IEICE TRANS. INF. & SYST., VOL.E92–D, NO.2 FEBRUARY 2009

PAPER

A Message-Efficient Peer-to-Peer Search Protocol Based on Adaptive Index Dissemination

Yu WU[†a)], ***Nonmember***, **Taisuke IZUMI**[††], **Fukuhito OOSHITA**[†], ***Members***, **Hirotsugu KAKUGAWA**[†], ***Nonmember***, ***and*** **Toshimitsu MASUZAWA**[†], ***Member***

SUMMARY Resource search is a fundamental problem in large-scale and highly dynamic Peer-to-Peer (P2P) systems. Unstructured search approaches are widely used because of their flexibility and robustness. However, such approaches incur high communication cost. The index-dissemination-based search is a kind of efficient unstructured search approach. We investigate such approaches with respect to minimize the system communication cost. Based on a dynamic system model that peers continuously leave and join, we solve two problems. One problem is how to efficiently disseminate and maintain a given number of indices. Another is to determine the optimal number of indices for each resource object of a given popularity. Finally, we propose an optimized index dissemination scheme which is fully decentralized and self-adaptive. A remarkable advantage is that the scheme yields no additional communication cost to achieve the self-adaptive feature.

key words: *peer-to-peer, search, message cost, popularity, adaptability, index-dissemination*

1. Introduction

1.1 Background

Peer-to-Peer (P2P) applications, with prominent advantages such as scalability, robustness and low development cost, have developed quickly in recent years [1]. Different from traditional client-sever systems that place resources in central servers, P2P systems distribute resources all over the system. Therefore, the resource search, which is to find out the peer storing the target object, is one of the fundamental building blocks for P2P systems [2]. (Throughout this paper, we use a term 'object' to abstract all kind of resources including files, services and so on). Since P2P systems are usually large-scale and highly dynamic, search protocols are desired to be scalable, robust and adaptive [3], [4].

Some early P2P systems (e.g. Napster) adopt centralized search schemes [5]. Such systems maintain a number of servers in which peers register their objects' indices, each of which is an entry including the name, location, and other related information of the corresponding object. By accessing one or more index servers, searchers can obtain the location of their target objects. A serious drawback of the central-server approach is that index servers suffer heavy load to process search queries when the system size becomes large.

Decentralized search schemes have attracted more attentions because of their load balancing property and scalability [6]–[8]. In such systems, the task of processing search queries are shared by all member peers so that the system may not have the bottleneck even in large-scale systems. Decentralized search protocols are usually classified into two categories: the structured search protocols and the unstructured ones [9]. Most of structured search protocols are based on the distributed hash table (DHT) technique [10]–[12]. In these protocols, each peer places objects' indices (or the object itself) in specified peers and organizes structured network topologies where search queries can be routed to the target object with a few messages. However, since in real P2P systems most of peers stay in the system for a short time, frequent updates are required to maintain topological structures, which is very costly. Unstructured search protocols, which are widely used in modern P2P systems, do not require specified network topologies [13]–[16]. Such protocols usually use a kind of random dissemination strategies (e.g. flooding or random walk) to distribute query messages over the system. Although they incur higher message cost in search processes than structured ones, the cost imposed by leave or join of peers is quite limited. This advantage is also favorable in respect to fault resilience.

To reduce search cost in unstructured protocols, it is effective in many cases to distribute indices (or replicas) of objects. Then, the object with more indices can be found easier. However, if an object has many indices, the dissemination and maintenance cost of those indices becomes large. This implies that there is a trade-off between the index maintenance cost and the search cost. The popularity-based dissemination which disseminates more indices for popular objects and less for unpopular ones, is an efficient strategy to reduce the total communication cost of the system.

1.2 Our Contribution

We focus on the unstructured search schemes with the index dissemination mentioned above. The problem is to find out the optimal index dissemination scheme that minimizes the total communication cost which includes both index maintenance cost and search cost.

The problem is firstly investigated by theoretical approaches. We analyze the system under an uniform-random access model that each peer randomly send messages to

Manuscript received March 26, 2008.
Manuscript revised August 13, 2008.

[†]The authors are with the Graduate School of Information Science and Technology, Osaka University, Osaka-fu, 560–0043 Japan.
[††]The author is with the Graduate School of Engineering, Nagoya Institute of Technology, Nagoya-shi, 466–8555 Japan.
a) E-mail: wu-yu@ist.osaka-u.ac.jp

DOI: 10.1587/transinf.E92.D.258

other peers. We also introduce a dynamic churn model that peers join and leave frequently. Since disseminated indices disappears by leave of peers, the object's holder has to disseminate indices periodically. Thus, in this paper, we consider the optimal index dissemination problem consists of the following two subproblems: The first one is to find the optimal scheduling for disseminating a given number of indices. We propose the *Stream Method* that is to averagely disseminate the same number of indices in each time unit. The *Stream Method* is proved to minimize the expected search cost with a given number of indices. The second subproblem is how many indices an object should have, considering its popularity. We show that the communication cost of an object is minimized when its index dissemination cost equals to its search cost, which is called *Equal Rule*. Based on the *Stream Method* and *Equal Rule*, we work out the optimal index number for each object and the lower bound of the system total communication cost (i.e. the sum of the minimum cost of each object) under the uniform-random access model.

Then, we propose a distributed protocol that optimizes the index dissemination and minimizes the system communication cost in a self-adaptive manner. Our protocol does not need any global informations (i.e. number of peers, objects' popularities and etc.). In addition, it yields almost no additional communication cost (other than that for index dissemination and search) to achieve the distributed and self-adaptive features. The performance of the protocol is justified by simulation.

1.3 Related Works

Quorum-based Search. The quorum-based search protocol formulated the index-dissemination-based search approach [17]. Under the uniform-random access model, the work presents a quantitative analysis of the hit rate with given number of indices and search size. However, it does not provide any optimization arguments. Based on the same search principle, we optimize the index dissemination scheme and minimize the system total communication cost. Our work can be regarded as a completed version of the quorum-based search.

Peer Sampling Services. The uniform-random peer sampling service is an important functional module for many distributed algorithms. It is also a fundamental assumption of this paper. Many works are proposed to implement it in distributed ways. For example, some works firstly construct a random network-like topology. Then if a peer sends a message to a randomly selected neighbour, the message can be regarded as being sent to a randomly selected peer from the system [18], [19]. One can also use the random walk to relay the message to multiple random destinations. The service can also be achieved by more sophisticated approaches such as the Metropolis-Hastings algorithm which is applicable in arbitrary network topologies [20].

Most of such algorithms works proactively to maintain a network [18], [19], or collecting neighbouring peers' informations [20], with $O(d)$ messages for each peer where d is the average degree. However, when sampling, they cost no additional messages. Comparing with the ideal random sampling, such distributed approaches may be not completely random so that a same peer may be multiply sampled more often. That may slightly reduces the performance, but the problem is usually not critical.

Square-Root Replication. The Square-Root Replication (SRR) is a optimized storage assignment principle for replica-dissemination-based search protocols [9], [21]. With a similar purpose, the SRR adopts popularity-biased replica dissemination to minimize system search cost. Different from our approach that disseminate indices, the number of disseminated replicas is limited by the system storage capacity because a replica's size is usually large. The SRR shows that the search cost for all objects can be minimized when the number of each object's replicas is proportional to the square root of the object's popularity. However, if the system storage capacity is small, it is still difficult to find objects because each object can have only a small amount of replicas. On the other hand, since the SRR does not consider the cost for file replication, the cost may be huge in a system having a large storage. Modern P2P systems often adopt index-dissemination for search and optimize the storage assignment only for download.

1.4 Organization

The organization of this paper are as follows: Section 2 introduces the system model and the principle of the index-dissemination-based search; Section 3 investigates the optimal index dissemination scheme; Section 4 presents an adaptive protocol to achieve the optimal index dissemination; Section 5 evaluates the protocol by simulation; Section 6 discusses some supplemental issues in real system environments; and finally we give concluding remarks in Sect. 7.

2. Preliminaries

2.1 System Model and Definitions

Throughout this paper, we adopt the discrete-time model where continuous time is divided into a series of discrete time intervals of the same length. Each interval is called a time unit. In each time unit, peers can execute one or more searches to find some resources. We assume that every search executed during time unit t is necessarily terminates within t. Notice that time units are introduced only to simplify the system and that we do not require peers to synchronize. That is, in the protocol proposed in this paper, peers do not aware the global clock.

A P2P system is defined by a dynamic set of peers in which peers join and leave frequently. In time unit t, $m(t)$ peers join the system. When a peer joins the system, it is

assigned a random lifetime L drawn from some distribution $l(\tau, t)$ [22], [23]. That is, in time unit t, let L be the lifetime, $\Pr[L = \tau] = l(\tau, t)$, $l(\tau, t) \geq 0$ for any τ and $\sum_{\tau=0}^{\infty} l(\tau, t) = 1$ for any t. The lifetime distribution can be arbitrary as long as the expectation $E[L] = \sum_{\tau=1}^{\infty} \tau \cdot l(\tau, t)$ is finite. The lifetime of each peer decreases by 1 per time unit. After the lifetime decreased to 0, the peer leaves the system. Re-joined peers are regarded as newly-joining peers, i.e. if a peer leaves the system, its historical information is vanished.

There are some objects $\{a, b, c \ldots\}$ in the system. Each object is independent, i.e. the copies of the same data item are regarded as the same object. The popularity $f_x(t)(\geq 0)$ of object x is defined by the the total number of times that x (including all copies of x) is searched during time unit t. The popularity of each object is independent of the others.

We assume an ideal random peer sampling service that enables a peer send messages to peers selected from the system uniformly at random, i.e. each peer is selected with the probability $1/n$ where n is the number of peers. The service is abstract that the detailed implementation, including network topology and the routing method, is not specified. We also assume the communication cost for the service is fixed for the system, i.e. the cost is not affected by the number of samplings. This assumption is reasonable as we mentioned in Sect. 1.3. The ideal assumption is only used to obtain a tight lower bound of the system communication cost. We will show latter that our protocol works well with non-ideal sampling services.

For simple presentation, we measure the communication cost by the number of transferred messages. (The term 'message cost' is used instead of the term 'communication cost'.) This metric is reasonable because the sizes of messages used in index-dissemination-based search protocols are almost equal regardless of their types (i.e. query or index). Notice the message cost is the logical communication cost on an overlay network. It does not represent physical distance between peers. One can consider that the message cost is the average physical communication cost for delivering a message between any two peers in the network.

Finally, we introduce some terms which will appear in the following of this paper. A peer which currently attend the system is called an *active* peer. The peer which stores a copy of an object is called the *owner* of the object. If an index is stored in an active peer and points to an active owner of the object, we say the index is *available*. The variables (functions) $m(t), l(\tau, t)$ and $f_x(t)$ are called *environment parameters*. The environment parameters are not known by any peers. In Sect. 3, we present theoretical contributions when system environment parameters are given. In Sect. 4, we propose a self-adaptive protocol to minimize the system message cost when those parameters are unknown and dynamic.

2.2 Index-Dissemination-Based Search

We introduce the framework of the index-dissemination-based search. It is mainly abstracted from the quorum-based search protocol [17]. The description of the framework is as follows:

- Index dissemination: The owner of an object x disseminates some indices of x to some peers selected uniformly at random from the system.
- Search process: The searcher sends a query messages to a peer selected uniformly at random. If the query message is received by the peer which holds an index of the target object (or the object itself), the search process succeeds. Otherwise, the searcher sends the query to another peer. This process is repeated until the target object is found.
- Index maintenance: Each index has a predefined time-to-live (TTL) value. An index will be deleted when its lifetime is expired. Indices are maintained by periodical re-dissemination by the owner of the object.

Notice that although the limited lifetime of indices may delete some available indices, it is favorable for fault tolerance because bad indices (i.e. the indices pointing to some disappeared objects that have been deleted or left with their owners) can stay in the system before their lifetimes expired. Moreover, from the viewpoint of load balance, it also avoids old peers, which have joined the system for long time, to store too much indices.

Comparing with the quorum-based search, the framework is more general in the sense that it allows arbitrary sizes for query and index quorums. We show some mathematical results of the framework below:

Lemma 2.1: Let n, q and p respectively be the number of peers, the number of available indices of an object in the system and the number of query messages used to search for the object. The success probability that the searcher find the target object is at least $1 - e^{-qp/n}$.

Proof: Since query messages are sent to the peers selected uniformly at random, each query finds the target object' index with a probability q/n. Thus the object can be found with a probability

$$\begin{aligned}\rho &= 1 - (1 - q/n)^p \\ &\geq 1 - e^{-qp/n}.\end{aligned}$$

□

Lemma 2.2: Let n, q and p respectively be the number of peers, the number of available indices of an object in the system and the query messages used until the index of the target object is found. The expectation of p is $\mathrm{E}[p] = n/q$.

Proof: The probability that the first index of the object is found by the exactly the k th probing is

$$\Pr[p = k] = (1 - q/n)^{(k-1)} \cdot q/n.$$

That is, there must be $k - 1$ failed probes followed by the successful one. Thus, the random variable p follows a *geometric distribution* that each probe succeeds with the probability q/n. Therefore, we obtain $\mathrm{E}[p] = n/q$ [24].

□

3. Optimization of Index Dissemination

In this section, we analyze the system in a stable environment that $m(t) = m$, $m > 0$; $l(\tau, t) = l(\tau)$, $l(\tau) \geq 0$; $f_x(t) = f_x$, $f_x > 0$. Clearly, if t is large enough, the expected number of the peers join in time unit $t - i$ is $m \sum_{\tau=i}^{\infty} l(\tau)$. So the number of peers in a stable system converges to $n = m \sum_{i=1}^{\infty} \sum_{\tau=i}^{\infty} l(\tau)$. Then, we have

$$n = m \sum_{i=1}^{\infty} \sum_{\tau=i}^{\infty} l(\tau) = m \sum_{\tau=1}^{\infty} \sum_{i=1}^{\tau} l(\tau) = mE[L]. \quad (1)$$

As shown in Fig. 1, if m and t are large enough, we can approximately consider the number of peers is fixed to n.

3.1 Formulation of the System Message Cost

We consider the system message cost consists of the search cost and the index maintenance cost of all objects in the system. Notice the cost for the random peer sampling service is not omissible. However, as we mentioned in Sect. 1.3, those algorithms work proactively and that their cost is fixed for each peer [17], [18]. Therefore, the sampling cost does not affect the trade-off between index maintenance cost and search cost. For simple presentation, we do not count it in the following of the paper.

Due to independence of the message cost related to each object, the system message cost is minimized iff the message cost related to each object is minimized. Therefore, in the following of the paper, we focus on how to minimize the total message cost related to an single object x. Below, we give the definition of the system message cost, a summary is listed in Table 1.

Definition 3.1: (Search size). The search size, denoted by $p_{x,s}(t)$, is the number of search queries each searcher s uses to find object x in time unit t. The search size $p_{x,s}(t)$ is a random variable.

Definition 3.2: (Search cost). The search cost, denoted by $s_x(t)$, is the total number of query messages used to find object x during time unit t. That is, $s_x(t) = \sum_{\forall s} p_{x,s}(t)$. Since $p_{x,s}(t)$ is a random variable, $s_x(t)$ is also a random variable.

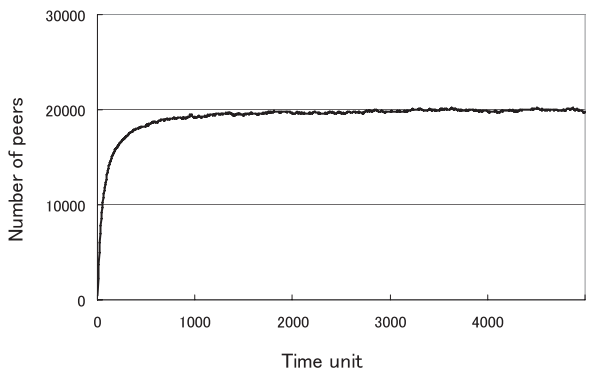


Fig. 1 The number of peers in a newly created system. $m = 400$; Pareto distribution for peers' lifetime: Prob$[L \leq \tau] = 1 - (1 + \tau/50)^{-2}$ that implies $E[L] = 50$, $n = 20000$.

Definition 3.3: (Index maintenance cost). The index maintenance cost, denoted by $q_x(t)$, is the number of the indices for object x disseminated during time unit t.

Definition 3.4: (Message Cost). The message cost, denoted by $m_x(t)$, is the sum of the index maintenance cost $q_x(t)$ and the search cost $s_x(t)$ of object x during time unit t. That is, $m_x(t) = q_x(t) + s_x(t)$. Since $s_x(t)$ is a random variable, $m_x(t)$ is also a random variable.

By the definitions, the message cost of object x is $m_x(t) = q_x(t) + \sum_{\forall s} p_{x,s}(t)$. Letting $M_x(t)$, $S_x(t)$ and $P_x(t)$ be the expectations of $m_x(t)$, $s_x(t)$ and $p_{x,s}(t)$ respectively, we obtain

$$M_x(t) = q_x(t) + S_x(t) = q_x(t) + f_x \cdot P_x(t). \quad (2)$$

Notice that no matter which peer is the searcher, the expected search size is the same because each searcher sends query messages to randomly selected peers in the system.

3.2 Index Dissemination Method

A disseminated index may disappear in two cases. One case is that the index's TTL value is expired. Anther case is that the peer which stores the index leaves the system. Therefore, the number of available indices for each object is decided by not only how many but also when those indices were disseminated.

The leave of peers can be described as follows. In time unit t, the expected number of peers with lifetime τ is $\eta(\tau) = m \sum_{i=0}^{\infty} l(\tau + i)$ where $ml(\tau + i)$ is the expected number of peers joined in time unit $t - i$. Those $\eta(\tau)$ peers will leave the system in time unit $t + \tau$. We define a damping function by $d(\tau) = \sum_{i=1}^{\tau} \eta(i)/n$ which indicates the probability that peers in the current system leave after τ time units. Clearly, $d(\tau)$ is monotonically increasing and $0 \leq d(\tau) \leq 1$ for any τ. Notice $\eta(\tau)$ and $d(\tau)$ are independent of t.

For case study, we analyze two index dissemination methods: the *burst method* which is used in the quorum-based search protocol [17], and the *stream method* we newly proposed. In the burst method, the owner of an object x disseminates Q_x indices once per T time units (called a TTL cycle), where T is the predefined TTL value of indices. In this case, the number of available indices decreases during each TTL cycle. In the τ-th time unit of each TTL cycle, the expected number of available indices $q_x^B(\tau)$ is

$$q_x^B(\tau) = Q_x \cdot (1 - d(\tau)). \quad (3)$$

In contrast, in the stream method, the owner disseminates Q_x/T indices in each time unit. In this case, the expected number of available indices $q_x^S(t)$ is fixed that

Table 1 Symbols' definition, time unit t.

n	The number of peers in the system.
m	The number of peers join the system in each time unit.
$l(\tau)$	The lifetime distribution of peers.
$d(\tau)$	The probability of peers leave after τ time units.
f_x	The popularity (search frequency) of object x.
$p_{x,s}(t)$	(A random variable). The search size for searcher s to find object x.
$P_x(t)$	The expectation of $p_{x,s}(t)$.
$s_x(t)$	(A random variable). The search cost of object x, $s_x(t) = \sum_{\forall s} p_{x,s}(t)$.
$S_x(t)$	The expectation of $s_x(t)$, $S_x(t) = f_x \cdot P_x(t)$.
$q_x(t)$	The index maintenance cost of object x.
$m_x(t)$	(A random variable). The message cost of object x, $m_x(t) = q_x(t) + s_x(t)$.
$M_x(t)$	The expectation of $m_x(t)$, $M_x(t) = q_x(t) + S_x(t) = q_x(t) + f_x P_x(t)$.

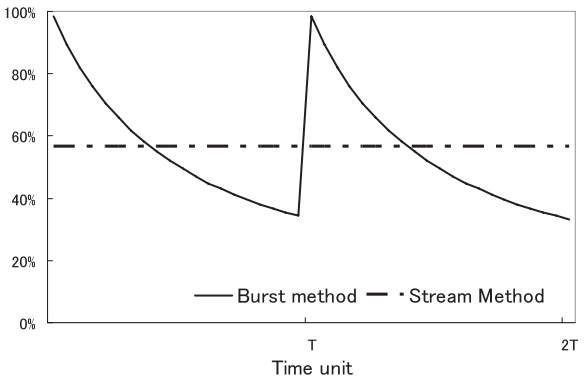


Fig. 2 The percentage of the number of available indices number, $T = 50$, $m = 400$; Pareto distribution for peers' lifetime: Prob$[L \le \tau] = 1 - (1 + \tau/50)^{-2}$.

$$q_x^S(t) = \sum_{\tau=1}^{T} (1 - d(\tau)) \cdot Q_x/T. \tag{4}$$

Figure 2 shows the percentage of the number of available indices adopting two index dissemination methods. The same number of indices, marked by 100%, are disseminated in each TTL cycle.

Let M_x^B and M_x^S be the average message cost of the burst method and the stream method respectively. According to lemma 2.2 and equality 2,we obtain:

$$\begin{aligned} M_x^B &= \frac{Q_x}{T} + \frac{1}{T} \sum_{\tau=1}^{T} \frac{f_x n}{Q_x \cdot (1 - d(\tau))} \\ &= \frac{Q_x}{T} + \frac{f_x n}{Q_x T} \sum_{\tau=1}^{T} \frac{1}{(1 - d(\tau))}; \end{aligned} \tag{5}$$

$$\begin{aligned} M_x^S &= \frac{Q_x}{T} + \frac{f_x n}{\sum_{\tau=1}^{T} (1 - d(\tau)) \cdot Q_x/T} \\ &= \frac{Q_x}{T} + \frac{f_x n T}{Q_x} \frac{1}{\sum_{\tau=1}^{T} (1 - d(\tau))}. \end{aligned} \tag{6}$$

Then we have

$$\begin{aligned} M_x^B &= \frac{Q_x}{T} + \frac{f_x n}{T Q_x} \sum_{\tau=1}^{T} \frac{1}{(1 - d(\tau))} \\ &\ge \frac{Q_x}{T} + \frac{f_x n}{Q_x} \frac{1}{\sqrt[T]{\prod_{\tau=1}^{T} (1 - d(\tau))}} \\ &\ge \frac{Q_x}{T} + \frac{f_x n T}{Q_x} \frac{1}{\sum_{\tau=1}^{T} (1 - d(\tau))} \\ &= M_x^S. \end{aligned}$$

Therefore, the stream method achieves lower search cost than the burst method even the index maintenance cost is the same. Actually, we can show that the stream method is the best index dissemination scheme in the sense that it minimizes the expected search cost.

Theorem 3.1: The stream method is the optimal index dissemination method that minimizes the expected search cost under a fixed index maintenance cost.

Proof: We assume the number of indices being disseminated in each time unit follows a periodic function $g(\tau)$, $1 \le \tau \le \Gamma$, where Γ is the cycle of $g(\tau)$. Without loss of generality, we assume $\Gamma > T$. (By combining several consecutive short cycles, we can regard $g(t)$ as a function of a long cycle.) The expected number of available indices in the τ-th time unit, denoted by $a(\tau)$, is

$$a(\tau) = \sum_{t=\tau-T}^{\tau-1} g(t)(1 - d(\tau - t)), \tag{7}$$

where a nonpositive time label t indicates the $(\Gamma - t)$th time unit of the previous cycle. Clearly, for any τ, $a(\tau) \ge 0$.

Letting q be the fixed average number of the indices being disseminated in each time unit, we obtain

$$\sum_{\tau=1}^{\Gamma} g(\tau) = \Gamma \cdot q. \tag{8}$$

By Equality 7 and 8, we obtain

$$\sum_{\tau=1}^{\Gamma} a(\tau) = q\Gamma \sum_{t=1}^{T} (1 - d(t)).$$

Letting S_x be the sum of the expected search cost in each cycle, we obtain:

$$\begin{cases} S_x = \sum_{\tau=1}^{\Gamma} f_x \cdot n/a(\tau) \\ \sum_{\tau=1}^{\Gamma} a(\tau) = q\Gamma \sum_{t=1}^{T} (1 - d(t)). \end{cases} \tag{9}$$

Notice $\sum_{\tau=1}^{\Gamma} a(\tau)$ is a finite constant which is independent of both t and $g(t)$. Therefore, by basic inequalities, we can obtain

$$S_x \ge f_x \cdot n \cdot \Gamma/q \sum_{t=1}^{T} (1 - d(t)).$$

Equality holds when

$$\forall \tau (1 \le \tau \le \Gamma),\ a(\tau) = q \sum_{t=1}^{T} (1 - d(t)).$$

Therefore, the search cost is minimized when the number of

available indices is uniform in each time unit. Trivially, it is only achievable by the stream method.

The theorem holds even if we consider non-periodical dissemination methods because the same argument is possible if Γ is infinitely long. □

3.3 Optimal Index Number

Next, we investigate the minimum message cost when adopting the stream method. By Equality 6 and the basic inequality $x + C/x \geq 2\sqrt{C}$, we obtain the minimum message cost $\min[M_x]$ of object x that

$$\min[M_x] = 2\sqrt{\frac{f_x n}{\sum_{\tau=1}^{T}(1 - d(\tau))}}. \quad (10)$$

Then, the optimal number of indices, denoted by $\hat{q_x}$, to be disseminated in each time unit is

$$\hat{q_x} = \sqrt{\frac{f_x n}{\sum_{\tau=1}^{T}(1 - d(\tau))}}. \quad (11)$$

Equality 10 shows the theoretical lower bound of the total message cost. The result indicates that there can not be any implementations of the random sampling service or any optimizing strategy of index dissemination can solve the search problem with less cost, as long as the system accord with the uniform-random sampling model.

4. A Self-Adaptive Protocol

In Sect. 3, we obtained the optimal index number $\hat{q_x}$. However it can not be directly computed from Equality 11 because m, $l(\tau)$ and f_x are not known by any peer. In this section, we propose a self-adaptive protocol that implements the optimal index-dissemination without those global parameters.

4.1 The Equal Rule

Theorem 4.1: (The Equal Rule). The message cost of an object is minimized when its index maintenance cost equals to its search cost.

Proof: Let $\hat{s_x}(t)$ be the expected search cost of x when $q_x(t) = \hat{q_x}$. According to Equality 10 and 11, we obtain

$$\hat{s_x}(t) = \min[M_x] - \hat{q_x} = \hat{q_x}. \quad (12)$$

□

The Equal Rule indicates that, if we disseminate the same number of indices as the number of the search queries disseminated in each time unit, the total message cost is minimized. By the Equal Rule, we obtain the skeleton of our protocol below:

- Search: When searching for an object, the searcher repeatedly sends query messages to randomly selected peers until the object is found. During the search, the searcher counts the number of query messages used.
- Index maintenance: After the search succeeds, the searcher disseminates the same number of indices to some randomly selected peers. Each index has a lifetime counter which is increased per time unit. An index will be deleted when its lifetime counter exceeds the predefined TTL value.

Notice that we allow the searchers to disseminate the indices. Since the searcher usually downloads the target object after finding it, the indices disseminated by the searcher can include anyone of two object locations; the searcher or the original owner. Such flexibility is favorable in terms of load balancing.

4.2 Index Dissemination Schemes

The following factors should be considered when each peer disseminates indices. First, as we show in Theorem 3.1, the search cost is minimized when the number of available indices is stable. However, it may make the number of available indices instable to disseminate straightforwardly the same number of indices at each time unit because of the fluctuation of search cost by the random noise effect. Second, the system environment parameters are usually not static, even change rapidly at sometimes. For example, when an object becomes a hot spot, its popularity, together with the search cost, drastically increases in a short period of time [4]. The number of indices should adapt to such changes.

By referring the statistical estimation methods, we propose three approaches for deciding the number of indices to be disseminated.

- RT (Real-Time) mode:

$$q_x(t) = s_x(t-1).$$

- SMA (Simple Moving Average) mode:

$$q_x(t) = \sum_{\tau=1}^{T} s_x(t-\tau)/T.$$

- EMA (Exponential Moving Average) mode:

$$q_x(t) = \sum_{\tau=1}^{T} s_x(t-\tau) \cdot 2^{-\tau}.$$

The above descriptions indicate how we use the historical information of search cost to decide the number of disseminated indices in the global view. In the followings, the index dissemination schemes from the viewpoint of each searcher are described. After the searcher s completes the search process for object x by $p_{x,s}(t)$ query messages in time unit t, it disseminates some indices, denoted by $q_{x,s}(t+\tau), 0 \leq \tau \leq T-1$, in the following T time units:

- RT mode:

$$\begin{cases} q_{x,s}(t+\tau) = p_{x,s}(t), & \tau = 0 \\ q_{x,s}(t+\tau) = 0, & \tau > 0. \end{cases}$$

- SMA mode:

$$q_{x,s}(t+\tau) = p_{x,s}(t)/T, \quad 0 \le t \le T-1.$$

- EMA mode:

$$q_{x,s}(t+\tau) = p_{x,s}(t) \cdot 2^{-1-\tau}, \quad 0 \le t \le T-1.$$

Obviously, there is a trade-off between the stability and adaptability. The RT mode has the fastest adaptation speed when the system environment parameters change. However it works in the stream method only when the object is frequently searched. In the contrast, the SMA mode can stabilizes the system well because the indices are disseminated averagely during the following T time units after each search event. However it may not be able to adapt to a highly dynamic system environment. The EMA mode is a middle approach between RT and SMA modes.

5. Simulation

In Sect. 3, we have the lower bound of the index-dissemination based search under the random peer sampling model. And in Sect. 4, we proposed a distributed protocol to achieve it. In this section, we compare the message cost of our protocol with the theoretical minimum message cost to justify its effectiveness. Unfortunately, we can not find any related works for comparative evaluation. For example, as we mentioned in Sect. 1.3, the quorum-based search and the square-root replication principle are not optimized for the total message cost, so fair comparison with them are impossible.

This section is divided to two parts. Section 5.1 justifies the adaptability of the protocol and compare the performance of the three index dissemination schemes. Section 5.2 justifies the practical impact of the protocol under realistic system environment settings.

5.1 Adaptability

According to the theoretical analysis, we know that the *Stream Method* and *Equal Rule* are the necessary conditions of the optimal index dissemination. However the theoretical results are obtained in a stable system environment. It is unclear how our protocol performs in unstable system environments because the *Equal Rule* is difficult to achieve in those cases. Moreover, because our protocol disseminate indices after each search, the search frequency decides the index dissemination timing that affects the *Stream Method*. This subsection evaluate the protocol with dynamic environment parameters and compare the performance of the three index dissemination schemes with some special simulation settings.

The environment parameters include $m(t), l(\tau, t)$ and $f_x(t)$. We mainly evaluate the protocol under dynamic settings of $f_x(t)$ because $m(t)$ and $l(\tau, t)$ do not vary quickly (often change in cycles of one day) in large-scale systems [25] and they do not affect the index dissemination timing. The simulation environments are as follows. In each time unit, 400 peers join the system (i.e. $m = 400$). The lifetime distribution of each peer is drawn from the Pareto distribution which is proved to be the peers' lifetime distributions in many real P2P systems [22]. The cumulative distribution function (CDF) of the distribution is $l_C(\tau) = \text{Prob}[L \le \tau] = 1 - (1 + \tau/50)^{-2}$ which implies $E[L] = 50$. We execute the protocol under the ideal random sampling service to estimate the best performance of the proposed protocol. The initial lifetime of indices is set to $T = 50$.

This time we evaluate only one object x to show the difference of the three index dissemination schemes clearly. Notice our protocol minimize the message cost related to each object respectively, the distribution of objects' popularities do not affect the evaluation result. In time unit t, $f_x(t)$ searchers are selected randomly from the system. To have a stable result against the randomness of the protocol, we repeat the simulation for 1000 times and show the sum of the message cost in each independent execution.

The simulation results are shown by the total message cost in each time unit. In our protocol, the search process continues until the target object is found, i.e. the success rate is always 1. Notice the result consists of both the maintenance cost (i.e. index dissemination cost) and search cost (i.e. query dissemination cost). Since the protocol is designed according to the *Equal Rule*, The maintenance and search cost occur exactly 50% of the total cost. Then from the results and $f_x(t)$, one can easily obtain the average search size and the number of indices disseminated. Such data will not be shown respectively for lack of space. For comparing, we show the theoretical minimum message cost by the curve 'Ideal'.

Figure 3 shows the results for discontinuous change of search frequency. All of the three schemes can converge to the theoretical minimum message cost and stabilize within $2T$ time units. The RT mode quickly responses but has the highest peak traffic. In Fig. 4 and 5, the search frequency changes continuously. Figure 4 shows the message cost under a slowly changing $f_x(t)$. In this case, all the three methods work as expected. However, when the $f_x(t)$ changes

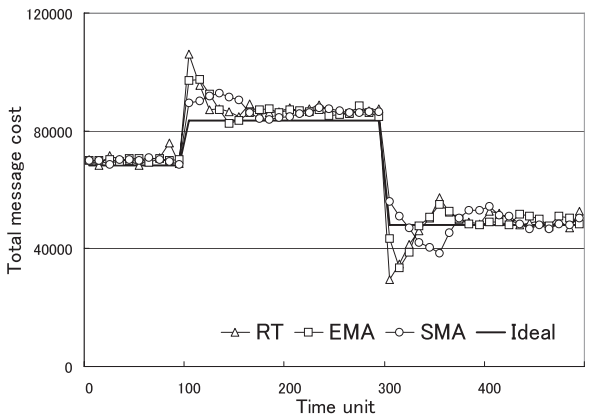


Fig. 3 The total message cost (the sum of 1000 executions), $f_x(t) = 2$ for $t \in [0, 100)$; $f_x(t) = 3$ for $t \in [100, 300)$; $f_x(t) = 1$ for $t \in [300, 500)$.

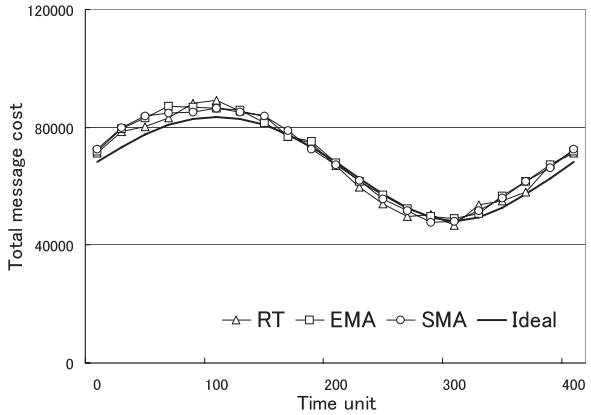


Fig. 4 The total message cost (the sum of 1000 executions), $f_x(t) = 2 + \sin(2\pi t/400)$.

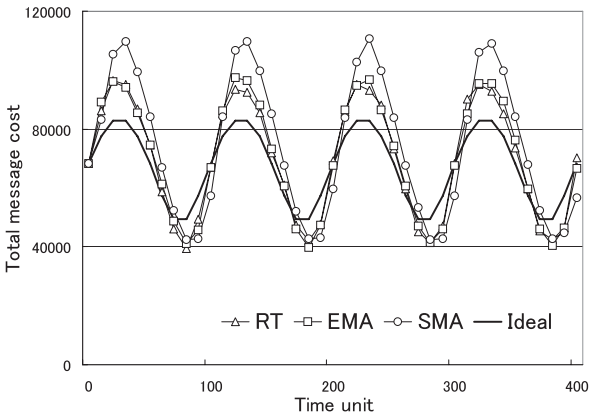


Fig. 5 The total message cost (the sum of 1000 executions), $f_x(t) = 2 + \sin(2\pi t/100)$.

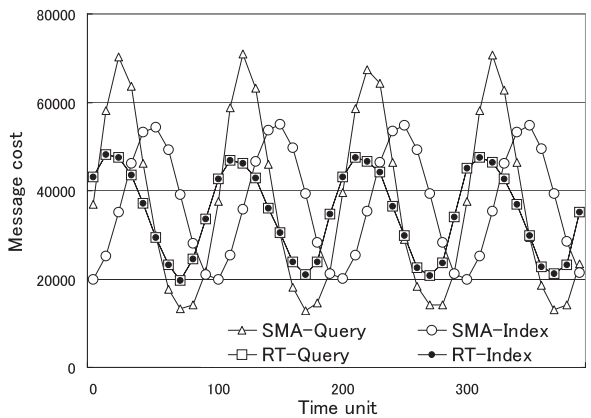


Fig. 6 The message cost of index and query dissemination in RT and SMA mode respectively (the sum of 1000 executions); $f_x(t) = 2 + \sin(2\pi t/100)$.

rapidly (Fig. 5), we can see all the three curves depart from the curve 'Ideal' and incur higher message cost. Especially the SMA mode consumes more messages than others. As Fig. 6 shows, in each time unit, the number of index and query messages are quite different in the SMA mode al-

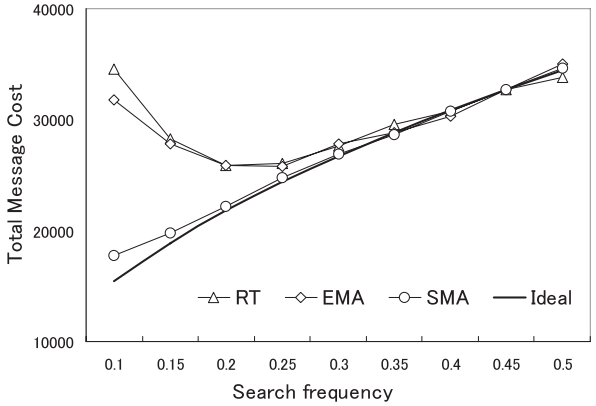


Fig. 7 The total message cost (the sum of 1000 executions), $0.1 \leq f_x \leq 0.5$.

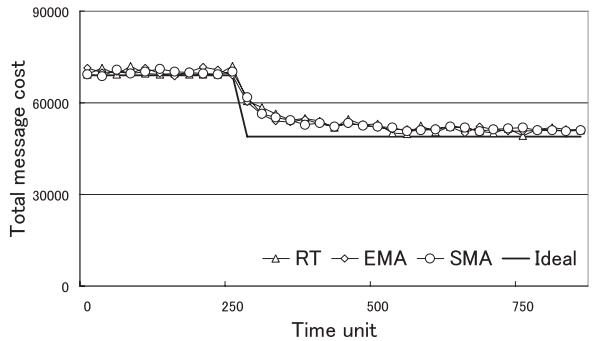


Fig. 8 The total message cost (the sum of 1000 executions), $f_x = 2$; $m(t) = 400$ for $t \in [0, 250)$, $m(t) = 200$ for $t > 250$.

though the total numbers are the same during the simulation. That implies the *Equal Rule* is not well achieved by the SMA mode in highly dynamic environments. As the result, the SMA mode cost more messages than the RT mode which achieves the *Equal Rule* much better.

Figure 7 shows the result when the object is rarely searched. We can see, when the f_x is lower than 0.2, both the RT mode and the EMA mode incurs much higher message cost than the theoretical result. Because the index dissemination process is executed after each search, the RT mode can not achieve the *Stream Method* well when the search frequency is low. The simulation result also implies that the *Stream Method* is much efficient than the *Burst Method*.

The above results show the trade-off between adaptability and stability of the three index dissemination schemes. The RT mode and the EMA mode have good adaptability while the SMA mode has good stability. However, in many P2P file sharing systems, the popularity of objects follow long-tail distributions that most of the objects are rarely searched. Therefore, the SMA mode seems to be more suitable for those systems.

At last, Fig. 8 shows that the protocol can also adapt to the change of $m(t)$. We can find that there is no obvious difference among the three index dissemination schemes' performance. Although it seems that the protocol requires more time to converge, the long converge time does not im-

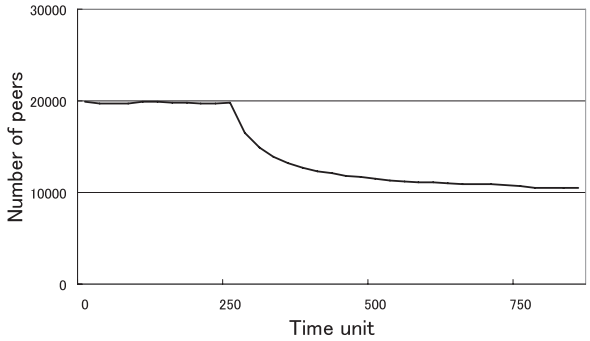
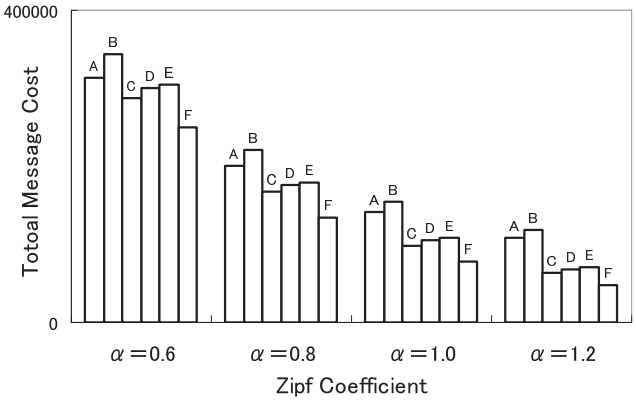


Fig. 9 The number of peers in the system. $m(t) = 400$ for $t \in [0, 250)$, $m(t) = 200$ for $t > 250$.



A: Fixed index number, theoretical minimum.
B: Fixed index number, name thread network.
C: Adaptive index number, theoretical minimum.
D: Adaptive index number, random network.
E: Adaptive index number, name thread network.
F: Adaptive index number, random-growing network.

Fig. 10 Simulation results in implementations of the random sampling service.

ply the protocol has bad adaptability in this case. That is because the system itself takes time to stable as shown in Fig. 9. Similar results can be seen by adopting dynamic lifetime distributions of peers.

5.2 Feasibility

Up to now, we discuss the problem under the assumption of the ideal random sampling service. However, to implement the ideal random sampling is very costly in distributed systems, e.g. each peer may have to know the whole peer set. Therefore, it is necessary to evaluate the protocol with non-ideal but cheap implementations. In this subsection, we show the performance of our protocol in realistic environments with feasible implementations of the random sampling service. We also compare the performance of our protocol with the protocols adopting fixed number of indices for every object to show the advantage of the popularity-biased index dissemination.

We adopt random walk to disseminate messages. A message (index or query) is carried by a random walker and then the peers on the trace of the random walker receive the message. Three kinds of overlay networks are adopted. All of those networks are directed and each peer keeps 30 outgoing links. The first network is the random network which is still an ideal setting but much easier to approach than the ideal random sampling [18]. In each time unit, the network is re-built in order to delete bad links pointing to the peers which have left the system. The second one is the *name thread* network which is proposed by the quorum-based search [17]. It approximately generates a random network by gossip-based link exchange among peers. The third one is the random-growing network [26]. The network converges to a power-law in-degree distribution which is an exceptional case.

The simulation parameters are almost the same as which used in Sect. 5.1. The only difference is that we use 1000 objects and each of them has a different search frequency. Each object x has a unique popularity rank, denoted by $r_x, 1 \le r_x \le 1000$. The search frequency f_x of object x is $f_x = 1000/r_x^{\alpha}, 0.6 \le \alpha \le 1.2$. That implies the objects' popularities follow the Zipf-distribution where α is the Zipf coefficient. The Zipf distribution and the scope of the Zipf coefficient are proved to be consistent to the objects' popularity distributions in P2P file sharing systems [7].

The simulation results are shown in Fig. 10. The three index dissemination schemes have almost the same result in this simulation because the objects' popularities are fixed and the most unpopular object is searched at least $1000/1000^{1.2} > 0.25$ times each time unit. The item A is the theoretical minimum message cost while each object has the same number of indices. Item B is the minimum cost of adopting fixed index number and random walk in the name thread network. Item C is the theoretical minimum message cost of our protocol which adopts adaptive index number. The items D, E, F are the results of our protocol in the random network, the name thread network and the random-growing network respectively.

Comparing A with C (or B with D, E), we can see that the popularity-based index dissemination can effectively decrease the system message cost, especially when the popularity distribution are highly skewed (the cases $\alpha = 1.0$ and $\alpha = 1.2$). Comparing C with items D, E, it can be found that the message cost of adopting nonideal random sampling implementations are at most 10% higher than that of the ideal system model. The result justifies that our protocol is practicable in real system environments since its performance is nearly optimal even adopting nonideal sampling service.

The item F is an exceptional case. Clearly, the results of F are much lower than those of C, D, E disregarding the value of α. In a power-law network, random walk samples high-degree node with high probability. This implies that indices and query messages is concentrated in a small part of peers. The system is out of the scope of the random sampling model and yields the problem of load balancing. Of course, the theoretical minimum message cost is not suitable for such non-uniform sampling services. Further investiga-

tions are left as the future work.

6. Supplemental Remarks

6.1 Message Size

We assumed that both the dissemination of an index and a query message equally cost one message. This assumption can be easily removed. Letting I_x and P_x be the communication cost of disseminating a index and a query message respectively, we obtain

$$M_x(t) = \frac{Q_x}{T} \cdot I_x + \frac{f_x n T}{Q_x \cdot \sum_{\tau=1}^{T} (1 - d(\tau))} \cdot P_x.$$

Thus, the optimal index number is

$$\hat{q_x} = \sqrt{\frac{f_x n P_x}{\sum_{\tau=1}^{T} (1 - d(\tau)) I_x}},$$

while the search cost is

$$s_x = \sqrt{\frac{f_x n I_x}{\sum_{\tau=1}^{T} (1 - d(\tau)) P_x}}.$$

Therefore, the Equal Rule still holds because the communication cost of an object is minimized when $q_x I_x = s_x P_x$, where $q_x I_x$ and $s_x P_x$ are the object's index maintenance cost and search cost respectively.

6.2 Queries for Inexistent Objects

When a peer search for some inexistent objects, the search process can not terminate because it continues searching until the object is found. To prevent the infinite search, a upper bound of search size, denoted by H, should be set. However, the bounded search size yields another problem that some rarely-searched objects are difficult to find because they have almost no indices. To increase the success rate of searching for those objects, we can introduce the minimum number of indices disseminated in each time unit, denoted by L. By Lemma 2.1 and Equality 4, the minimum success rate, denoted by ρ, for searching for any object is

$$\rho \geq 1 - e^{-HL \sum_{\tau=1}^{T} (1 - d(\tau))/n}.$$

The system message cost is still approximately the minimum because if L is small enough, the additional index maintenance cost is little.

7. Conclusions

In this paper, we investigated the index-dissemination-based search approaches under a general churn model that peers' lifetime distribution can be arbitrary. The objective is to minimized the system total communication cost including both the index maintenance cost and search cost. Under the uniform-random sampling assumption, the main theoretical contributions consists of a tight lower bound of the total system communication cost and two principles for optimal index dissemination which are natural but have never been proved under a general churn model. The first principle is the *Stream Method* that shows the best index dissemination method is to incrementally disseminates the same number of indices at each time unit. The method can stabilize the available index number in the system and minimize the expected search cost against the loss of indices when peers leave the system. The second principle is the *Equal Rule* that shows the optimal balance point of the trade-off between the search cost and index maintenance cost is to assign the same communication on the query and index dissemination.

According to the two principles, we proposed a fully distributed search protocol to achieve the optimal index dissemination adapting to the system environment. A remarkable advantage of the protocol is that the it yields almost no additional communication cost to achieve the self-adaptive feature. By simulation, we justify the protocol's effectiveness in both dynamic and realistic system environments.

Future work

This work is based on the uniform-random sampling model that peers disseminate queries and indices to any other peers with the same probability. As the simulation results shown in Fig. 10, the theoretical minimum communication cost is not suitable for the systems adopting non-uniform sampling such as in super-peer systems, e.g. the modern Gnutella network [27]. Although the *Stream Method* and the *Equal Rule* seem to be also optimal in those systems, the theoretical proof has not been done. We are looking forward to find some approaches for modeling those non-uniform sampling based systems under churn and complete the proof of the argument.

Acknowledgements

We specially appreciate the reviewers for their valuable comments which help us with the enhancement of the theoretical model and presentation.

This work is supported in part by MEXT: "Global COE (Centers of Excellence) Program", JSPS: Grant-in-Aid for Scientific Research ((B)19300017 and (B)17300020), MEXT: Grant-in-Aid for Scientific Research on Priority Areas (16092215), MEXT: Grand-in-Aid for Young Scientists ((B)18700059), MIC: Strategic Information, Communications R&D Promotion Programme (SCOPE), The Nakajima Foundation and The Ookawa Foundation Research Grant.

References

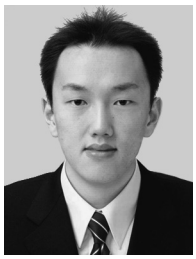
[1] D.S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, S. Rollins, and Z. Xu, "Peer-to-peer computing," Technical Report HPL-2002-57, HP Laboratories Palo Alto, March 2002.

[2] H. Balakrishnan, M.F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Looking up datas in p2p systems," Commun. ACM, vol.46, no.2, pp.43–48, Feb. 2003.

[3] E. Adar and B. Huberman, "Free riding on gnutella," First Monday, vol.5, Oct. 2000.

[4] I. Ari, B. Hong, E.L. Miller, S.A. Brandt, and D.D.E. Long, "Managing flash crowds on the internet," 11th IEEE/ACM International

Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2003), 2003.
[5] Napster website: http://www.napster.com
[6] M. Castro, M. Costa, and A. Rowstron, "Performance and dependability of structured peer-to-peer overlays," DSN '04: Proc. 2004 International Conference on Dependable Systems and Networks (DSN'04), p.9, Washington, DC, USA, 2004.
[7] Kunwadee Sripanidkulchai. The popularity of gnutella queries and its implications on scalability. http://www.cs.cmu.edu/~kunwadee/research/p2p/paper.html, 2001.
[8] S. Zhao, D. Stutzbach, and R. Rejaie, "Characterizing files in the modern gnutella network: A measurement study," Multimedia Computing and Networking 2006, vol.6071, no.1, p.60710M, 2006.
[9] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," ICS '02: Proc. 16th International Conference on Supercomputing, pp.84–95, New York, NY, USA, 2002.
[10] I. Stoica, R. Morris, and D. Karger, "Chord: A scalable peer-to-peer lookup service for internet application," Proc. SIGCOMM, pp.149–160, 2001.
[11] M.F. Kaashoek and D.R. Karger, "Koorde: A simple degree-optimal distributed hash table," Peer-to-Peer Systems II: Second International Workshop, IPTPS 2003 Berkeley, vol.2735/2003, pp.98–107, 2003.
[12] S. Ratnasamy, P. Francis, M. Handley, M.F. Karp, and S. Schenker, "A scalable content-addressable network," Proc. SIGCOMM, pp.161–172, 2001.
[13] Gnutella website: http://gnutella.wego.com
[14] Freenet website: http://freenet.sourceforge.net
[15] KaZaA website: http://www.kazaa.com
[16] Reference website: http://en.wikipedia.org/wiki/Winny
[17] K. Miura, T. Tagawa, and H. Kakugawa, "A quorum-based protocol for searching objects in peer-to-peer networks," IEEE Trans. Parall. Distrib. Syst., vol.17, no.1, pp.25–37, Jan. 2006.
[18] M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen, "The peer sampling service: Experimental evaluation of unstructured gossip-based implementations," Middleware '04: Proc. 5th ACM/IFIP/USENIX International Conference on Middleware, pp.79–98, New York, NY, USA, 2004.
[19] P. Mahlmann and C. Schindelhauer, "Peer-to-peer networks based on random transformations of connected regular undirected graphs," SPAA '05: Proc. Seventeenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, pp.155–164, New York, NY, USA, 2005.
[20] M. Zhong and K. Shen, "Random walk based node sampling in self-organizing networks," SIGOPS Oper. Syst. Rev., vol.40, no.3, pp.49–55, 2006.
[21] E. Cohen and S. Shenker, "Replication strategies in unstructured peer-to-peer networks," SIGCOMM Comput. Commun. Rev., vol.32, no.4, pp.177–190, 2002.
[22] D. Leonard, V. Rai, and D. Loguinov, "On lifetime-based node failure and stochastic resilience of decentralized peer-to-peer networks," SIGMETRICS Perform. Eval. Rev., vol.33, no.1, pp.26–37, 2005.
[23] Z. Yao, X. Wang, D. Leonard, and D. Loguinov, "On node isolation under churn in unstructured p2p networks with heavy-tailed lifetimes," INFOCOM 2007. 26th IEEE International Conference on Computer Communications., pp.2126–2134, May 2007.
[24] M. Mitzenmacher and E. Upfal, Probability and Computing, Camberidge, 2005.
[25] D. Stutzbach and R. Rejaie, "Characterizing today's gnutella topology," Technical Report CIS-TR-04-02, Dec. 2004.
[26] P.L. Krapivsky, G.J. Rodgers, and S. Redner, "Degree distributions of growing networks," Phys. Rev. Lett., vol.86, no.23, pp.5401–5404, June 2001.
[27] D. Stutzbach, R. Rejaie, and S. Sen, "Characterizing unstructured overlay topologies in modern p2p file-sharing systems," IMC'05: Proc. Internet Measurement Conference 2005 on Internet Measurement Conference, p.5, Berkeley, CA, USA, 2005.



Yu Wu received the B.E. degree in engineering in 2002 from Shanghai University and M.E. degree in computer science in 2006 from Osaka University. He is now a student of Graduate School of Information Science and Technology, Osaka University.

Taisuke Izumi received the M.E. and D.I. degrees in computer science from Osaka University in 2003 and 2006. He is now an Assistant Professor of Graduate School of Engineering, Nagoya Institute of Technology.

Fukuhito Ooshita received the M.E. and D.I. degrees in computer science from Osaka University in 2002 and 2006. Since 2003, he has been an Assistant Professor in the Graduate School of Information Science and Technology at Osaka University.

Hirotsugu Kakugawa received the B.E. degree in engineering in 1990 from Yamaguchi University, and the M.E. and D.E. degrees in information engineering in 1992, 1995 respectively from Hiroshima University. He is currently an associate professor of Osaka University.

Toshimitsu Masuzawa received the B.E., M.E. and D.E. degrees in computer science from Osaka University in 1982, 1984 and 1987. He had worked at Osaka University during 1987–1994, and was an associate professor of Graduate School of Information Science, Nara Institute of Science and Technology (NAIST) during 1994–2000. He is now a professor of Graduate School of Information Science and Technology, Osaka University.