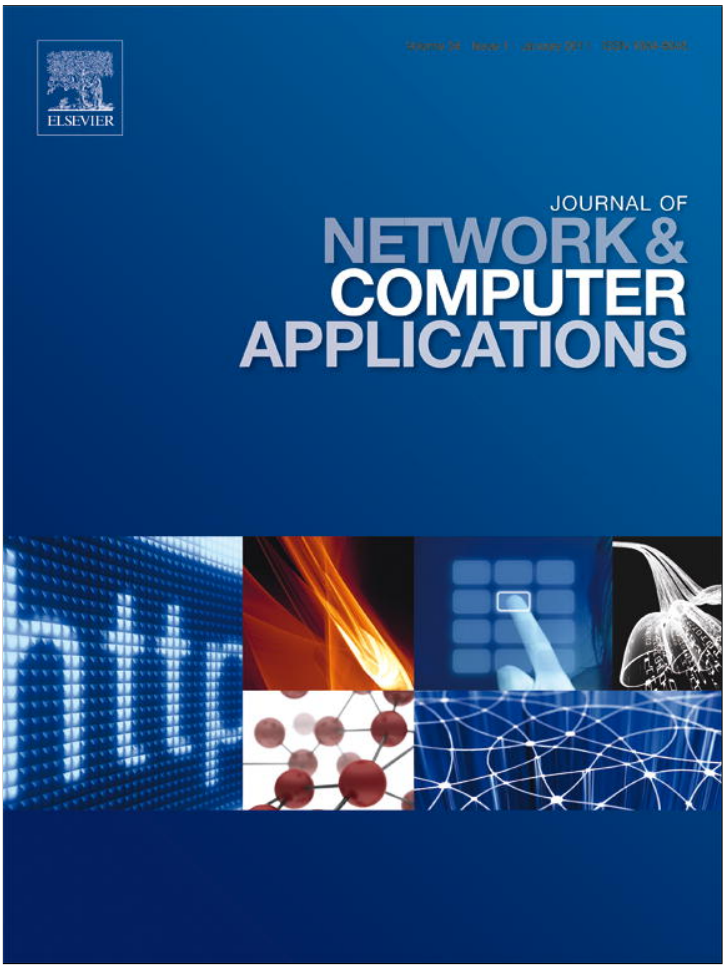

Journal of Network and Computer Applications 34 (2011) 252–261

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

# Randomized load balancing strategies with churn resilience in peer-to-peer networks

Song Fu [a,*], Cheng-Zhong Xu [b], Haiying Shen [c]

[a] Department of Computer Science and Engineering, University of North Texas, Denton, TX 76207, USA
[b] Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA
[c] Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, USA

ARTICLE INFO

*Article history:*
Received 27 August 2009
Received in revised form
27 April 2010
Accepted 14 July 2010

*Keywords:*
Peer-to-peer systems
Randomized probing
Load balancing
Heterogeneous and bounded node capacity
Churn

ABSTRACT

The objective of load balancing in peer-to-peer (P2P) networks is to balance the workload of peer nodes in proportion to their capacity so as to eliminate performance bottlenecks. It is challenging because of the dynamic nature in overlay networks, the time-varying load characteristics, and the inherent load imbalance caused by consistent hashing functions. It is known that simple randomized load balancing schemes can balance load effectively while incurring only a small overhead in general parallel and distributed computing contexts. Existing theoretical works which analyze properties of randomized load balancing schemes cannot be applied in the highly dynamic and heterogeneous P2P systems. In this paper, we characterize the behaviors of randomized load balancing schemes in a general P2P environment. We extend the supermarket model by investigating the impact of node heterogeneity and churn on load distribution in P2P networks. We prove that by using $d$-way random choices schemes, the length of the longest queue in a P2P system with heterogeneous nodes and churn for $d \geq 2$ is $c * \log\log n/\log d + O(1)$ with high probability, where $c$ is a constant. Our results have wide applicability and are of interest beyond the specific applications.



## 1. Introduction

Peer-to-peer (P2P) networks have become, in a short period of time, one of the fastest growing and most popular Internet applications. An important class of P2P overlay networks is the distributed hash tables (DHTs) that map keys to peer nodes based on a consistent hashing function. Representatives of DHTs include Chord (Stoica et al., 2003), Pastry (Rowstron and Druschel, 2001), Tapestry (Zhao et al., 2004), CAN (Ratnasamy et al., 2001), and Cycloid (Shen et al., 2006). In a DHT, each node and key has a unique ID, and a key is mapped to a node according to the DHT definition. The ID space of a DHT is partitioned among nodes and each of them is responsible for the keys whose IDs are located within its ID space. An important goal in designing DHTs is to achieve a balanced partition of the hash space among peer nodes. It is often desirable that each node takes responsibility for a portion of the hash space which is proportional to its processing capacity, measured in terms of its processor speed, available bandwidth, and/or storage capacity. This property should be sustained as nodes join and leave the system. A similar goal is desirable in unstructured P2P networks as well.

However, consistent hashing (Stoica et al., 2003) produces a bound of $O(\log n)$ imbalance of key distribution among peer nodes, where $n$ is the number of nodes in a P2P network. Things are even worse in unstructured P2P systems, where no commonly accepted load distribution mechanism is supported. In addition, users may query geographically adjacent nodes and those that have popular files. These lead to even more imbalanced distribution of workload in the network. When a node becomes overloaded, it cannot store any additional files or respond to user queries any more, which affects the system utilization and user satisfaction. To balance load among peer nodes in a network, lightly loaded peers should be selected to store files or serve queries. Load balancing in P2P networks is an important topic and many related works have been conducted recently (Tewari and Kleinrock, 2006; Shen and Xu, 2006; Zheng et al., 2006; Godfrey and Stoica, 2005; Bienkowski et al., 2005).

It is known that simple randomized load balancing schemes can balance load effectively while incurring only a small overhead in general parallel and distributed computing environments (Xu and Lau, 1997), which makes such schemes appealing for practical systems. The approach of multiple random choices in P2P networks was used in Shen and Xu (2006), Godfrey and Stoica (2005), Kenthapadi and Manku (2005), Zhu and Hu (2005), and Byers et al. (2003). Several peer nodes are probed and the one with the least load is selected to store a file on it or service a user query. Random choice algorithms are scalable and they require a small number of control messages and data structures (Shen and Xu, 2006; Byers et al., 2003). More importantly, they work well in P2P systems with churn, a situation where a large percentage of

* Corresponding author.
*E-mail addresses:* song.fu@unt.edu (S. Fu), czxu@wayne.edu (C.-Z. Xu), shenh@clemson.edu (H. Shen).

1084-8045/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2010.07.011

nodes join and leave continuously and rapidly, which leads to an unpredictable network size.

To theoretically analyze the effectiveness of random-choice schemes for balancing workload in distributed systems, researchers have proposed several approaches. Azar et al. (2000) introduced a *layered induction* method, by which the random choice problem was modeled by balls-and-bins. It provides nearly tight results. The *witness tree* approach was proposed by Cole et al. (1998) to tackle the random-choice problem. The probability of a certain event can then be bounded by the probability of occurrence of a witness tree. Generally speaking, the witness tree approach involves the most complexity, and it has been proved to be the most challenging one in obtaining tight results. The *fluid limit* model (Mitzenmacher, 2001; Mitzenmacher et al., 2002) characterizes system dynamics by constructing a family of differential equations, which makes this approach simple yet flexible. In cases where the system dynamics can be modeled by this method, the differential equations generate very accurate numerical results.

However, these theoretical works only analyzed systems with compute nodes having homogeneous and infinite capacities. Moreover, node churn, a defining characteristic of P2P networks, is not modeled by the existing approaches. As a result, we cannot conclude directly that the performance bounds derived in those works will still be valid in P2P networks. In this paper, we analyze the dynamic behavior of randomized load balancing algorithms in a general P2P network, where peer nodes join and leave the network at runtime and they have heterogeneous and bounded capacities.

We model a dynamic P2P system as follows. Load queries arrive as a Poisson stream to a collection of $n$ peer nodes, where $n$ is a random variable reflecting node churn. Nodes are heterogeneous with bounded capacity. For each query, a number of $d$ nodes are chosen independently and uniformly at random, and the query is queued at the node which currently accommodates the fewest number of queries. We refer to such multiple choice processing as $d$-way probing. Queries arrive to peer nodes at rate $\lambda$ which is relative to the node population. They are served according to the FIFO protocol, and the service time for a query is exponentially distributed with mean 1.

We extend the supermarket model (Mitzenmacher, 2001) to formulate behaviors of the preceding dynamic system in general cases and to derive system properties. We are interested in characterizing the average response delay and the maximum load on active nodes. However, quantifying these metrics in a general P2P system is nontrivial. It is difficult to find closed-form solutions to the differential equations describing the system dynamics, after we remove certain constraints, such as static system configuration, and homogeneous and infinite node capacities. Instead of solving the equations directly, we study the lower and upper bounds of state variables at equilibrium points with reference to those in special cases. Based on these bounds, we quantify the average response delay and the maximum load, and derive the following properties of $d$-way randomized probing in P2P networks.

**Theorem 1.** *For any fixed time interval $I$, the expected time that a query spends in a dynamic P 2P system with $d$-way randomized probing ($d \geq 2$), denoted by $T_d(\lambda)$, over interval $[0, I]$ satisfies that $T_d(\lambda)/\log T_1$ is close to $\alpha(1/\log d)$, for $\lambda$ close to 1, where $\alpha$ is a constant whose value depends on capacities of peer nodes and the change rate of the node population.*

**Theorem 2.** *For any fixed time interval $I$, the length of the longest queue in the dynamic P 2P models with $d$-way randomized probing ($d \geq 2$) over interval $[0, I]$ is $c \log\log n + O(1)$ with high probability $(1-O(1/n))$, where $c$ is a constant depending on capacities of peer nodes, $d$ and the arrival rate of queries, and the form of the $O(1)$ term depends on $I$ and some constants.*

The theorems show that two-way randomized probing is asymptotically superior to one-way probing. However, by increasing the number of choices further, efficiency of the load balancing algorithms does not improve significantly. These results are consistent with the findings by the supermarket model (Mitzenmacher, 2001) in special case where the number of servers is fixed, and their capacities are homogeneous and infinite. We also conduct experiments on a P2P network. Experimental results confirm the correctness of our findings. Although the randomized probing algorithms for load balancing are designed and analyzed within the context of P2P networks, the results have wide applicability and are of interest beyond the specific applications.

The remainder of this paper is organized as follows. The basic supermarket model is briefly described in Section 2. Section 3 formulates $d$-way randomized probing in P2P networks. We investigate the influences of node capacity in Section 3.1 and node churn in Section 3.2. By analyzing the equilibrium points of a P2P system, we quantify the expected time of a query in the system and the length of the longest queue among peer nodes. Experimental results are shown in Section 4. Section 5 presents the related work. Conclusions and remarks on future works are presented in Section 6.

## 2. Basic supermarket model

A load balancing scheme distributes user requests or storage loads among compute nodes and avoids hot spots. Mitzenmacher (2001) presents a supermarket model based on differential equations to analyze both static and dynamic load balancing strategies. In this section, we briefly describe this model, and in the subsequent sections we will present our extension of the model to formalize randomized probing algorithms for load balancing in general P2P systems.

Supermarket model analyzes properties of randomized load balancing strategies in a special distributed environment. In this environment, user requests arrive as a Poisson stream at a collection of servers. For each request, some constant number of servers are chosen independently and uniformly at random with replacement from the servers, and request waits for service at the server which currently accommodates the fewest requests. Requests are served according to the FIFO protocol, and the service time for a request is exponentially distributed with mean 1. Three underlying assumptions (Mitzenmacher, 2001) are: (a) unbounded server capacities; (b) static server configuration; and (c) homogeneous servers. The author derived the average time of a request staying in the system and the maximum workload of the servers by solving the differential equations of system states.

## 3. Randomized probing in general P2P systems

In large-scale P2P systems, a great number of peer nodes share resources and send queries to each other. More often than not, they have heterogeneous configurations of storage capacity and processing speed. In addition, dynamics is a defining characteristic of P2P networks, with nodes joining and leaving the network frequently. Load balancing in such large-scale and dynamic distributed environments is challenging. Obtaining the capacity information of all active nodes before dispatching jobs to the most

lightly loaded nodes is expensive. Randomized probing is a remedy to this problem.

By applying randomized probing algorithms, we make dispatch decisions based on the load status of a small number of nodes that are selected randomly. In this way, the number of load query messages that are exchanged is reduced significantly. The scalability of such algorithms is assured because the number of control messages for each decision making is almost constant even when the system scale expands. However, theoretically analyzing behaviors of randomized load balancing algorithms in general cases is challenging. Node heterogeneity and churn make the problem intractable. In this section, we extend the supermarket model to formulate behaviors of randomized probing algorithms in general P2P networks and to quantify system dynamics with regard to the node workload and average response time based on our model. Our extension is made in two orthogonal dimensions: server capacity (from the homogeneous and unbounded case to the heterogeneous and bounded case) and node dynamics (from static configuration to dynamic configuration).

### 3.1. Heterogeneous and bounded node capacity

In the basic supermarket model, as described in Section 2, servers are modeled to be homogeneous with unbounded capacity. However, in real-world P2P systems, peer nodes have limited and different storage capacity and processing speed. In this section, we analyze behaviors of random-choice algorithms for load balancing in P2P networks where nodes have heterogeneous and bounded capacity. We extend the basic supermarket model in two steps: bounding node capacity in a homogeneous environment (Section 3.1.1) and then allowing heterogeneous node capacity (Section 3.1.2). Here we assume a P2P network has static composition, where the number of peer nodes in the network is $N$, a constant. Node churn or dynamic composition will be studied in Section 3.2.

#### 3.1.1. Homogeneous and bounded case

In a homogeneous P2P system with bounded node capacity, we use $C$ to denote the uniform capacity of peer nodes. The value of $C$ is set to the maximum number of queries that a node can queue at runtime. When a node receives a query from a peer, it serves the query if there is extra capacity to handle it. Otherwise, it drops the query by its admission controller. Therefore, we adopt the saturation policy as follows. A query is turned down when all of the $d$ nodes, selected randomly and independently, are saturated, i.e. their load equals to their capacity.

Let $n_i(t)$ denote the number of nodes queuing $i$ queries at time $t$; $m_i(t)=\sum_{k=i}^{C} n_k(t)$, i.e. the number of nodes queuing at least $i$ queries at time $t$; $p_i(t)=n_i(t)/n$ be the fraction of nodes that have queues of size $i$; $s_i(t)=\sum_{k=i}^{C} p_k(t)=m_i(t)/n$ be the tails of the $p_i(t)$. We drop the reference to $t$ in the notation where the meaning is clear. The $s_i$ value is more convenient to work with than the $p_i$ value. In an empty system, which corresponds to one with no query, $s_0=1$ and $s_i=0$ for $1\le i\le C$. The expected number of queries per node at any time $t$ is $\sum_{i=1}^{C} s_i(t)$, and it is finite because each node can queue at most $C$ queries at a time.

The state of the system at any given time can be represented by a finite vector $\vec{s}=[s_0,s_1,\ldots,s_C]$. For each value of query size, it contains the corresponding number of nodes queuing that value. We can derive the maximum load of peer nodes and the average response time of queries, based on the dynamics of this state information. This resulting model can be considered as a Markov chain on the above state space.

We now extend the basic supermarket model to formulate and analyze randomized probing among peer nodes with homogeneous and bounded capacity. The time evolution of the P2P system is specified by the following set of differential equations:

$$\begin{cases} \dot{s}_i=\lambda(s_{i-1}^d-s_i^d)-(s_i-s_{i+1}) & \text{for } i<C,\\ \dot{s}_C=\lambda(s_{C-1}^d-s_C^d)-s_C,\\ s_0=1, \end{cases} \tag{3.1}$$

where $\dot{s}_i$ denotes $ds_i/dt$.

Let us explain why we have Eq. (3.1). In a P2P network with $N$ nodes, we will determine the expected change of the number of nodes with at least $i$ queries over a small period of time $dt$. The probability that a query arrives during this period is $\lambda N dt$, and the probability that an arriving query is dispatched to a node queuing $i-1$ queries is $s_{i-1}^d-s_i^d$, i.e. all of the $d$ nodes chosen by the new query have queues of size at least $i-1$, but not all of size at least $i$. The probability a query leaves a node with queue size $i$ in this period is $N(s_i-s_{i+1})\,dt$, for $i<C$. Because each node can serve no more than $C$ queries at a time, the probability a query leaves a node with queue size $C$ in this period is $Ns_C\,dt$.

Next, we try to find the equilibrium points of (3.1). At the equilibrium points, the volume of incoming queries equals the volume of outgoing queries, i.e. $\dot{s}_i=0$. For a special case $d=1$, system (3.1) becomes stable at states

$$\pi_i=\frac{\lambda^i-\lambda^{C+1}}{1-\lambda^{C+1}} \quad \text{for } 1\le i\le C.$$

We denote the expected time that a query spends in the P2P system with homogeneous and bounded-capacity nodes by $T_d(\lambda)$. As we mention before, the probability that an incoming query arriving at time $t$ becomes the $i$th query in the queue is $s_{i-1}(t)^d-s_i(t)^d$. Therefore, the expected time a query that arrives at time $t$ spends in the system is $T_d(\lambda)=\sum_{i=1}^{C} i(s_{i-1}(t)^d-s_i(t)^d)=\sum_{i=0}^{C-1} s_i(t)^d-Cs_C(t)^d$. For $d=1$, it is clear that at the equilibrium point,

$$T_1(\lambda)=\sum_{i=0}^{C-1} s_i-Cs_C=\frac{1-(C+1)\lambda^C+C\lambda^{C+1}}{(1-\lambda)(1-\lambda^{C+1})}.$$

Then, we consider the convergence of sequence $\{s_i(t)\}_{i=0}^{C}$ for $d\ge 1$.

**Definition 1.** A sequence $\{x_i\}$ is said to decrease doubly exponentially if and only if there exist positive constants $N$, $\alpha<1$, $\beta>1$, and $\gamma$ such that for $i\ge N$, $x_i\le\gamma\alpha^{\beta^i}$.

Then, we show that every trajectory of the system (3.1) converges to a fixed point.

**Corollary 2.** *Suppose there exists some $j$ such that $s_j(0)=0$. Then the sequence $\{s_i(t)\}_{i=0}^{C}$ decreases doubly exponentially for all $t\ge 0$, where the associated constants are independent of $t$.*

**Proof.** According to the definition of sequence $\{s_i(t)\}_{i=0}^{C}$, it is clear that $s_0\ge s_1\ge\cdots\ge s_C$, i.e. a monotone decreasing sequence. We first increase $s_i(0)$ such that $s_i(0)=s_{i-1}(0)-\varepsilon$, where $\varepsilon$ is a small constant. Let $v=\max(s_i(0)\cdot\lambda^{-(d^i-1)/(d-1)})^{1/d^i}$. In the original system $s_i(t)\le s_i(0)$ for all $t\ge 0$. Then, $s_i(t)\le\lambda^{(d^i-1)/(d-1)}\cdot v^{d^i}$. Based on the result in Mitzenmacher (1997), we conclude that $\{s_i(t)\}_{i=0}^{C}$ decreases doubly exponentially for all $t\ge 0$. Next, we further find the upper and lower bounds of the equilibrium points. □

Sequence $\{s_i(t)\}_{i=0}^{C}$ decreases doubly exponentially to a fixed point. Let $\vec{\pi}=[\pi_0,\pi_1,\ldots,\pi_C]$ denote the equilibrium points of states $\{s_i\}$ in the system (3.1). We now examine the expected time that a query spends in the homogeneous and bounded-capacity P2P system.

**Theorem 3.** *For $\lambda \in [0,1]$ and $d \geq 2$, $T_d(\lambda) \leq \alpha(\log T_1(\lambda))$, where $\alpha$ is a constant whose value depends only on d and C. Moreover,*

$$\lim_{\lambda \to 1^-} \frac{T_d(\lambda)}{\log T_1(\lambda)} = \frac{C}{\log_2^C \log d}. \tag{3.2}$$

**Proof.** First, we prove $\pi_i \leq \lambda^{(d^i-1)/(d-1)}$ by induction. For $i=0$, $\pi_0 = \lambda^{(d^i-1)/(d-1)}|_{i=0} = 1$. For $0 \leq i \leq k$, we assume $\pi_i \leq \lambda^{(d^i-1)/(d-1)}$. Then for $i=k+1$, we compare the equilibrium points of (3.1) and those of the following unbounded-capacity system

$$\begin{cases} \dot{s}_i = \lambda(s_{i-1}^d - s_i^d) - (s_i - s_{i+1}), & i \geq 1, \\ s_0 = 1. \end{cases} \tag{3.3}$$

Let $(\hat{\pi}_i)$ denote the equilibrium points of (3.3). In Mitzenmacher (1997), it was proved that $\hat{\pi}_i = \lambda^{(d^i-1)/(d-1)}$. Then, we only need to prove that $\pi_i \leq \hat{\pi}_i$ for $0 \leq i \leq C$, which are as follows. From (3.3), we have $\hat{\pi}_i = \lambda \hat{\pi}_{i-1}^d$, $i \geq 1$. According to (3.1), $\pi_i = \lambda(\pi_{i-1}^d - \pi_C^d)$, $1 \leq i \leq C$. Because $\pi_i$, $\hat{\pi}_i \geq 0$, $\pi_i \leq \lambda \pi_{i-1}^d$. Based on the assumption $\pi_i \leq \hat{\pi}_i$ for $i \in [0,k]$, we get $\pi_{k+1} \leq \lambda \pi_k^d \leq \lambda \hat{\pi}_k^d = \hat{\pi}_{k+1}$. Therefore, the equilibrium points $\pi_i \leq \lambda^{(d^i-1)/(d-1)}$ for $1 \leq i \leq C$. On the other end, we prove $\pi_i \geq a \hat{\pi}_i^b$, where $a = \lambda^{1/(d-1)}$ and $b = 2 - \log(1+\lambda)/\log\lambda$. The induction step is as follows: assume $\pi_i \geq a \hat{\pi}_i^b$ for $0 \leq i \leq k-1$, and then according to (3.1) $\pi_k + \lambda \pi_C^d = \lambda \pi_{k-1}^d \Rightarrow \lambda \pi_{k-1}^d \leq \pi_k + \lambda \pi_k^d \leq (1+\lambda)\pi_k$. Thus, $\pi_k \geq (\lambda/(1+\lambda))\pi_{k-1}^d \geq \lambda/(1+\lambda) \cdot a^d \cdot \hat{\pi}_{k-1}^{bd}$. Because $\hat{\pi}_k = \lambda \hat{\pi}_{k-1}^d$ from (3.3), we have $\pi_k \geq 1/(1+\lambda) \cdot a^{d-1}/\lambda^{b-1} \cdot a \hat{\pi}_k^b = (1/(1+\lambda)\lambda^{b-2}) a \hat{\pi}_k^b = a \hat{\pi}_k^b$.

Based on the result that $a \hat{\pi}_i^b \leq \pi_i \leq \hat{\pi}_i$ for $0 \leq i \leq C$, we can calculate the upper and lower bounds of $T_d(\lambda)$. $T_d(\lambda) = \sum_{i=1}^C i(\pi_{i-1}(t)^d - \pi_i(t)^d) = \sum_{i=0}^{C-1} \pi_i^d - C\pi_C^d = \sum_{i=1}^C \pi_i$. Therefore, $a \sum_{i=1}^C \hat{\pi}_i^b \leq T_d(\lambda) \leq \sum_{i=1}^C \hat{\pi}_i$. When $\lambda \to 1^-$, $\pi_i$ tends to be $\lambda^{\gamma(d^i-1)/(d-1)}$, where $\gamma$ is a constant within $[1, \frac{3}{2}]$. Therefore, $T_d(\lambda) = \sum_{i=1}^C \lambda^{\gamma(d^i-1)/(d-1)}$. Because $\prod_{i=0}^{C-1}(1+\lambda^{d^i}+\lambda^{2d^i}+\cdots+\lambda^{(d-1)d^i}) = (1-\lambda^{d^c})/(1-\lambda)$, we get $\sum_{i=0}^{C-1} \log(1+\lambda^{d^i}+\lambda^{2d^i}+\cdots+\lambda^{(d-1)d^i}) = \log(1-\lambda^{d^c}) - \log(1-\lambda)$. Based on the result in Mitzenmacher (1997), we have (3.2). □

Adding a constraint on node capacity makes it difficult to find the closed-form solution to (3.1) with parameters $\lambda$ and $d$. To analyze the dynamic behaviors of the P2P system with homogeneous and bounded nodal capacity, we conduct simulations to trace the changes of state variables in an example system. In the system, we have four peer nodes, each of which can queue up to six queries at a time, and the arrival rate of queries is $\lambda = 0.99$. The initial system is empty and queries come and leave the P2P system following the model described at the beginning of this section. Figs. 1(a)–(c) depict the dynamics of the node queue length, when one, two and three choices are made at random in probing. It is clear that as $d$ increases the number of nodes with the longest queues decreases. This is because incoming queries are more evenly distributed among nodes for greater $d$. Fig. 1(d) shows the value of $C/(\log(C/2)\log d) \cdot T_d(\lambda)/\log T_1(\lambda)$ converges closely to 1 as time goes on. There is a small deviation from 1, because the arrival rate of queries $\lambda$ is set to 0.99 in order to simulate $\lambda \to 1^-$. The system is stable and we have different trajectory for different value of $d$. From this figure, we can measure the expected time that a query stays in the P2P system, i.e. $T_d$.

We apply Kurtz's theorem (Shwartz and Weiss, 1995) to our randomized probing model ($d \geq 2$) to obtain bounds on the maximum load:

**Theorem 4.** *For any fixed time interval I, the length of the longest queue in an initially empty P 2P system with homogeneous and bounded node capacity over an interval [0,I] is $(C/(\log(C/2)\log d))\log\log N + O(1)$ with high probability, where C is the node capacity and the form of the $O(1)$ term depends on I and $\lambda$.*

Hence in comparing a system where queries have one choice with that having $d \geq 2$ choices, we can see that the second one produces an exponential improvement with regard to both the expected time of a query in the system and the maximum observed load of nodes, for sufficiently large $N$.

### 3.1.2. Heterogeneous and bounded case

In Section 3.1.1, we extended the supermarket model to analyze the effect of $d$-way random probing in balancing load in P2P systems with homogeneous and bounded nodal capacity. However, in practical P2P networks, participant nodes generally have different configurations. To tackle this nodal heterogeneity, we extend the preceding model to analyze behaviors of peer nodes with different capacities in face of randomized probing to balance load.

Here, we still consider P2P systems with static composition. Let us assume a system has $N$ peer nodes to process queries. Their correspondent capacities are $\{c_1, c_2, \ldots, c_N\}$, which are positive and finite. We assume $c_i$'s take nonuniform values, otherwise we can analyze the system by applying the model presented in Section 3.1.1. Next, we will try to model an investigate behavior of the P2P system with heterogeneous and bounded nodal capacity by utilizing results derived in the preceding section.

Let $c^*$ denote the maximum values in the sequence $\{c_i\}_{i=1}^N$. Then, we calculate the residue capacity as $\{c^* - c_i\}_{i=1}^N$. We treat these residue capacity as the initial load of their corresponding nodes. Thus, value of the state variables $s_i$ for $0 \leq i \leq c^*$ at time $t=0$ equals to $s_i(0) = |\{c_i | c_i \leq c^* - i\}|/N$, i.e. the fraction of nodes bearing initial load (residue capacity) no less than $i$. When the system runs and queries come/leave, the area of residue capacity is reserved and load changes in the rest area within $c^*$. With this transformation, peer nodes have homogeneous capacity as $c^*$ so that we can model the dynamic system by (3.1). The state variables satisfy the following equations:

$$\begin{cases} \dot{s}_i = \lambda(s_{i-1}^d - s_i^d) - (s_i - s_{i+1}), & i < C, \\ \dot{s}_C = \lambda(s_{C-1}^d - s_C^d) - s_C, \\ s_i(0) = \dfrac{|\{c_i | c_i \leq c^* - i\}|}{N}, & i \leq C, \\ s_i(t) \geq s_i(0). \end{cases} \tag{3.4}$$

For example, in a small-scale P2P network having four nodes, they can accommodate at most 3, 3, 4 and 6 queries at a time, respectively. Thus, $c^* = 6$ and their residue capacities are $\bar{c} = \{3,3,2,0\}$, which determines the initial load of the corresponding nodes $\{s_i(0)\}_{i=0}^6 = \{1, 0.75, 0.75, 0.5, 0, 0, 0\}$. The system dynamics can be modeled by (3.4) and its solution describes the steady states of the P2P system.

Eqs. (3.4) are established by exerting constraints on the initial values and the range of state variables to (3.1). Their equilibrium states have certain relations.

**Corollary 5.** *Let $\tilde{T}_d(\lambda)$ denote the expected time a query spends in the system* (3.4)*. Then, $\tilde{T}_d(\lambda)$ is bounded by*

$$T_d(\lambda) \leq \tilde{T}_d(\lambda) \leq T_d(\lambda) + S,$$

*where $T_d(\lambda)$ is the expected time a query spends in the homogeneous and bounded-capacity system* (3.1) *and S is a constant which equals to $\sum_{i=1}^C s_i(0)$.*

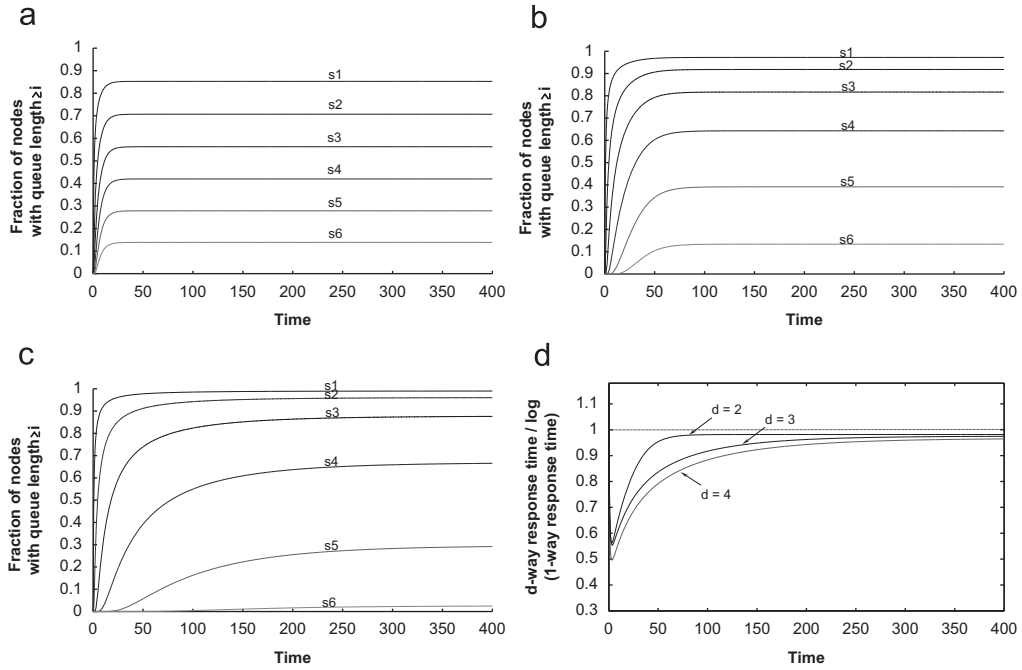


**Fig. 1.** Dynamics of query response time and node queue length in a simulated P2P network with homogeneous and bounded-capacity nodes. (a) State dynamics when $d=1$. (b) State dynamics when $d=2$. (c) State dynamics when $d=3$. (d) $\alpha T_d(\lambda)/\log T_1(\lambda)$ changes with time.

**Proof.** In Section 3.1.1, we prove that (3.1) converges doubly exponentially to its equilibrium state $\{\pi_i\}$. We now derive the solution to (3.4) based on $\{\pi_i\}$. Let $\{m_i\}_{i=1}^{C}=\{\max(\pi_i,s_i(0))\}_{i=1}^{C}$. We calculate the remains of $(\lambda m_{i-1}^d-m_i)$ for $1\le i\le C$. Let $r=\min\{(\lambda m_{i-1}^d-m_i)_{i=1}^{C}\}$. Then, we obtain the solution $(\tilde{\pi}_i)$ by iteratively computing $\tilde{\pi}_i=\lambda\tilde{\pi}_{i-1}^d-m_i$ for $1\le i\le C$, starting with $\tilde{\pi}_0=1$.

If $\{\pi_i\}$ is no less than the value of the initial state $\{s_i(0)\}$ in (3.4), then $\{\pi_i\}$ is also the steady state of (3.4). $\tilde{T}_d(\lambda)=\sum_{i=1}^{C} i(\tilde{\pi}_{i-1}^d-\tilde{\pi}_i^d)=\sum_{i=1}^{C}\tilde{\pi}_i$. Therefore, $\tilde{T}_d(\lambda)\ge T_d(\lambda)$. If $\{\pi_i\}$ is less than $\{s_i(0)\}$, we have $\tilde{\pi}_i\le\pi_i+s_i(0)$. The expected time a query spends in the system satisfies $\tilde{T}_d(\lambda)\le T_d(\lambda)+S$, where $S$ is a constant which equals to $\sum_{i=1}^{C} s_i(0)$. □

Based on Corollary 5, we apply Kurtz's theorem to derive the upper bound of the length of queues in the P2P system with heterogeneous and bounded nodal capacity.

**Theorem 6.** *For any fixed time interval $I$, the length of the longest queue in an initially empty P 2P system with heterogeneous and bounded nodal capacity for $d\ge 2$ over the interval $[0, I]$ is $(c^*/\log(c^*/2)\log d)\log\log N+O(1)$ with high probability, where $c^*$ is the upper bound of nodal capacity, and the $O(1)$ term depends on $c^*$, $s_i(0)$, $I$ and $\lambda$.*

The result of Theorem 6 is reduced to the one in Theorem 4 when the system is homogeneous. This theorem indicates that the maximum load of peer nodes is affected by the distributions of nodal capacity. However, the power of 2-way randomized probing is still valid in P2P networks with heterogeneous and bounded nodal capacity.

### 3.2. Node dynamics

The composition of a P2P network is dynamic. Compute nodes join and leave the P2P system in its lifetime. Guha et al. (2006) observed that only 30–40% supernodes were online at any given time in a P2P network. Nodal churn must be considered when we analyze the behaviors of randomized probing to balance load among peer nodes. In this section, we model the dynamic composition of peer nodes in a P2P network by using a random variable. Then we investigate the impact of nodal churn to the expect time a queue spends in the system and the longest length of queues among peer nodes. To make our analysis tractable, we first discuss randomized probing in a dynamic P2P network where nodes have infinite capacity, as in Section 3.2.1. Then, peer nodes with bounded capacities are investigated in Section 3.2.2.

#### 3.2.1. Dynamic nodes with unbounded capacity

In Section 3.1, we analyze the properties of randomized probing in static P2P systems by focusing on the factor of nodal capacity. In a dynamic P2P system with nodal churn, node composition is not fixed any more. We use a random variable $n$ to characterize the number of peer nodes that changes with time. Then, variable $m_i(t)$, the number of nodes whose loads are at least $i$, is also random. Their ratio $m_i(t)/n$ denoted by $s_i(t)$ still describes the fraction of peer nodes bearing loads that are at least $i$.

Existing research work (Yao et al., 2006; Stutzbach and Rejaie, 2006; Castro et al., 2005; Krishnamurthy et al., 2005; Rhea et al., 2004) on analyzing churn in P2P systems found that the arrival/departure processes of peer nodes can be modeled by a Poisson distribution when the system size becomes sufficiently large. Thus, we assume new nodes join the current P2P network in a Poisson distribution with rate $\lambda_{in}$ and existing nodes leave the system in a Poisson distribution with rate $\lambda_{out}$ relative to the node population, $\lambda_{in},\lambda_{out}<1$. When a peer node leaves, its original load will be removed from the P2P system. We assume the probability with which a node leaves the system is uniformly distributed among the existing nodes. As a result, the number of nodes join/leave the P2P system is $\Delta n=(\lambda_{in}-\lambda_{out})n\Delta t$ in a small time interval $\Delta t$. A new node can service coming queries when they are dispatched to it.

To characterize the system dynamics, we look into the change of random variable $m_i$, the number of nodes that have load at least $i$. Its value changes when a query is dispatched to a node queuing $i-1$ queries, or a query is serviced and removed from a node

having $i$ queries or such a node leaves the system. Therefore,

$$\frac{\Delta m_i}{\Delta t} = \lambda n\left[\left(\frac{m_{i-1}}{n}\right)^d - \left(\frac{m_i}{n}\right)^d\right] - n\left(\frac{m_i}{n} - \frac{m_{i+1}}{n}\right) - \lambda_{out} n \frac{m_i - m_{i+1}}{n}.$$

Thus, the influence of nodal churn on system behavior is incorporated in the value of random variable $m_i$. Because $s_i = m_i/n$, we have $\dot{s}_i = (\dot{m}_i n - m_i \dot{n})/n^2$. We use $\{s_i\}_{i=0}^{\infty}$ and $n$ as state variables and the state equations characterizing system dynamics are as follows.

$$\begin{cases} \dot{s}_i = \lambda(s_{i-1}^d - s_i^d) - (1+\lambda_{in})s_i + (1+\lambda_{out})s_{i+1}, & i \geq 1, \\ \dot{n} = (\lambda_{in} - \lambda_{out})n, \\ s_0 = 1, \end{cases} \tag{3.5}$$

with the initial condition $s_i(0) = 0$, $1 \leq i \leq n$ and $n(0) = N$, where $N$ is the number of nodes in the initial system.

The population of the P2P system depends on the values of $\lambda_{in}$ and $\lambda_{out}$. If $\lambda_{in} > \lambda_{out}$, $n$ tends to increase. On the other hand, when $n$ increases, more queries will arrive at the system. A similar situation happens when $\lambda_{in} < \lambda_{out}$. Thus, although nodes join/leave the system in runtime, the state variables $\{s_i\}_{i=0}^{\infty}$ can converge to steady state by adjusting the volume of queries.

**Corollary 7.** *Let $T_d(\lambda)$ denote the expected time a query spends in the system* (3.5). *Then, $T_d(\vec{\lambda})$ is bounded by*

$$0 \leq T_d(\vec{\lambda}) \leq \frac{1+\lambda_{in}}{1+\lambda_{out}} \lambda^{d-1} \hat{T}_d(\lambda),$$

*where $\vec{\lambda} = \{\lambda, \lambda_{in}, \lambda_{out}\}$ and $\hat{T}_d(\lambda)$ is the expected time a query spends in the homogeneous and infinite-capacity system* (Mitzenmacher, 2001).

**Proof.** Let $\dot{s}_i = 0$ in (3.5) for $i \geq 1$, and we get $s_{i+1} = (1/(1+\lambda_{out}))[(1+\lambda_{in})s_i - \lambda s_{i-1}^d + \lambda s_i^d]$.
$T_d(\vec{\lambda}) = \sum_{i=1}^{\infty} i(\pi_{i-1}^d - \pi_i^d) = \sum_{i=0}^{\infty} \pi_i^d$. By using induction, we can prove the expression in the corollary. □

Figs. 2(a)–(c) depict dynamics of the nodal queue length when one, two and three choices are made by random in node search: $\lambda_{in} = 0.2$, $\lambda_{out} = 0.1$ and $\lambda = 0.99$. It is clear that as $d$ increases the number of nodes with the longest queues decreases, because incoming queries are more balanced distributed among nodes for larger $d$. Fig. 2(d) presents the values of $(1+\lambda_{out})/(1+\lambda_{in})\log d \cdot T_d(\lambda)/\log T_1(\lambda)$ for as $\lambda \to 1^-$. We can see they converge closely to 1 as time goes on. The small deviation from 1 is because the arrival rate of queries $\lambda$ is set to 0.99 as to simulate $\lambda \to 1^-$.

We apply Kurtz's theorem (Shwartz and Weiss, 1995) to our randomized probing model in P2P network to obtain bounds on the maximum load:

**Theorem 8.** *In an initially empty P2P system where nodes' arrival/departure follows a Poisson distribution with rate $\lambda_{in}n$ and $\lambda_{out}n$ and nodes have infinite capacity, for any fixed $I$, the length of the longest queue in the system for $d \geq 2$ over the interval $[0, I]$ is $O(\log\log n)$ with high probability.*

### 3.2.2. Dynamic nodes with bounded capacity

A more general case is a P2P system that peer nodes arrive/depart at runtime and the participant nodes have heterogeneous and bounded capacities. In this section, we construct the state equations to describe the system dynamics and derive the bounds on length of the longest queue.

Let $c_1, c_2, \ldots, c_n$ denote the capacities of peer nodes in the system. The number of nodes $n$ is a random variable. We assume $\{c_i\}_{i=1}^{n}$ follows a Pareto distribution, with a shape parameter $k_c$. The number of nodes bearing load at least $i$, $m_i(t)$, follows the equations:

$$\begin{cases} \dot{m}_i &= \lambda n\left[\left(\frac{m_{i-1}}{n}\right)^d - \left(\frac{m_i}{n}\right)^d\right] - n\left[\frac{m_i}{n} - \frac{m_{i+1}}{n}\right] \\ &\quad + \lambda_{in} n \cdot \Pr\{c^* - c \geq i\} - \lambda_{out} n \frac{m_i - m_{i+1}}{n}, \quad i < c^*, \\ \dot{m}_{c^*} &= \lambda n\left[\left(\frac{m_{c^*-1}}{n}\right)^d - \left(\frac{m_{c^*}}{n}\right)^d\right] - n\frac{m_{c^*}}{n} - \lambda_{out} n \frac{m_{c^*}}{n}, \quad i = c^*, \\ \dot{n} &= (\lambda_{in} - \lambda_{out})n, \end{cases}$$

where $c^*$ is a sufficiently large value that is greater than any possible value of node capacity. According to the CDF of Pareto distribution, $\Pr\{c^* - c \geq i\} = \Pr\{c \leq c^* - i\} = 1 - ((c^* - i)/c_{min})^{-k_c}$, where $c_{min}$ is a minimum capacity.

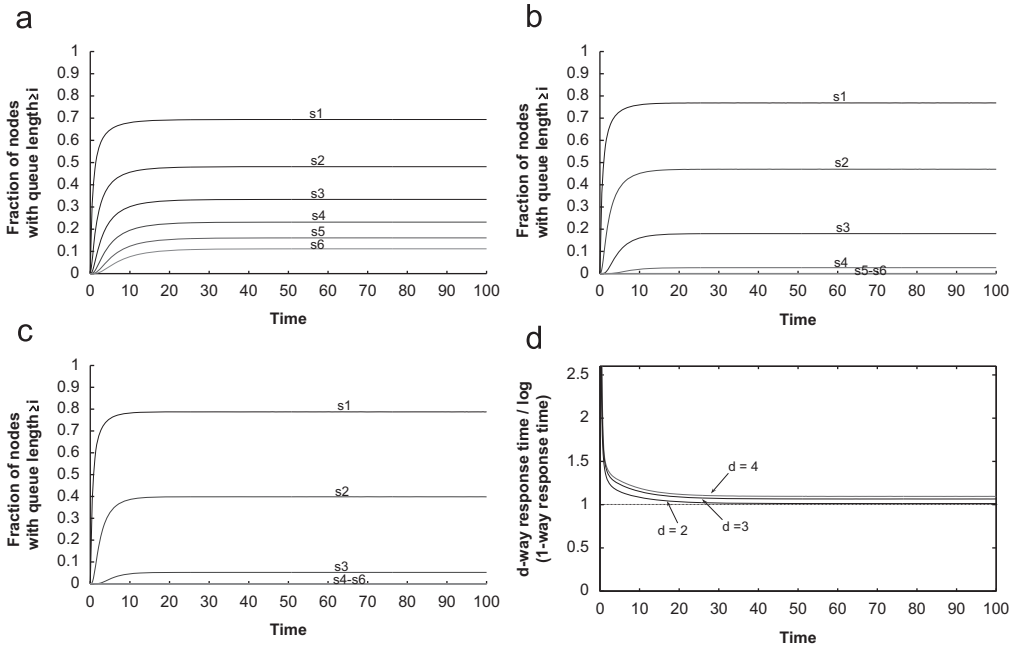


**Fig. 2.** Dynamics of query response time and nodal queue length in a simulated P2P network with churn and unbounded-capacity nodes. (a) State dynamics when $d=1$. (b) State dynamics when $d=2$. (c) State dynamics when $d=3$. (d) $\alpha T_d(\vec{\lambda})/\log T_1(\vec{\lambda})$ changes with time.

By applying $\dot{s}_i=(\dot{m}_i n-m_i\dot{n})/n^2$, we construct the following state equations:

$$\begin{cases} \dot{s}_i &= \lambda(s^d_{i-1}-s^d_i)-(1+\lambda_{in})s_i+(1+\lambda_{out})s_{i+1} \\ & \quad +\lambda_{in}\left[1-\left(\dfrac{c^*-i}{c_{min}}\right)^{-k_c}\right], \quad i\le c^*-c_{min}, \\ \dot{s}_i &= \lambda(s^d_{i-1}-s^d_i)-(1+\lambda_{in})s_i+(1+\lambda_{out})s_{i+1}, \quad c^*-c_{min}<i<c^*, \\ \dot{s}_{c^*} &= \lambda(s^d_{c^*-1}-s^d_{c^*})-(1+\lambda_{in})s_{c^*}, \quad i=c^*, \\ \dot{n} &= (\lambda_{in}-\lambda_{out})n, \\ s_i(0) &= \dfrac{|\{c_i|c_i\le c^*-i\}|}{N}, \quad i\le c^*, \\ s_i(t) &\ge s_i(0) \end{cases} \tag{3.6}$$

with initial condition $s_0(0)=1, s_i(0), 1\le i\le n(0)$ set according to the server configuration at $t=0$.

It is difficult to calculate closed form solutions to (3.6). However, it is numerically solvable. Figs. 3(a)–(c) show the dynamics of state variables $\{s_i\}$ in an example system. The system consists of 10 nodes at $t=0$. Their capacity are within the set of {6,7,…,10}, and for each value in the range there are two nodes having that capacity. $\lambda_{in}$, $\lambda_{out}$ and $\lambda$ are 0.2, 0.1 and 0.99, respectively. The capacity of newly joined nodes follows a Pareto distribution with shape parameter $k_c=2$ and the minimum capacity $c_{min}=5$. The figures show the system evolves to steady states as time goes on. We can calculate $T_d(\vec{\lambda})$ based on values of these steady states.

## 4. Experimental results

To quantify the performance of random choices schemes in real P2P systems, we conducted simulations on Cycloid (Shen et al., 2006) P2P network. Cycloid is a constant-degree DHT based on the network topology of cube-connected-cycle. We designed and implemented a simulator in Java for evaluation of the load balancing algorithms on Cycloid. Table 1 lists the parameters of the simulation and their default values.

GT-ITM (transit-stub and tiers) (Zegura et al., 1996) is a network topology generator, widely used for the construction of overlay networks. We used GT-ITM to generate transit-stub topologies for Cycloid, and get physical hop distance for each pair of Cycloid nodes. We selects the routing table entries pointing to the physically nearest among all nodes with nodeID in the desired portion of the ID space.

We use landmark clustering and Hilbert number (Xu et al., 2003) to cluster Cycloid nodes. Landmark clustering is based on the intuition that close nodes are likely to have similar distances to a few landmark nodes. Hilbert number can convert $d$ dimensional landmark vector of each node to one dimensional index while still preserve the closeness of nodes. We selected 15 nodes as landmark nodes to generate the landmark vector and a Hilbert number for each node cubic ID. Because the nodes in a stub domain have close (or even same) Hilbert numbers, their cubic IDs are also close to each other. As a result, physically close nodes are close to each other in the DHT's ID space, and nodes in one cycle are physically close to each other. For example, assume nodes $i$ and $j$ are very close to each other in physical locations but far away from node $m$. Nodes $i$ and $j$ will get approximately equivalent landmark vectors, which are different from $m$'s. As a result, nodes $i$ and $j$ would get the same

**Table 1**
Experiment settings and algorithm parameters.

| Environment parameter | Default value |
|---|---|
| Object arrival location | Uniform over Id space |
| Uniform over Id space | 4096 |
| Node capacity | Bounded Pareto: shape 2<br>Lower bound:2500, upper bound: 2500*10 |
| Number of items | 20 480 |
| Existing item load | Bounded Pareto: shape: 2<br>Lower bound: mean item actual load/2<br>Upper bound: mean item actual load/2*10 |

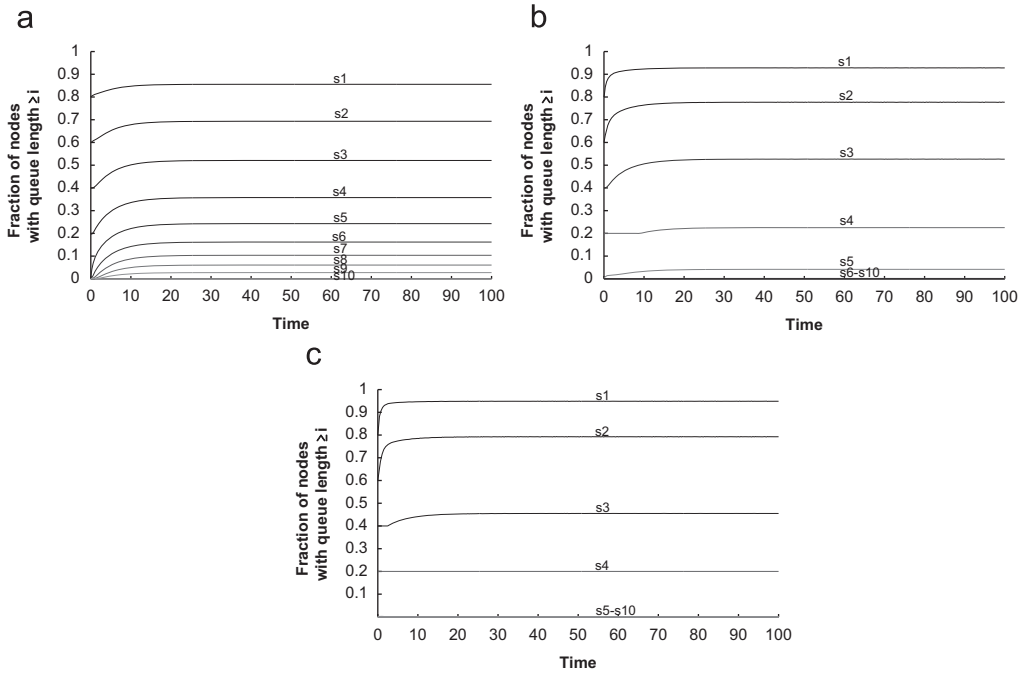


**Fig. 3.** Dynamics of query response time in a simulated P2P network with churn and heterogeneous, bounded-capacity nodes. (a) State dynamics when $d=1$. (b) State dynamics when $d=2$. (c) State dynamics when $d=3$.

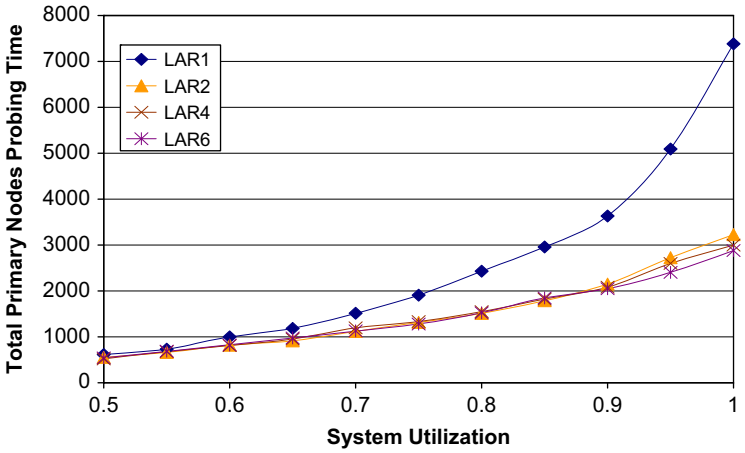


**Fig. 4.** Total node probing time.

cubic IDs and be assigned to the circle different from $m$'s. In the landmark approach, for each topology, we choose landmarks at random with the only condition that the landmarks are separated from each other by four hops.

Our experiments are built on two transit-stub topologies: "ts5k-large" and "ts5k-small" with approximately 5000 nodes each. In the topologies, nodes are organized into logical domains. We classify the domains into two types: transit domains and stub domains. Nodes in a stub domain are typically an endpoint in a network flow; nodes in transit domains are typically intermediate in a network flow. "ts5k-large" has five transit domains, three transit nodes per transit domain, five stub domains attached to each transit node, and 60 nodes in each stub domain on average. "ts5k-small" has 120 transit domains, five transit nodes per transit domain, four stub domains attached to each transit node, and two nodes in each stub domain on average. "ts5k-large" has a larger backbone and sparser edge network (stub) than "ts5k-small". "ts5k-large" is used to represent a situation in which Cycloid overlay consists of nodes from several big stub domains, while "ts5k-small" represents a situation in which Cycloid overlay consists of nodes scattered in the entire Internet and only few nodes from the same edge network join the overlay. To account for the fact that inter-domain routes have higher latency, each inter-domain hop counts as 3 hops of units of latency while each intra-domain hop counts as 1 hop of unit of latency. We run each trial of the simulation for $20T$ simulated seconds, where $T$ is a parameterized load balancing period, and its default value is set to 60 s in our test. The item and node join/departure rates are modeled by Poisson processes. The default rate of item join/departure rate is 0.4; that is, there are one item join and one item departure every 2.5 s. We range node inter-arrival time from 10 to 90 s, with 10 s increment in each step.

To balance load in a P2P network, probes are sent to a number of peer nodes. Among the responders, the one that has the least load will be selected. We refer to this class of randomized load balancing algorithms as $d$-way probing, denoted by $LAR_d$, $d=1,2,\ldots$. We compare the performance of 1, 2, 4, and 6-way random probe schemes in terms of node probing time and total number of load rearrangements. Figs. 4 and 5 show that the probing efficiency of $LAR_d$ ($d>2$) is almost the same as $LAR_2$, even though they need to probe more nodes than $LAR_2$. For example, as the system utilization reaches 85%, the probing time and number of load arrangements by $LAR_2$ are reduced by more than 38% and 35%, respectively, compared with those by $LAR_1$, while the performance is almost the same as that by $LAR_3$. The experimental results are consistent with the theorems that we derive in Sections 3.1 and 3.2 on the performance of randomized load balancing algorithms in P2P systems. A two-way probing method leads to an exponential improvement over one-way probing, but a $d$-way ($d>2$) probing leads to much less substantial additional improvement.

## 5. Related work

In most early DHT structures (Stoica et al., 2003; Manku et al., 2003; Malkhi et al., 2002), each node chooses at random a point in the hash space, typically, the unit interval [0, 1), and becomes associated with the points of the hash space closest to the selected point. Assuming random node departures, this scheme guarantees that the ratio of largest to average node segment size is $O(\log n)$, with high probability (Stoica et al., 2003). Virtual server approach has been used to mitigate imbalance of key assignment between nodes. It was proposed that each real server works as $\log n$ virtual servers, thus greatly decreasing the probability that some server will get a large part of the ring. Some extensions of this method were proposed in Rao et al. (2003) and Surana et al. (2006), where more schemes based on virtual servers were introduced and experimentally evaluated. However, these schemes assume that nodes are homogeneous, objects have the same size, and object IDs are uniformly distributed.

CFS (Dabek et al., 2001) accounts for node heterogeneity by allocating to each node some number of virtual servers proportional to the node capacity. In addition, CFS proposes a simple solution to shed the load from an overloaded node by having the overloaded node remove some of its virtual servers. However, this scheme may result in thrashing as removing some virtual servers from an overloaded node may result in another node becoming overloaded.

Byers et al. (2003) proposed the use randomized search to achieve better load balance. Each object is hashed to $d\geq 2$ different IDs, and is placed in the least loaded node of the nodes responsible for those IDs. The other nodes are given a redirection pointer to the selected node so that searching is not slowed significantly. For homogeneous nodes and objects and a static system, picking $d=2$ achieves a load balance within a $\log\log n$ factor of optimal. However, this scheme was not analyzed or simulated for the case of heterogeneous node capacities and node churn, which are defining characteristics of P2P networks. The paradigm of multiple random choices was also used in Shen and

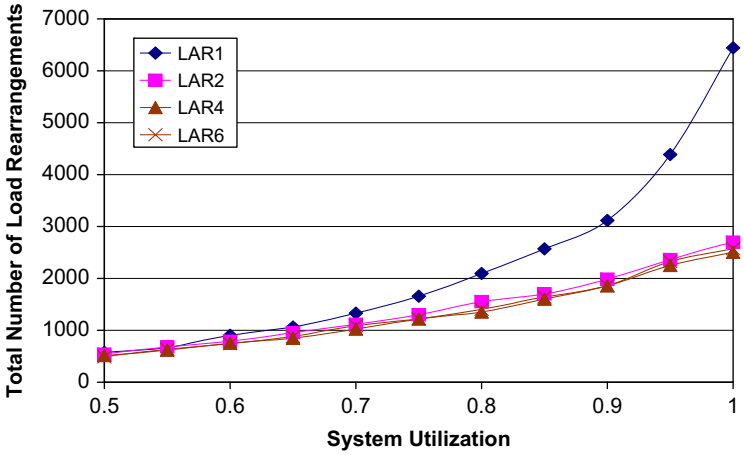


**Fig. 5.** Total number of load rearrangements.

Xu (2006, 2007), Kenthapadi and Manku (2005), and Giakkoupis and Hadzilacos (2005). Several peer nodes are probed before store a file to or dispatch a user query to the least loaded one.

Besides applying randomized probing algorithms to balance load in P2P networks, researchers also analyzed the characteristics of random choices theoretically. The main techniques used to analyze random choice problems are layered induction, witness trees, and fluid limits via differential equations. The *layered induction* technique pioneered by Azar et al. (2000). The random choice problem was modeled by balls-and-bins. It bounded the maximum load by bounding the number of bins with $k$ or more balls via induction on $k$. The layered induction approach provides nearly tight results. An alternative technique for handling the problem called the *witness tree* method (Cole et al., 1998). The key idea of this approach is to show that if a "bad event" occurs, i.e. if some node is heavily loaded, one can extract the history of the process a suitable tree of events called the witness tree. The probability of the bad event can then be bounded by the probability of occurrence of a witness tree. Generally, witness tree arguments involves the most complexity, and they have proved to be the most challenging in terms of obtaining tight results. The third technique studies algorithms that use random choices paradigm via *fluid limit* models (Mitzenmacher, 2001; Byers et al., 2004). The system dynamics can be described by a family of differential equations. This approach is simple and flexible. When the system dynamics can be modeled by this method, the differential equations generally yield accurate numerical results.However, these theory work analyzed a system where compute nodes have homogeneous and infinite capacities. Moreover, node churn, a defining characteristic of P2P systems, is not modeled by these approaches. In this paper, we analyze the dynamic behavior of random choice paradigm in general P2P systems, where peer nodes join/leave at runtime and they have heterogeneous and bounded capacities.

To develop resilient systems, Fu and Xu (2010, 2007a, 2007b) analyzed and modeled the failure correlation in large-scale compute cluster and grid systems. They also proposed failure-aware resource management mechanisms by using reconfigurable distributed virtual machines in networked computer environments (Fu, 2009, 2010a, 2010b).

## 6. Conclusions

In this paper, we model the randomized probing in general peer-to-peer systems. Theoretical analysis shows that two-way random probing is asymptotically superior to the one-way choice approach. However, by increasing the number of choices further, efficiency of the search algorithm does not improve significantly. Our random probing model is general in that the influence by nodal heterogeneity, capacity distribution and churn on search efficiency is investigated. The random approach is less sensitive to the node churn and heterogeneity in terms of the number of probes conducted before finding suitable nodes and the average response time of queries. Simulation and experiment results confirm our analysis. It is difficult to calculate the closed form solutions to state equations of the most general case. However, we can find the steady states numerically. For completeness in theory, we include the analysis of P2P systems consisting heterogeneous and bounded-capacity nodes with churn and simulation results. In this paper, we design and analyze the random probing algorithms within the context of P2P networks. However, our results have wide applicability and are of interest beyond the specific applications.

Although we analyze the performance of randomized probing in the general P2P environments, we still introduce some simplifying assumptions to make the problem tractable. One such assumption is that each peer node can service every query. However, in practice certain queries or requests may only be serviced by those nodes that have the required resources. To address this situation, we need to extend the composition model of a P2P network in a way that each query maps to a subset of nodes that can service it, and the changes of peer nodes' queue length will be quantified by distinguishing these different sets. However, this will make the analysis quite difficult. Another assumption is that the service time for a query is exponentially distributed with mean 1. In reality, different queries may require different amount of work and the processing power of peer nodes may vary. As a result, the service time follows a more complicated model. Although we do not discuss this case in our paper, our model can be extended to formulate the case by introducing other distributions of service time to the state equations of the system and numerically analyze the state dynamics. In this paper, we evaluate the performance of LAR, a generic load balancing algorithm based on $d$-way probing. We also plan to conduct more experiments to compare the performance of several other load balancing algorithms in addition to LAR.

## Acknowledgments

This research was supported in part by US NSF Grant CNS-0915396 and LANL Grant IAS-1103. The authors thank Dr. Mohammed Atiquzzaman and anonymous reviewers for their

constructive comments and suggestions. A preliminary version of this paper was published in the Proceedings of the 22nd ACM/IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2008 (Fu et al., 2008).

## References


Azar Y, Broder AZ, Karlin AR, Upfal E. Balanced allocations. SIAM Journal on Computing 2000;29(1):180–200 A preliminary version of this paper appeared in Proceedings of the 26th ACM symposium on Theory of computing (STOC), 1994.

Bienkowski M, Korzeniowski M, auf der Heide FM. Dynamic load balancing in distributed hash tables. In: Proceedings of international workshop on peer-to-peer systems (IPTPS), February 2005.

Byers J, Considine J, Mitzenmacher M. Simple load balancing for distributed hash tables. In: Proceedings of the 2nd international workshop on peer-to-peer systems (IPTPS), 2003.

Byers JW, Considine J, Mitzenmacher M. Geometric generalizations of the power of two choices. In: Proceedings of the 16th ACM symposium on parallelism in algorithms and architectures (SPAA), 2004.

Castro M, Costa M, Rowstron A. In: Proceedings of the 2nd symposium on networked system design and implementation (NSDI), 2005.

Cole R, Maggs BM, auf der Heide FM, Mitzenmacher M, Richa AW, Schrder K, et al. Randomized protocols for low-congestion circuit routing in multistage interconnection networks. In: Proceedings of the 13th ACM symposium on theory of computing (STOC), 1998.

Dabek F, Kaashoek MF, Karger D, Morris R, Stoica I. Wide-area cooperative storage with cfs. In: Proceedings of the 18th ACM symposium on operating systems principles (SOSP), 2001.

Fu S. Failure-aware construction and reconfiguration of distributed virtual machines for high availability computing. In: Proceedings of IEEE/ACM international symposium on cluster computing and the grid (CCGrid), 2009.

Fu S. Dependability enhancement for coalition clusters with autonomic failure management. In: Proceedings of IEEE international symposium on computers and communications (ISCC), 2010a.

Fu S. Failure-aware resource management for high-availability computing clusters with distributed virtual machines. Journal of Parallel and Distributed Computing 2010b;70(4):384–93.

Fu S, Xu C-Z. Exploring event correlation for failure prediction in coalitions of clusters. In: Proceedings of ACM/IEEE conference on supercomputing (SC), 2007a.

Fu S, Xu C-Z. Quantifying temporal and spatial correlation of failure events for proactive management. In: Proceedings of IEEE international symposium on reliable distributed systems (SRDS), 2007b.

Fu S, Xu, C-Z. Quantifying event correlations for proactive failure management in networked computing systems. Journal of Parallel and Distributed Computing 2010, doi:10.1016/j.jpdc.2010.06.010.

Fu S, Xu C-Z, Shen H. Random choices for churn resilient load balancing in peer-to-peer networks. In: Proceedings of the 22nd ACM/IEEE international symposium on parallel and distributed processing (IPDPS), 2008.

Giakkoupis G, Hadzilacos V. A scheme for load balancing in heterogenous distributed hash tables. In: Proceedings of the 24th ACM symposium on principles of distributed computing (PODC), 2005.

Godfrey B, Stoica I. Heterogeneity and load balance in distributed hash tables. In: Proceedings of IEEE conference on computer communications (INFOCOM), 2005.

Guha S, Daswani N, Jain R. An experimental study of the skype peer-to-peer voip system. In: Proceedings of 5th international workshop on peer-to-peer systems (IPTPS), 2006.

Kenthapadi K, Manku GS. Decentralized algorithms using both local and random probes for P2P load balancing. In: Proceedings of the 7th ACM symposium on parallelism in algorithms and architectures (SPAA), 2005.

Krishnamurthy S, El-Ansarh S, Aurell E, Haridi S. A statistical theory of chord under churn. In: Proceedings of 4th international workshop on peer-to-peer systems (IPTPS), 2005.

Malkhi D, Naor M, Ratajczak D. Viceroy: a scalable and dynamic emulation of the butterfly. In: Proceedings of the 21st ACM symposium on principles of distributed computing (PODC), 2002.

Manku G, Bawa M, Raghavan P. Symphony: distributed hashing in a small world. In: Proceedings of the 4th USENIX symposium on internet technologies and systems (USITS), 2003.

Mitzenmacher M. On the analysis of randomized load balancing schemes. In: Proceedings of ACM symposium on parallel algorithms and architectures (SPAA), 1997.

Mitzenmacher M. The power of two choices in randomized load balancing. IEEE Transactions on Parallel Distributed Systems 2001;12(10):1094–104.

Mitzenmacher M, Prabhakar B, Shah D. Load balancing with memory. In: Proceedings of the 43rd IEEE symposium on foundations of computer science (FOCS), 2002.

Rao A, Lakshminarayanan K, Surana S. Load balancing in structured P2P systems. In: Proceedings of the 2nd international workshop on peer-to-peer systems (IPTPS), 2003.

Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A scalable content addressable network. In: Proceedings of conference on applications, technologies, architectures, and protocols for computer communications (SIGCOMM), 2001.

Rhea S, Geels D, Roscoe T, Kubiatowicz J. Handling churn in a DHT. In: Proceedings of USENIX annual technical conference, 2004.

Rowstron A, Druschel P. Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems. In: Proceedings of the 18th IFIP/ACM international conference on distributed systems platforms (Middleware 2001), 2001.

Shen H, Xu C-Z. Elastic routing table with provable performance for congestion control in DHT networks. In: Proceedings of the 26th IEEE international conference on distributed computing systems (ICDCS), 2006.

Shen H, Xu C-Z. Locality-aware and churn-resilient load balancing algorithms in structured peer-to-peer networks. IEEE Transactions on Parallel and Distributed Systems 2007;18(6):849–62.

Shen H, Xu C-Z, Chen G. Cycloid: a constant-degree and lookup-efficient P2P overlay network. Performance Evaluation 2006;63(3):195–216.

Shwartz A, Weiss A. Large deviations for performance analysis: queues, communication and computing. London, UK: Chapman and Hall; 1995.

Stoica I, Morris R, Karger D, Kaashoek F, Balakrishnan H. Chord: a scalable peer-to-peer lookup service for internet applications. IEEE/ACM Transactions on Networking 2003;11(1):17–32.

Stutzbach D, Rejaie R. Understanding churn in peer-to-peer networks. In: Proceedings of ACM internet measurement conference (IMC), 2006.

Surana S, Godfrey B, Lakshminarayanan K, Karp R, Stoica I. Load balancing in dynamic structured peer-to-peer systems. Performance Evaluation 2006; 63(3):217–40.

Tewari S, Kleinrock L. Proportional replication in peer-to-peer networks. In: Proceedings of IEEE conference on computer communications (INFOCOM), 2006.

Xu C-Z, Lau F. Load balancing in parallel computers: theory and practice. Springer-Verlag, Kluwer Academic; 1997.

Xu Z, Tang C, Zhang Z. Building topology-aware overlays using global soft-state. In: Proceedings of the 23rd IEEE international conference on distributed computing systems (ICDCS), 2003.

Yao Z, Leonard D, Wang X, Loguinov D. Modeling heterogeneous user churn and local resilience of unstructured P2P networks. In: Proceedings of 14th international conference on network protocols (ICNP), 2006.

Zegura EW, Calvert KL, Bhattacharjee S. How to model an internetwork. In: Proceedings of IEEE conference on computer communications (INFOCOM), 1996.

Zhao BY, Huang L, Stribling J, Rhea SC, Joseph AD, Kubiatowicz J. Tapestry: an infrastructure for fault-tolerant wide-area location and routing. Journal on Selected Areas in Communications 2004;12(1):41–53.

Zheng C, Shen G, Li S, Shenker S. Distributed segment tree: support of range query and cover query over DHT. In: Proceedings of the 5th international workshop on peer-to-peer systems (IPTPS), 2006.

Zhu Y, Hu Y. Efficient, proximity-aware load balancing for DHT-based P2P systems. IEEE Transactions on Parallel and Distributed Systems 2005;16(4): 349–61.