," in *Proc. of the 33rd ACM symposium on Theory of computing*, Hersonissos, Greece, 2001, pp. 163–172.

[11] J. Kleinberg, "The Small-world Phenomenon: An Algorithm Perspective," in *Proc. of the 32nd ACM symposium on Theory of computing*, Portland, OR, 2000, pp. 163–170.

[12] S. Tilak, A. Murphy, and W. Heinzelman, "Non-Uniform Information Dissemination for Sensor Networks," in *Proc. of the 11th IEEE International Conference on Network Protocols*, Atlanta, GA, Nov. 2003, pp. 295–304.

[13] 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.*, vol. 21, pp. 1087–1092, 1953.

[14] Y. Azar, A. Broder, A. Karlin, N. Linial, and S. Phillips, "Biased Random Walks," in *Proc. of the 24th ACM Sympo. on the Theory of Computing*, 1992, pp. 1–9.

[15] 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 the ACM SIGCOMM*, San Diego, CA, Aug. 2001, pp. 149–160.

[16] V. King and J. Saia, "Choosing A Random Peer," in *Proc. of the 23rd ACM Sympo. on Principles of Distributed Computing*, 2004, pp. 125–130.

[17] W. Doeblin, "Exposé de la théorie des chaînes simples constantes de Markov á un nombre fini d'états," *Mathématique de l'Union Interbalkanique*, vol. 2, pp. 77–105, 1938.

[18] P. Diaconis and D. Stroock, "Geometric Bounds for Eigenvalues of Markov Chains," *Annals of Applied Probability*, vol. 1, pp. 36–61, 1991.

[19] A. Sinclair, "Improved Bounds for Mixing Rates of Markov Chains and Multicommodity Flow," *Combinatorics, Probability and Computing*, vol. 1, pp. 351–370, 1992.

[20] A. Sinclair and M. Jerrum, "Approximate Counting, Uniform Generation and Rapidly Mixing Markov Chains," *Information and Comput.*, vol. 82, pp. 93–133, 1989.

[21] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," in *Proc. of the ACM SIGCOMM*, San Diego, CA, Aug. 2001, pp. 161–172.

[22] "Gnutella," http://www.gnutella.com.

[23] I. Clarke, T. Hong, S. Miller, O. Sandberg, and B. Wiley, "Protecting Free Expression Online with Freenet," *IEEE Internet Computing*, vol. 6, no. 1, pp. 40–49, 2002.

[24] C. Law and K. Siu, "Distributed Construction of Random Expander Networks," in *Proc. of the IEEE INFOCOM*, San Francisco, CA, Mar. 2003.

[25] M. Jovanović, F. Annexstein, and K. Berman, "Modeling Peer-to-peer Network Topologies Through Small-world Models and Power Laws," in *IX Telecommunications Forum*, 2001.

[26] T. S. Eugene Ng and H. Zhang, "Predicting Internet Network Distance with Coordinates-based Approaches," in *Proc. of the IEEE INFOCOM*, New York, NY, June 2002.

[27] B. Bollobás, *Random Graphs*, Academic Press, London, UK, 1985.

[28] C. Gkantsidis, M. Mihail, and A. Saberi, "Conductance and Congestion in Power Law Graphs," in *Proc. of ACM SIGMETRICS*, San Diego, CA, June 2003, pp. 148–159.

[29] B. Bollobás and O. Riordan, "The Diameter of a Scale-free Random Graph," *Combinatorica*, vol. 24, no. 1, pp. 5–34, 2004.

[30] A. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, pp. 509–512, 1999.

[31] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnady, "The Degree Sequence of a Scale-free Random Graph Process," *Random Structures and Algorithms*, vol. 18, no. 3, pp. 279–290, 2001.