Master's Thesis

Title

Replication Methods for Enhancing Search Performance in Peer-to-Peer Services

Supervisor

Prof. Masayuki Murata

Author

Yoshihiro Gotou

February 12th, 2003

Department of Informatics and Mathematical Science

Graduate School of Engineering Science

Osaka University

Master's Thesis

Replication Methods for Enhancing Search Performance

in Peer-to-Peer Services

Yoshihiro Gotou

Abstract

Many services have recently been offered based on a peer-to-peer (P2P) communication model. Peers connect to each other and build an overlaid logical network and available services are communicated over this network. The robustness of the P2P network against frequent peer failure must be considered. This includes when the peers leave the network and this directly affects the stability of the entire logical network. Replication of the content is one of the most useful techniques to increase robustness. However, the overall effectiveness of replication is heavily dependent on the topology of the logical network.

As topology of networks, including the Internet and P2P, follows a Power-Law distribution pattern, we first investigate the effect of the logical network topology (especially of the Power-Law characteristics) on replication methods. We use a search method called "n-walkers random walk" in which multiple queries move randomly across the P2P logical network. We use a "path replication method," to create replicas at all the intermediate nodes on the path between the requesting and responding nodes. Through simulations experiments, we observed that peers with a large number of degrees (e.g., $degree > 10$) make four times as many replicas as peers with a small number of degrees. In addition, replicas on large degree's peers are used ten times as frequently as those on peers with small degrees.

Based on these observations, we propose a query forwarding method that considers the Power-Law property of the network topology in order to improve the performance of the P2P service. In our method the queries are transmitted with different probabilities, dependending on the degree of each adjacent node. Our simulation results show that our proposed method can greatly improve the query performance by considering the characteristics of Power-Law. Our method reduces the average hop count in finding replicas by up to 60% compared with the random forwarding method.

Keywords

Peer to Peer (P2P)

replication

power-law distribution

logical network

degree based forwarding

Contents

List of Figures

List of Tables

1 Introduction

The Peer-to-Peer (P2P) model has recently attracted attention as a new network model. Unlike the server-client model, which is conventionally used today, P2P model does not need a special node (i.e., *a server*) to provide services. The nodes participating in the P2P service are called "peers," and these construct a logical community (we call it the *P2P community*). All the peers are treated equally in the P2P community. Sometimes a peer offers a service (i.e., acts as a *server* in the server-client model), and sometimes it receives a service (i.e., *a client*). The actual capability of the P2P community can be regarded as the aggregation of all the services provided by all the participating peers in the community. Unlike the server-client model, services are not centralized in a special node but distributed to the smaller peers. Moreover, since peers operate autonomously, the P2P model has advantages in terms of both scalability and robustness against network failure.

The P2P model is widely used nowadays. We summarize the services and projects that use P2P technology in Table 1. File sharing is a typical example of the P2P model. Each peer displays a list of shareable files. When another peer requests one of these listed files, the peer transfers the file directly to the peer that made the request. Consequently, a P2P community can be considered as a huge virtual storage area that provides numerous kinds of files. The word "P2P" came to be widely known after the appearance of *Napster*, released in 1998. Napster provided a place to exchange MP3 (MPEG-1 Audio Layer 3) audio files. Just by joining Napster, a user could get any famous artists' newest album for free. However, this caused a serious copyright problem and Napster had to stop its service in 2001.

Distributed computing is a form of collaboration, used to solve very complex problems (e.g., analysis of huge radio emissions to search for signs of extraterrestrial intelligence).

Table 1: P2P Applications

File sharing/transfer

Napster	http://www.napster.com/
gnutella	`http://gnutella.wego.com/`
KaZaA	`http://www.kazaa.com/`
Freenet	`http:/www.freenetproject.org`

Instant messanger

AIM (AOL Instant Messanger)	`http://www.aim.com/`
ICQ	`http://www.icq.com/`
Windows Messanger	`http://www.microsoft.com/windowsxp/`
Yahoo! Messanger	`http://messenger.yahoo.com/`

Distributed computing

SETI@home	`http://setiathome.ssl.berkeley.edu/`
distributed net	`http://www.distributed.net/`
United Devices	`http://www.ud.com/`

The problem is divided into many small *tasks*. Each peer processes a task and returns the result. The community can be regarded as a single, virtual, and high performance computer used to solve a complex problem. For example, 300 million of peers are capable of 25 TFLOPS, which matches for world's fast supercomputers.

The P2P based applications described above can be roughly classified into two types: One is the *hybrid model*, which uses a combination of P2P and server-client models. For example, in distributed computing, a control server is used to collect results from peers, manage peers' available resources, and plan the assignment of tasks to the peers. Some file sharing services, like Napster, also use an indexing server to search the peer that has the target file. In the hybrid model, a peer first connects to the control server to query the service. The peer then connects to the relevant peer that it can receive the service from. The other model is the *pure model* which uses no special server. Gnutella is one of the most well known services that uses the pure model. Peers in this model connect to each other and build an overlaid logical network. Services are given by communicating them over the logical network. However, unlike the hybrid model, there is no explicit way to find the relevant peer which provides the target service. The peer asks its neighbor peers for the target service. If none of the neighbor peers can provide the service the query is forwarded to their neighbors. The query is therefore advertised to all peers on a hop-by-hop basis.

The hybrid and pure models both have advantages and disadvantages. Since the control server in the hybrid model gives the address of the relevant peer directly, a delay in finding the peer depends on the response performance of the control server. On the other hand, in the pure model, the time to find the peer is not known. Moreover, the query may be lost due to a failure in a neighboring peer.

One of the disadvantages of the hybrid model is that at least one control server is needed to reply to the query. The community cannot work when the control server is down. Furthermore, the control server can easily become a bottleneck. The scale of the community (i.e., the number of peers in the community) will be limited by the capability of the control server. This weakness implies that only a limited number of users can create a community with the hybrid model. The community in the pure model, however, can still work even if many peers depart from the network.

In this thesis, we focus on the pure model, in which need to consider the following aspects.

- **Improvement of the search speed:**

  As mentioned above, the forwarding of the query in the pure model is performed on a hop-by-hop basis. The delay for searching the peers depends heavily on the method used for forwarding queries.

- **Robustness against the instability of the logical network:**

  Since peers in a P2P logical network are run by the application software, it is important to consider the robustness against frequent peer failures. This includes the peers that leave the logical network as this directly affects the stability of the logical network.

- **Continuously offering of service:**

  A stable service is desirable so that even if the peer which first offered a service leaves, the service can continue to be available.

Replication [1] was proposed as a solution to the above problems. Services are copied to some other peers. The peer with the copied service (called *replica*) can also offer the service. As the number of relevant peers has increased, the delay in finding the peers should be shortened. Moreover, although the peer which originally had the target service has left the community, the service can still be offered by other peers holding the replica. The details of this replication are described in the Section 3.

Replication has advantages, such as an improvement in handling peer failures, and the shortening of the search time. Replication can be applied to both hybrid and pure models. However, in the hybrid model, managing the replication is quite difficult because it requires many additional operations to the control server. In the pure model, on the other hand, a benefit of replication is large. This is why we focus on the pure model. Note that the pure model has a serious problem about its scalability [2]. However, the replication will be one solution to meet the scalability.

The effectiveness of the replication seems to be influenced by the topology of the community's logical network. Recent studies have shown that the Internet displays the characteristics of Power-Law [3] and Small-World [4]. The logical network of the P2P community also shows evidence of Power-Law behavior [5]. This property means that peers in the logical network are not connected randomly, but most are connected to a limited number of peers. These peers have a big influence on the stability of the logical network.

In this thesis, we first investigate the effects of the logical network topology (especially regarding Power-Law) on replication methods. Based on our observations, we next propose a new query forwarding method that considers the Power-Law property of the logical topology, in order to improve the performance of the P2P service. In our method, queries are transmitted with different probabilities, depending on the degree of each adjacent node. Through simulation experiments, we evaluate the effect of our forwarding method on the replication. Our numerical results show that our new forwarding methods can discover more requested services with the shorter delay.

This thesis is constituted as follows. First, in Section 2, we describe about the P2P search service that we focused on. Next, we explain the replication used in the P2P search service in Section 3. In Section 4, we describe the characteristics of the Power-Law and clarify the influence of Power-Law property in the replication method. We then propose a new replication method that considers the nature of Power-Law. A description of our proposed methods and their evaluations is shown in Section 5. Finally, we have Section 6, in which we conclude the thesis and outline future research topics.

2 P2P Search Service

We describe the search service in our pure P2P network. The service uses a distributed search method to find the relevant resource. Peers in the P2P community connect to each other, and this constitutes the logical overlaid network. A typical search process is as follows (see, e.g., [6]). When a peer requires a service it sends a *query* to its adjacent peers. This query message contains the name of the service (e.g., the file name), the address of the service-request peer, and the TTL (Time To Live) field. A peer that receives the query first checks whether it can offer the service itself. If it can, it sends the *reply* with its address back to the sender peer. Otherwise, it forwards the query to its adjacent peers. Before forwarding, the peer decreases the value of the TTL field by one, or if the TTL value is zero it discards the query. This mechanism is effective for avoiding a loop in forwarding queries. When the sender peer receives the reply message, it directly connects to the peer specified in the reply so it can receive the requested service. An example of the search service is shown in Figure 1.

Message forwarding in the pure model is performed on a hop-by-hop basis. The search delay depends on the method used for forwarding queries. There are several methods that can be used [1]. The simple approach is the peer that receives the query passes it to another peer picked randomly from among the adjacent peers. This is called the **random walk** approach. A peer asks for the service from all the peers belonging to the community one at a time. The weakness of this approach is that it takes a lot of time to wait for the replies. For example, suppose the number of peers in the community is c, the expected number of peers that forward the query is $c/2$, or c in the worst case scenario. This approach is not suitable for a large community.

n**-walkers random walk** is one of the derived versions from the **random walk** approach. In n**-walkers random walk**, n of queries are sent simultaneously by the

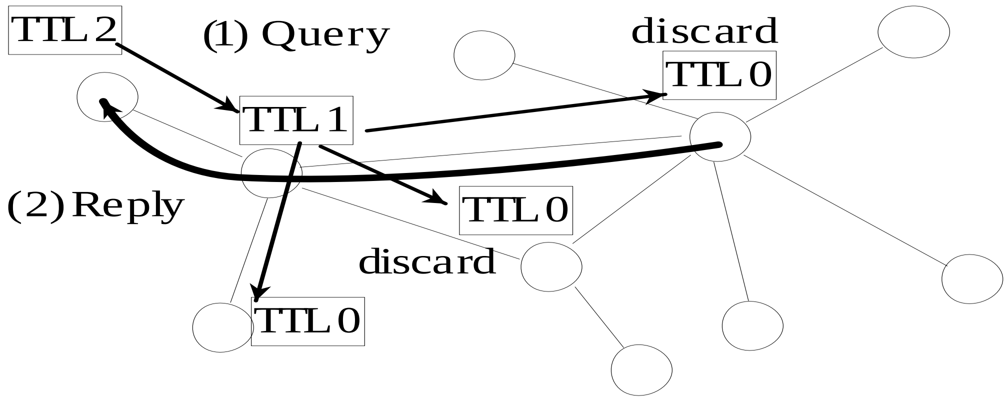

Figure 1: Example of P2P Search Service

sender peer. A peer receiving a query simply forwards to another peer, which is the same action as in the **random walk**. n**-walkers random walk** has a clear tradeoff between the value of n and the delay in searching: When the value of n increases, the expected number of peers forwarding the query decreases linearly. On the other hand, a large value of n demands a large volume of message traffic, which decreases the possible usage of the link bandwidth. The authors in [1] investigated the effectiveness of n**-walkers random walk**, and showed that between 16 and 64 queries is reasonable as the value of n irrespective of the network size.

Query flooding is another possible approach in which the peer receiving the query forwards (i.e., broadcasts) it to all adjacent peers. This approach is the fastest way of finding the relevant peer. However, **query flooding** has a strong disadvantage in terms of scalability. As the number of forwarding peers increases, the amount of consumption traffic increases exponentially. As a result, all of the link bandwidth is wasted by control messages. This is why the early versions of Gnutella could not be widely used [2].

We used the 16-walkers random walk approach in our experimental evaluations.

3 Replication Methods used in P2P Services

3.1 Types of Replication

Replication is an effective technique for decreasing delays in searching and in improving the stability of available services. A service is copied (called *replica*) to other peers in the community. The key question to ask is “how do we distribute the replicas?” In [1], three replication methods are proposed. See Figure 2 for an overview of these replication methods.

Owner Replication

In the owner replication method, the peer which received the service keeps a copy so it can offer the service itself if requested by other peers in the future. In other words, the receiver peer also becomes a service provider. The number of replicas will increase in proportion to the number of requests for the service. Nevertheless, it is insufficient for fully improve performance.

Random Replication

In random replication, replicas are randomly distributed amongst other peers. If we use random forwarding n**-walkers random walk**, **Random Replication** is the most effective approach for achieving both smaller search delays and smaller deviations in searches. However, to perform random replication, the peer must know the information of all the peers in the logical network. This is not easy to implement, because a peer can only know information about its adjacent peers (i.e., not all the peers in the logical network).

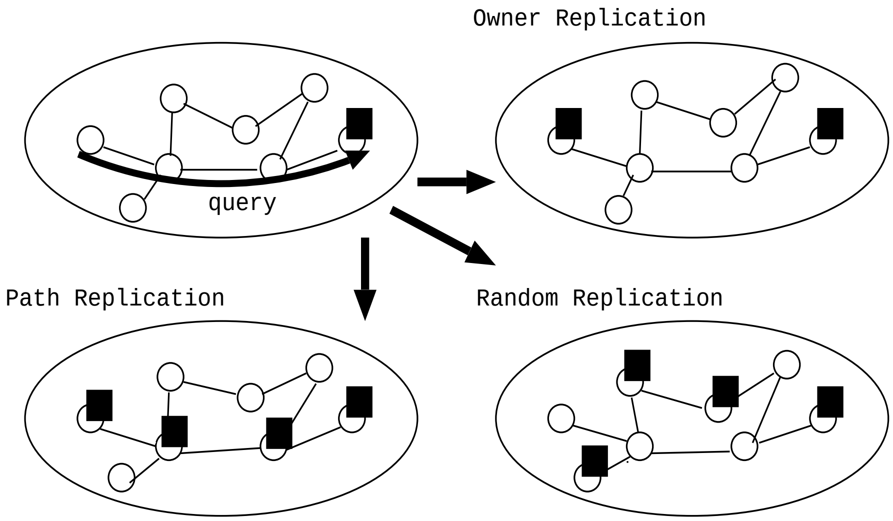

Figure 2: Examples of Replication

Path Replication

Path replication is another approach for distributing multiple replicas for each service. As the peers forward the query they record their address into the message. The service-providing peer receives the query which contains information about the sequence of peers (i.e., forwarding route) that forwarded the message. The provider peer can then send a reply and replica of the service in the reverse direction of the forwarding route. Simulation results have shown that path replication achieves a similar performance to random replication [1], while implementation is less complex than the random replication. For this reason, we use path replication as the replication method in this thesis.

3.2 Effect of Replication

We show some simulation results and the effectiveness of the replication. The simulation conditions used are shown in Table 2. We used a random network consisting 10,000 peers

Table 2: Simulation Conditions to Evaluate the Effect of the Replication

parameter	value
network topology	random
number of peers	10,000
number of links	20,000
number of items	100
search method	16-random walk
TTL	500

and 20,000 links, generated by a topology generatorfor the underlying network topology. The number of target items was set to 100. A 16-walkers random walk was used for the search method. The value of the TTL field in the query was initially set to 500. We defined a "search hit" as when the query arrived at the service-provider peer until the query was discarded (i.e., the TTL value is larger than 0). We varied the number of replicas from 0 to 20. For performance metrics, we introduced a *hit ratio*, which was calculated from the number of search hits divided by the number of queries. We also investigated the distribution of the forwarding peers.

First, we show the robustness against peer failures with replication. Figure 3 shows the relation between the hit ratio and the number of replicas. The lines represent the number of replicas for 1, 5, 10, 15, and 20 for each item, After distributing the replicas by random replication, we removed a certain ratio of peers randomly before starting the simulation. The horizontal axis of the figure is the ratio of randomly removed peers. By changing the ratio of removed peers, we can plot the hit ratio. If the replication is not performed, the hit ratio falls down to around 10% when the 30% of peers have left the logical network. On the other hand, if we place five replicas for each service, we cannot observe any degradation

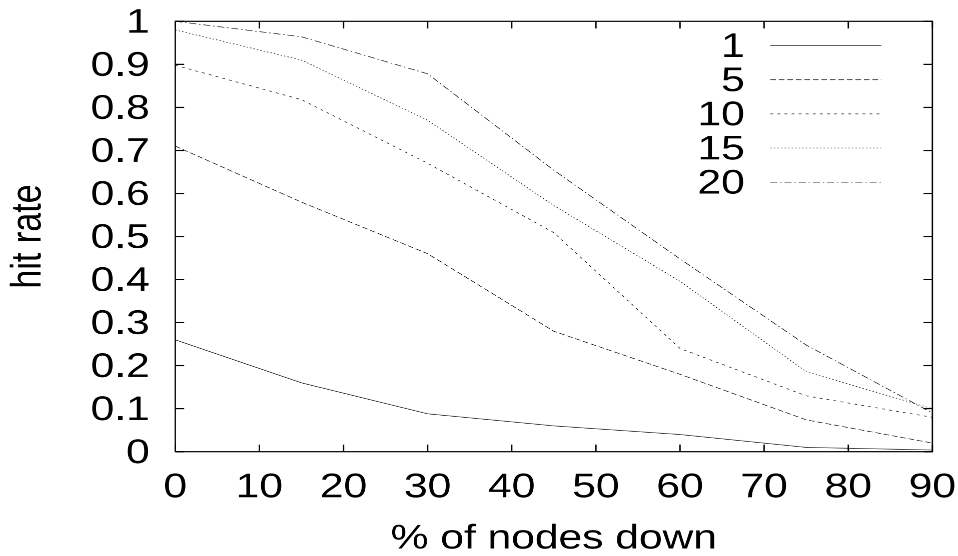

Figure 3: Search Hit Ratio vs. Peer-Removing Rate

in the hit ratio. The distribution of the hop count (the number of peers which forwards the query message) to the service-providing peer is shown in Figure 4. This shows that the hop count is significantly decreased by creating replicas. Figure 5 plots the relation between the number of replicas and the hit ratio. The hit ratio can also be improved by increasing the number of replicas.

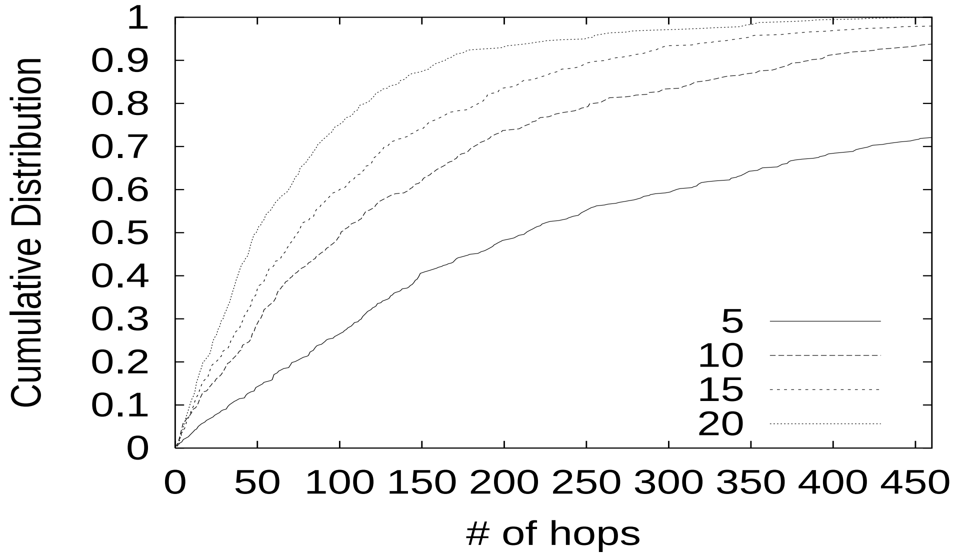

Figure 4: Distribution of Hop Count

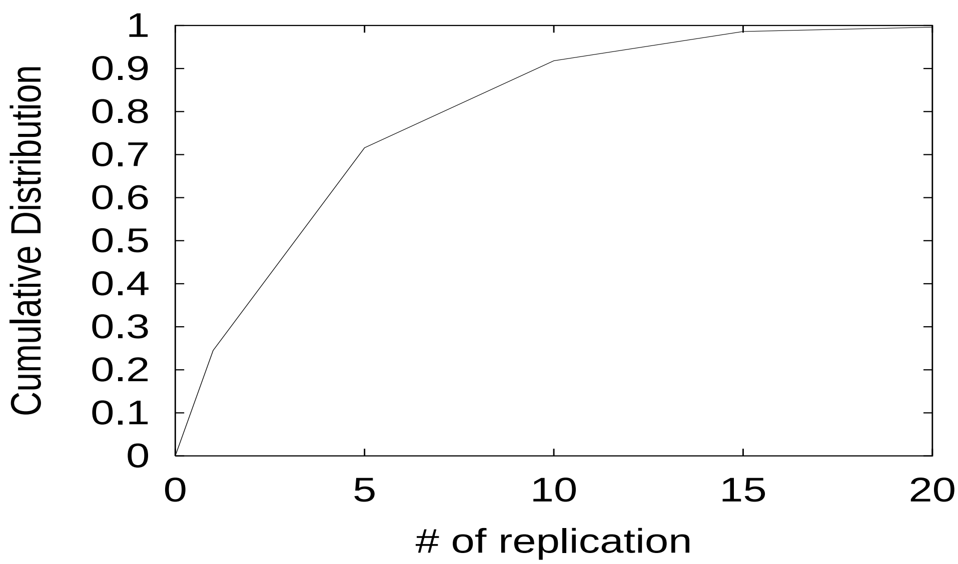

Figure 5: Effect of Number of Replication on the Hit Ratio

4 Effect of Power-Law Property of Logical P2P Topology on Replication

4.1 Outline of Power-Law Property

Recent studies have shown that the Internet has the characteristic of Power-Law [3] and the logical network of P2P community also follows the nature of Power-Law [5]. This property means that peers in the logical network are not connected randomly, but most of peers are connected to limited number of peers. Such peers make a big influence on the stability of the logical network.

Suppose the node i is on the network. We define the number of adjacent nodes of the node i as "degree of i," which is denoted by x_i. If x_i follows the function $f(x)$ defined by Eq. (1), x_i satisfies the Power-Law property.

$$f(x) = x^{\alpha} \tag{1}$$

The effect of the Power-Law property to the communication quality have been studied in some literatures (e.g.,[7, 8]). In the Power-Law network, a few nodes have a large number of degrees while most other nodes have only a small number of degrees. For example, our observation have shown that 5% of nodes have high degrees larger than 10, and 50% of nodes only have a single degree. Therefore, when we randomly select a node in the Power-Law and random network respectively, the degree of the node in the Power-Law network is almostly smaller than the one in the random network. That is, the impact of a node failure on the Power-Law network is smaller than the impact on the random network in most cases [7]. However, if the node with high degree fails, the Power-Law logical network becomes easily unstable. Intentional attacks to the nodes with high degrees will cause a serious damage to the logical network [8].

In respect of the query performance, the Power-Law network has a tendency to follow

Table 3: Simulation Conditions to Evaluate the Effect of the Power-Law Property on Replication

parameter	value
Network topology	Power-Law (BA model)
Number of peers	10,000
Number of links	20,000
Number of items	100
Search method	16-random walk
TTL	500

character of the Small-World [4], in which the message can be forwarded to much more nodes with small number of hops by sending nodes with higher degrees. By using this phenomenon, the query message can be advertised to other peers faster by sending the query message to the peers having higher degrees aggressively.

Although the influence of the Power-Law property to the query forwarding have been studied so far, the influence to the replication is not clear yet. In [1] authors have only investigated the effects of the replication in the random network topology, i.e., the effects in the Power-Law network topology have not been shown. Howerver, the similar aspects shown in the query forwardings may be expected in the replication.

4.2 Influence of Power-Law Property to Replication

For above reason, we first clarify the influence of the Power-Law to the replication methods in this section. Simulation conditions used in section are shown in Table 3. Parameters are used with the same values as in the previous subsection (see Table 2) except the type of network topology. We use the Power-Law topology consisting 10,000 peers and

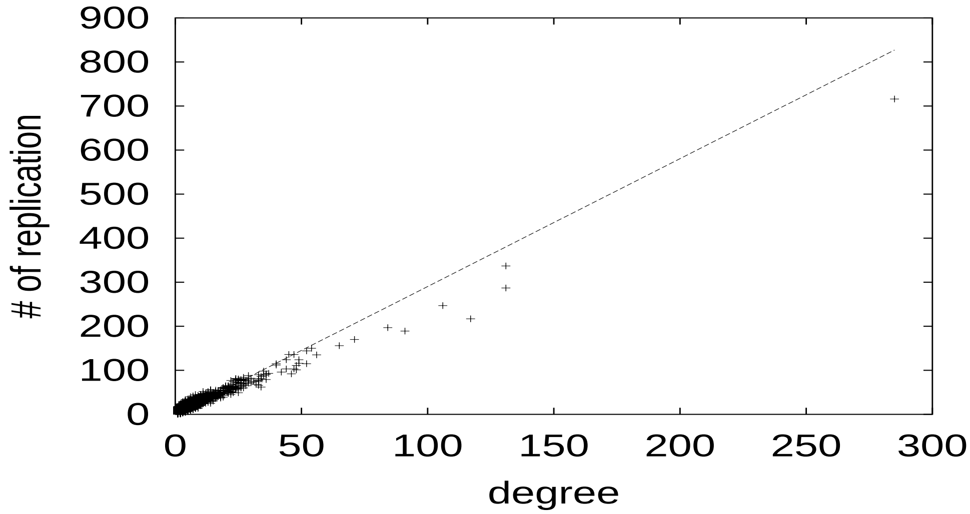

Figure 6: Relation between Degree and Number of Replica Creations (Power-Law)

20,000 links generated by the BRITE [9] topology generator using BA (Barabasi-Albert) model [10]. The number of target services is set to 100, and 16 random walk is used for the search method. The value of TTL field is set to 500. Of course, the appropriate value of TTL should be existed on this simulation condition. Choosing the appropriate TTL value is our future research topic.

4.2.1 Distribution of Replicas

First, we show the effect of the Power-Law property on the number of replicas created on peers. We repeated 50,000 searches in the simulation. the Power-Law logical network shown in Table 3. We do not limit the number of replica for each peer (Henceforth, it is called as “replica capacity”) now. Figure 6 show relation between degree and the number of replica creations. There is a strong correlation in the relation between degree and the number of replica creations (Correlation-coefficient: 0.953) . Figure 7 shows the number of replica creations also shows the character of the Power-Law. This means the possibility that the high degree peer is on a search course becomes high, and replicas are replaced

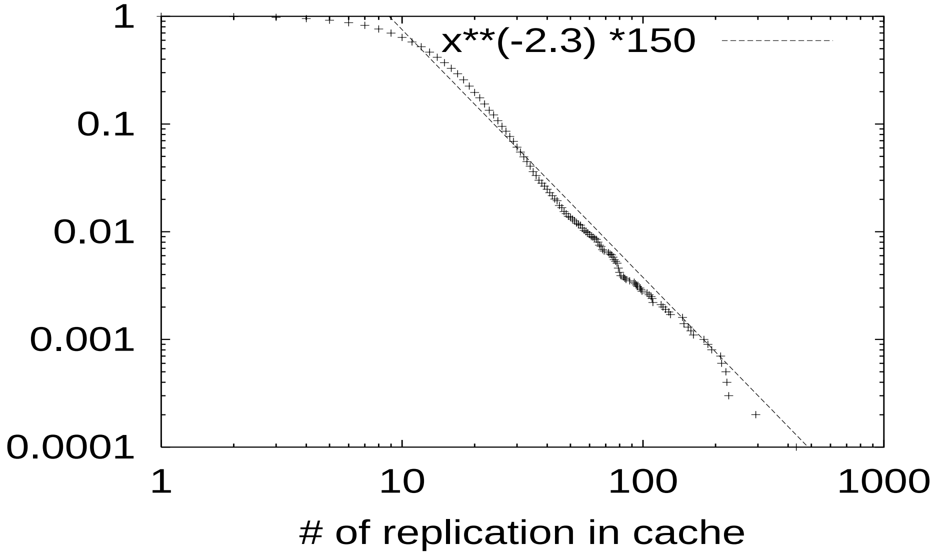

Figure 7: Distribution of Replicas

frequently.

4.2.2 Access Frequencies of Replicas

Because the peer having the more degrees receives the more query messages, the frequencies of access of replicas may differ according to the number of degrees of the peer owning the replica.

Here, we investigate the effect of degrees on the frequency of access of replicas. We classify all peers into two groups: *high group* in where peers have many degrees ($x_i > 10$), and *low group* in small number of degrees ($x_i \leq 10$). Table 4 summarizes the statistics of two groups. In Table 4, the second row is the number of peers classified for each group. The third and fourth rows are total and average number of replicas owned by peers respectively. The fifth and sixth rows are total and average number of times accessed replicas. The last row is what percentage of all accesses is made by peers belonging to the group. As shown in this table, replicas on peers in the *high group* are accessed more than 20 times as large as those of *low group* peers, while the number of replicas is less than four times. The

Table 4: Differences in the Replica Use Frequency between Degree Groups

Groups	*low group* (≤ 10)	*high group* (> 10)
Number of peers	9497	503
Total number of replicas	128494	25507
Average	13.53	50.71
Total number of accesses	23932	26090
Average	2.52	51.87
Access ratio (%)	4.6	95.3

most interesting result is that only 5% of all peers reply 95% of all queries. That is, in the Power-Law network, most of peers (95% in this result) could not provide the capability of replication efficiently. We consider that in the good use of such inefficient replicas the effective of the replication can be much improved.

4.2.3 Effects of the replica capacity

In the above, we assume that peers have the infinite replica capacity. Here, the *replica capacity* is how many replicas can be stored on the peer. However, in actual, the replica capacity directly is associated with the size of the storage which the peer can offer, the replica capacity is finite value. We then show the effect of the replica capacity to the performance of the replication. For introducing the replica capacity, we need to consider the policy for replacement of replica in case of the occupancy of the replica capacity. We use the Least Recently Used (LRU) policy for the replica replacement.

Figure 8 shows the relation between the replica capacity and the hit ratio where the replica capacity is varied from 0 to 5. In this figure, we can observe that the hit ratio in the random network topology has a small advantage rather than in case of the Power-Law

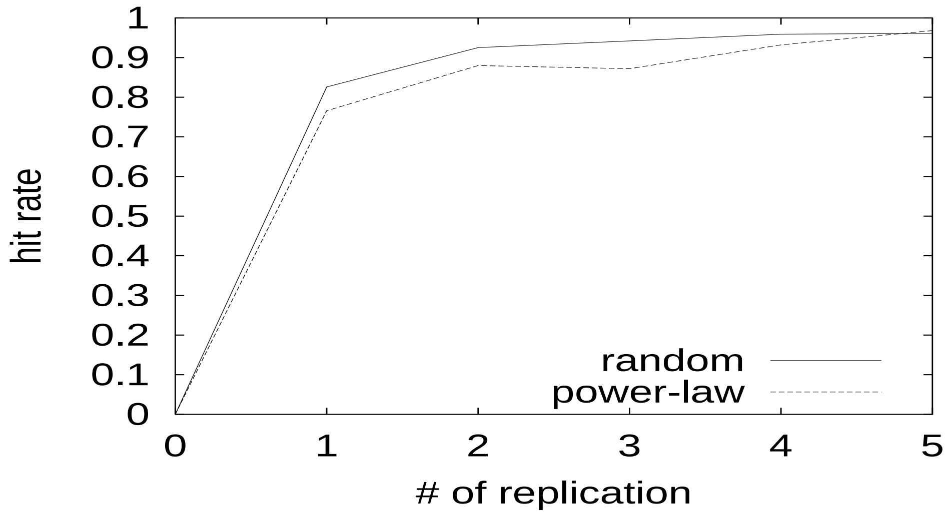

Figure 8: Effect of Replica Capacity to Hit Ratio

network. As described before, the access frequencies of replicas are biased in the Power-Law network (i.e., query messages consentrate in peers having large degrees). As a result, replicas on the peers in large degrees are replaced frequently. The peer sometimes fails to reply the query. On the other hand, in the random network topology, the frequencies are well distributed and the replacement of replicas is less frequent than in case of the Power-Law network.

We next show relation between the replica capacity and the average number of hops required to find the peer. Figures 9 and 10 show the time dependent variation of the average number of hops. In both cases, the required number of peers becomes decreased by increasing the replica capacity. By comparing the Power-Law and the random networks, the Power-Law property makes more effect in the replica capacity. It is because the replacements of replicas on peers are directly decreased by the increase of the replica capacity. However, in peers having small degrees, the impact of increasing the replica capacity is quite limited, and most of replicas have bad use efficiency. Therefore, we need to take the efficient use of replicas on peers having small degrees into consideration.

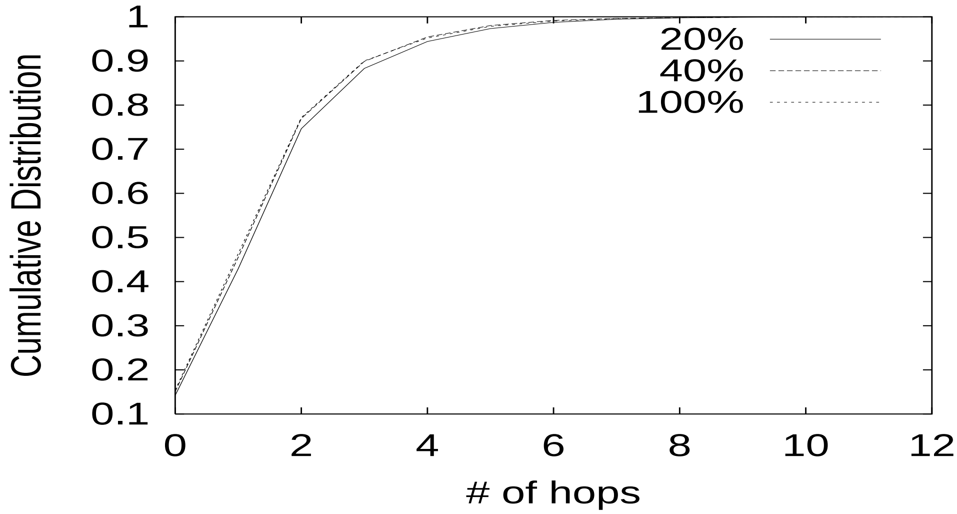

Figure 9: Effect of Replica Capacity to Average Hop Counts (Random Network)

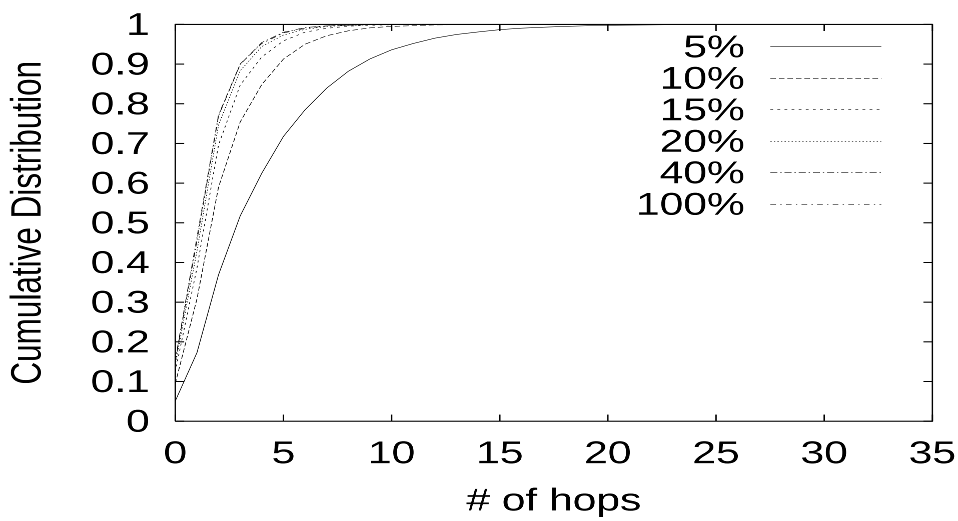

Figure 10: Effect of Replica Capacity to Average Hop Counts (Power-Law Network)

5 Query Forwarding Method

As we stated before, the importance of peers strongly depends on their degrees. So we should consider the replication methods taking the effect of degrees into consideration. In this section, we propose query message forwarding methods according to the number of degrees of adjacent peers. In the path replication used in this thesis, replicas are created along the search path on which the query message is passed. The effects of query forwarding method appear to both the search performance and the utilization of replicas. We consider an efficient replication with managing query transmission method.

In n-walkers random walk used in the previous section, the peer chose the next hop peer from its adjacent peers. On the other hand, in the proposed methods, the next hop is chosen based on the probability delivered from the degrees of adjacent peers. In our methods, we assume each peer can retrieve the degrees of adjacent peers. We call adjacent peers $n_1, \ldots, n_d$, and their degrees $d_{n_1}, \ldots, d_{n_d}$.

5.1 Degree based Query Forwarding Methods

The original n-walkers random walk uses the random selection policy for choosing the next hop peer when there are multiple adjacent peers connected with the peer. Results shown in the previous section have illustrated such random policy is not much efficient in terms of both the query speed and the utilization of replicas. From this reason, we propose new query forwarding methods to improve the above-mentioned performance by taking the degree of peers into consideration. Note here that we refer *Random* as the original n-walkers random walk.

5.1.1 HDF (High Degree Forwarding)

As shown in Section 4, peers having large degrees make more replicas than small degree peers. If we intentionally choose the larger degree peers for forwarding the query message, the search delay is expected to be shorten. For this purpose, we define a new forwarding method called HDF (High Degree Forwarding). In HDF, the next hop peer is chosen by the probability which is weighted according to the degree of peers. The probability of choosing the next peer p_H^i is given by

$$p_H^i = \frac{f_H(d_i)}{\sum f_H(d_k)}, \tag{2}$$

where $f_H(d_i)$ is the function of the degree of peer i (d_i). In this thesis, we use the simple function which is increased in proportional to the value of d_i, i.e., $f_H(d_i) = d_i$

5.1.2 LDF (Low Degree Forwarding)

Results in the previous section have also shown that replicas on peers with smaller degrees have less reference frequencies. Conversely, when the reference frequency of the peer having a large degree becomes higher, the load of the peer is much heavier. This phenomenon implies that the P2P community becomes serious damaged, if the peer with large degree fails from community. To solve this problem, the frequency of references are entirely distributed by all peers joining the community. In this thesis, we propose the LDF (Low Degree Forwarding) method for the query forwarding. In the LDF, peers with smaller degrees are intended for choosing as the next hop peer among adjacent peers. More specifically, the peer i chooses the next hop peer by the probability p_L^i, which is the reciprocal proportional function of the degree d_i. That is,

$$p_L^i = \frac{(1/f_L(d_i))}{\sum(1/f_L(d_k))}, \tag{3}$$

where $f_L(d_i)$ is same as $f_H(d_i)$, i.e., $f_L(d_i) = d_i$ in this thesis.

5.1.3 Extensions for Improving Performance in Forwarding Methods

While HDF and LDF have advantages, they also have some weak points. For example, in HDF, peers tend to be forward query messages to peers with large degrees. However, due to the Power-Law property, the number of peers with large degree is quite limited. If we simply apply HDF to the query forwarding, the peer continuously tries to forward the query message to already visited peers with large degrees many times. As a result, the peer becomes hard to receive the reply message when the degree of peer having the target service is small. There are some approaches to solve this problem: 1) Revisit Avoidance, 2) Weighted Function, and 3) Hybrid Forwarding. We describe detailed processes below.

All forwarding methods including HDF and LDF do not use the route of the message (i.e., what peers are traversed previously). Namely, the message may be passed to the already visited peer two or more times. Forwarding to the visited peer simply increases the search delay and wastes the network resource (e.g., link bandwidth). We introduce the simple mechanism to avoid revisits for peers as follows.

- Before forwarding the query message, the peer records the pair of (*identifier of the query message*, *address of the destination peer*).
- When the peer receives the query message, the peer obtains the set of adjacent peers to which the message is not sent yet by seeking above records.
- If there is a peer to be sent (i.e., the set is not empty), the peer forwards the message to the appropriate peer.
- Otherwise (i.e., the set is empty), the peer discards the query message.

Note here that the revisit avoidance mechanism requires both computational and storage overheads to maintain the revisit records. Such overhead cannot be negligible when the number of peers and/or the number of query messages increase.

As noticed before, when the number of forwarded peers increases, HDF or LDF serves as an opposite effect rather. In such case the random forwarding is better than HDF and LDF. As solutions for this problem we consider two approaches. One is to use the weighted function which shifts the probability p_i from the value of HDF/LDF to the value of random according to the increase of the number of visited peers. More specifically, the transmission probability $p^i_W(h)$ is the function of the number of visited peers h, which is defined as

$$p^i_W(h) = \beta(h)p^i + (1-\beta(h))\frac{1}{D_i}. \tag{4}$$

Where p^i is p^i_H in HDF and p^i_L in LDF. D_i is the number of adjacent peers of the peer i. Here, $\beta(h)$ is the shifting function which varies from 1 to 0 according to the increase of h. In this thesis, we apply the linear function, i.e.,

$$\beta(h) = \begin{cases} 1-h/b & (1-h/b>0) \\ 0 & (x \leq 0) \end{cases}, \tag{5}$$

where b is a parameter which defines the point β becomes zero.

In above forwarding methods, n query messages are sent from the receiver peer. All query messages are forwarded by the same forwarding methods. However, since each forwarding method has both strength and weakness, the combination of forwarding methods has the possibility to compensate their weakness each other. In the Hybrid Forwarding, we divide n query messages into some groups, and apply the different forwarding method for each group.

5.2 Simulation Results

In this subsection, we compare our proposed forwarding methods with the n-walkers random walk, and show the effects of forwarding methods.

Table 5: Simulation Conditions

parameter	value
Network topology	Power-Law (BA model)
Number of peers	10,000
Number of links	20,000
Number of items	1,000
Distribution of Request	Uniform or Zipf's Law
TTL	50

5.2.1 Simulation Conditions

Simulation conditions are shown in Table 5. For extension of the weighted function and the hybrid forwarding, we evaluate the weighted HDF and the hybrid HDF respectively. In the hybrid HDF, we apply HDF to the half of query messages (8 messages), and the rest are applied to Random. In our simulation, we assume that the frequency of request for each service is uniformly distributed (i.e., all services are requested equally). However, actually, the popularity of each service is not equal. Some measurement results (e.g., [11]) have shown that the distribution of requests of items in gnutella follows the Zipf's Law. Zipf's Law [12] is a deviation that request for i th popularity of item occurs by the frequency proportional to $1/i^{\alpha}$. The influence of Zipf's Law nature is shown in Subsection 5.2.5.

5.2.2 Effect of HDF

Figure 11 compares distributions of hop counts required for search among HDF, LDF, and Random. This figure shows that HDF can decrease the hop counts. This is because HDF can reach high degree peers which has many replicas in short hops with nature of its forwarding method. On the other hand, the number of peers in LDF is larger than the

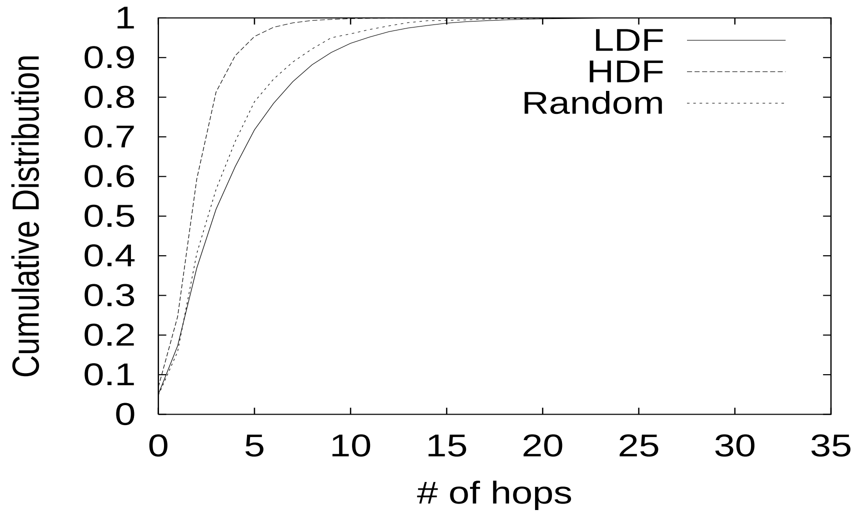

Figure 11: Distribution of the Number of Forwarded Peers (HDF, LDF, Random)

one in Random. This is because the LDF does not aim at time to search but uniformizing of load between peers. In path replication, Large number of hops leads to large number of replicas, LDF takes large number of hops to achieve many replicas.

5.2.3 Effect of Load Balancing by LDF

Figure 12 compares the time variation of total number of replicas among HDF, LDF, and Random. The number of replicas in LDF is increasing rapidly (about twice faster than in HDF). LDF tends to forward the query messages to peers with smaller degrees which are less utilized in HDF and Random. As a result, replicas are widely distributed.

Figure 13 shows the complementary cumulative distribution of numbers of replicas created on peers. As shown in this figure, the variation of the number of replicas in LDF is smaller than that of Random and HDF. Hence, the deviation of replica creation load is suppressed by using LDF. Although the utilization of the replica capacity can be normalized by using LDF, its performance (i.e., search delay and ratio of search hits) is not so high. We consider that the objective of LDF is not to increase the search performance,

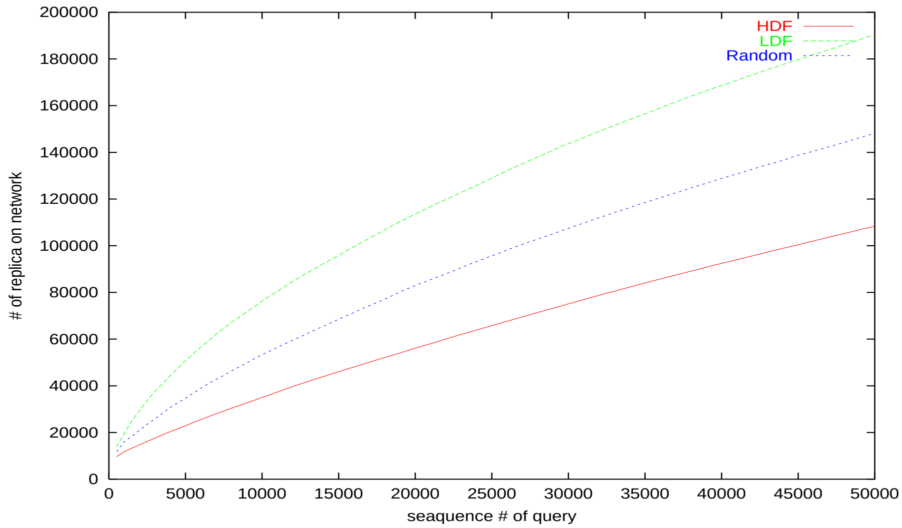

Figure 12: Time Variation Observations of Number of Replicas (HDF, LDF, Random)

but improve the stability of the community (i.e., logical network) entirely. We need to evaluate the scenario where peers join and/or leave the logical network frequently as our future research topic.

5.2.4 Effect of Revisit Avoidance Mechanism

Figures 15 and 14 compare the number of visited peers and the ratio of search hits among HDF and Random with/without revisit avoidance, respectively. By introducing the revisit avoidance, we can observe the performance improvement in both number of visited peers and hit ratio. Not only the implivement of HDF, Improvement of Random is large in Hop Count. This method has positive effect for each topology.

5.2.5 Influence of Deviation of Requests

We next show influence of the popularity of requests. Figures 16 and 17 show the results of the simulation. From these figures, on the influence of the variation of request, HDF is higher than Random. The ratio of improvement of HDF ($|Random - HDF|/Random$) is

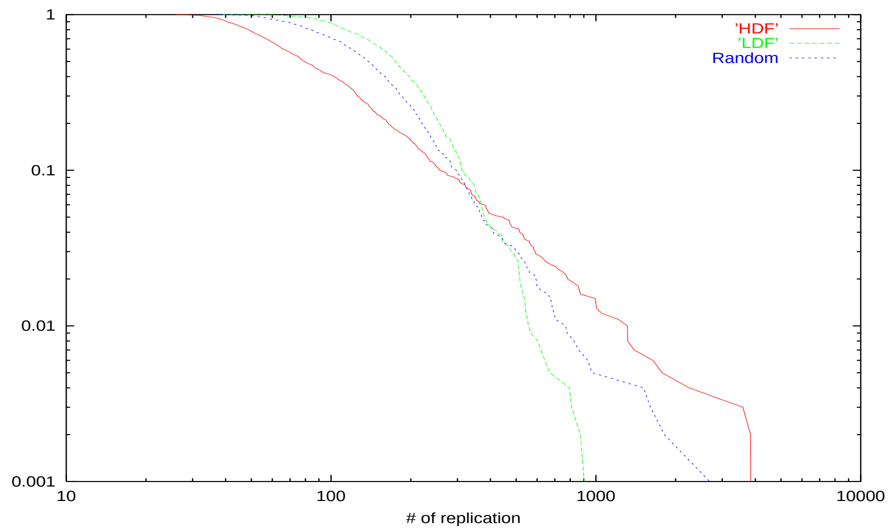

Figure 13: Complementary Cumulative Distribution of the Number of Replica Creations (HDF, LDF, Random)

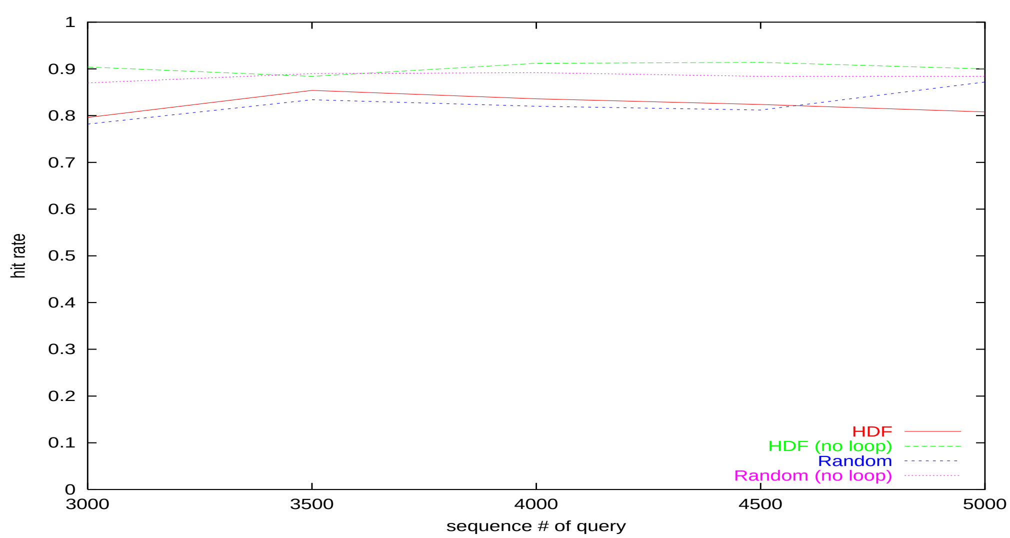

Figure 14: Improvement of Average Hit Ratio by Revisit Avoidance

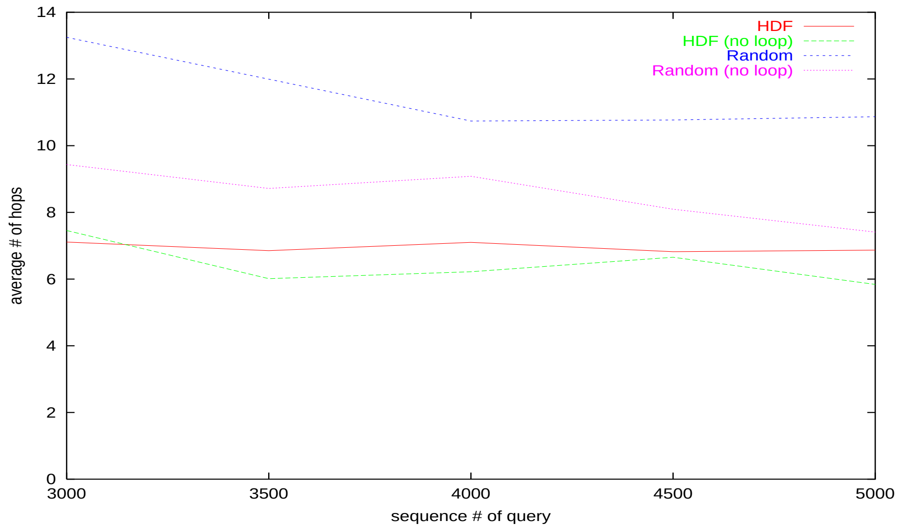

Figure 15: Improvement of Average Hop Counts by Revisit Avoidance

4.7% in uniform, and 18.6% in Zipf's Law. We consider that the more popular service has a tendency to be placed on the peer with the larger degree. Since HDF tries to forward to the peer with larger degree, the influence of Zip's Law property is thus higher in HDF.

5.2.6 Effects of Weighted HDF and Hybrid HDF

Figure 18, 19 are the result of simulation. In each figure, the horizontal axis represents forwarding method, the left end is Random Forwarding and the right end is HDF. Weighted HDF and Hybrid queries are shown between them. Weighted HDF and Hybrid queries show performance in both hit rate and hop count. We can see at x-intercepts of β equal 25, average hop count decrease (Figure 19) but hit rate doesn't decrease (Figure 18). This is because there is Random Forwarding query enough to keep hit rate, and some HDF query enhancing performance. This can see good combination about this parameter.

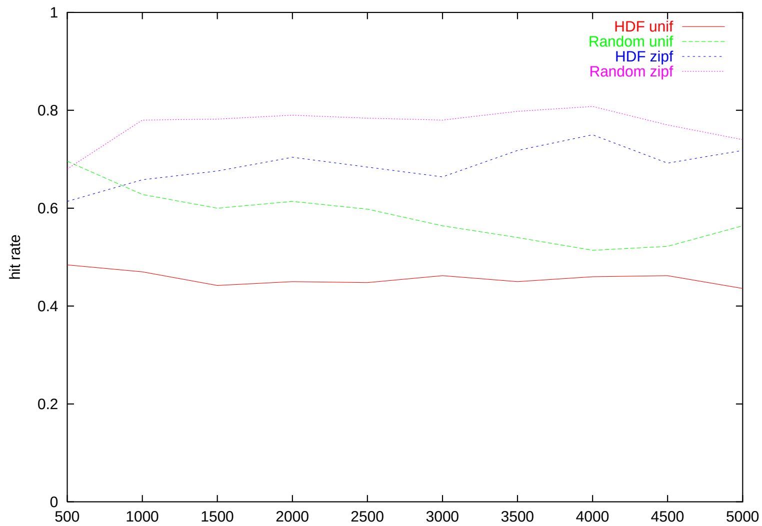

Figure 16: Influence of Requests following Zipf's Law Distribution (Hit Ratio)

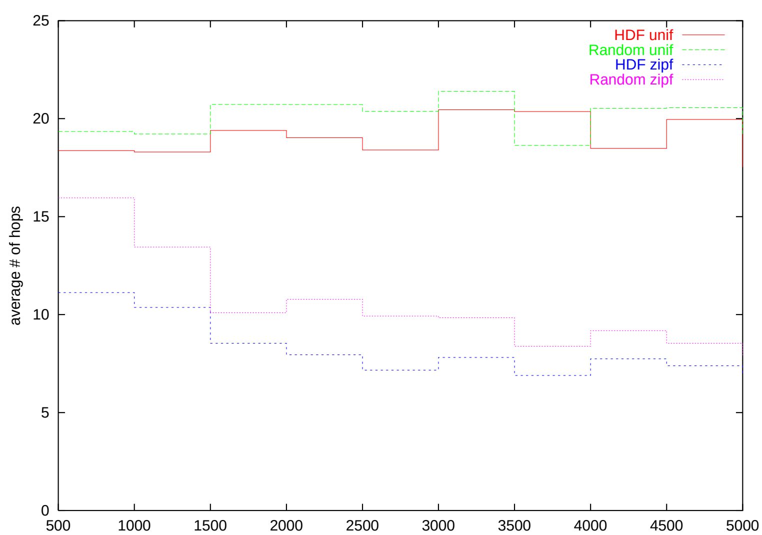

Figure 17: Influence of Requests following Zipf's Law Distribution (Hop Count)

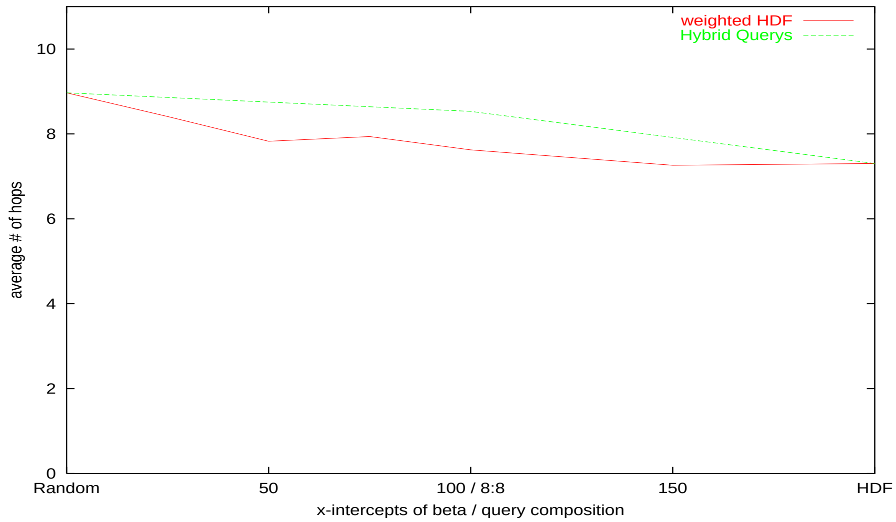

Figure 18: Effect of Weighted HDF and Hybrid Forwarding (Hit Ratio)

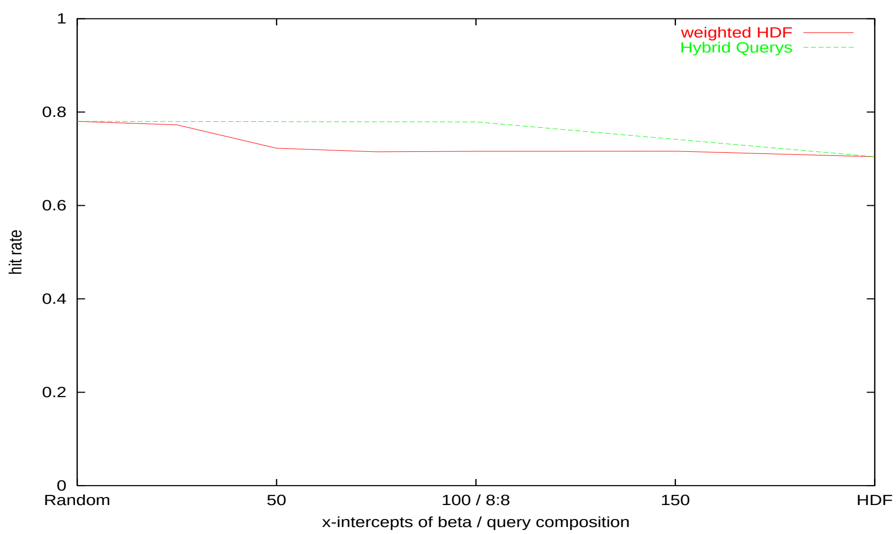

Figure 19: Effect of Weighted HDF and Hybrid Forwarding (Hop Count)

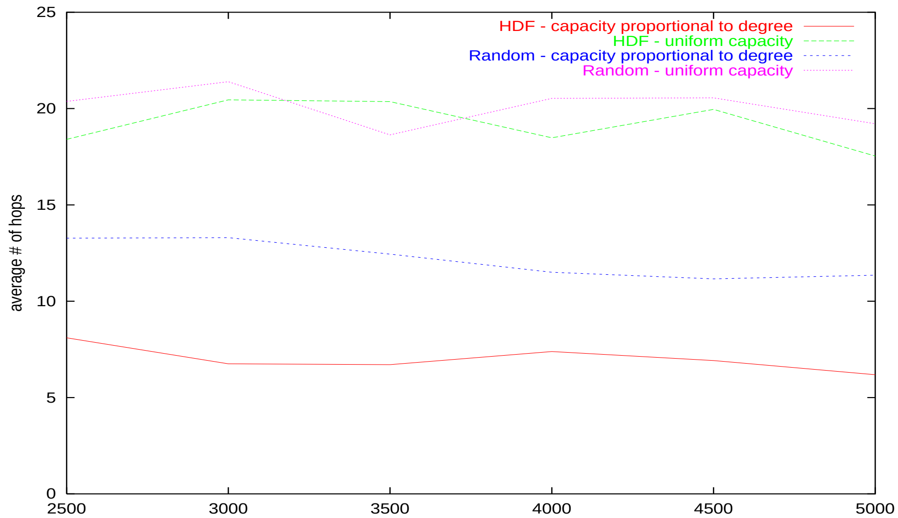

Figure 20: Effect of Linear Proportional Replica Capacity Adjustment (Hop Count)

5.3 Adjusting the Replica Capacity based on Degree

As shown in the previous subsection, the performance improvement in HDF is due to concentrated accesses at peers having large degrees. However, excessive concentration leads the performance degradation because of the frequent replacement of replicas. For this problem, adjusting the replica capacity according to the degree of peer seem to be effective. In this subsection, we evaluate the effect of adjusting replica capacity. Note that we don't change the total of replica capacity for the fair comparison.

Figures 20 and 21 show the results. In this figures, we set the replica capacity in proportional to the degree of peers. By setting the replica capacity dynamically, the average number of forwarded peers is reduced significantly by using HDF. This is simply because the probability of replacement of replicas becomes decreased by introducing additional capacity for replicas. However, in this result, the replica capacity on the largest degree peer is 245 times as large as the lowest degree's one. Then, we also evaluate the case where replica capacity is proportional to the sqare root of degree. This method can be expected to make the deviation of replica capacities smaller. Figure 22 and 23 are results

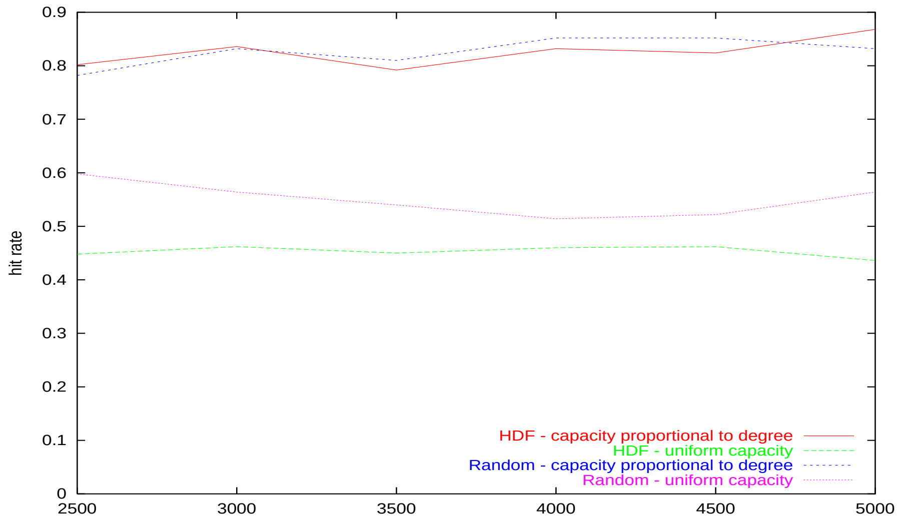

Figure 21: Effect of Proportional Replica Capacity Adjustment (Hit Ratio)

of the square root case. Although the performance improvement is decreased compared with the linear case, we can still observe a good result.

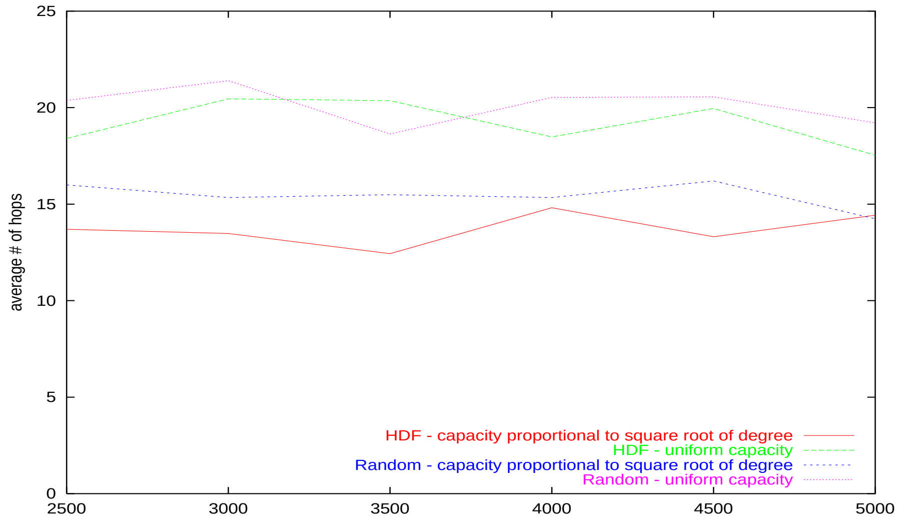

Figure 22: Effect of Square-Root Proportional Replica Capacity Adjustment (Hop Count)

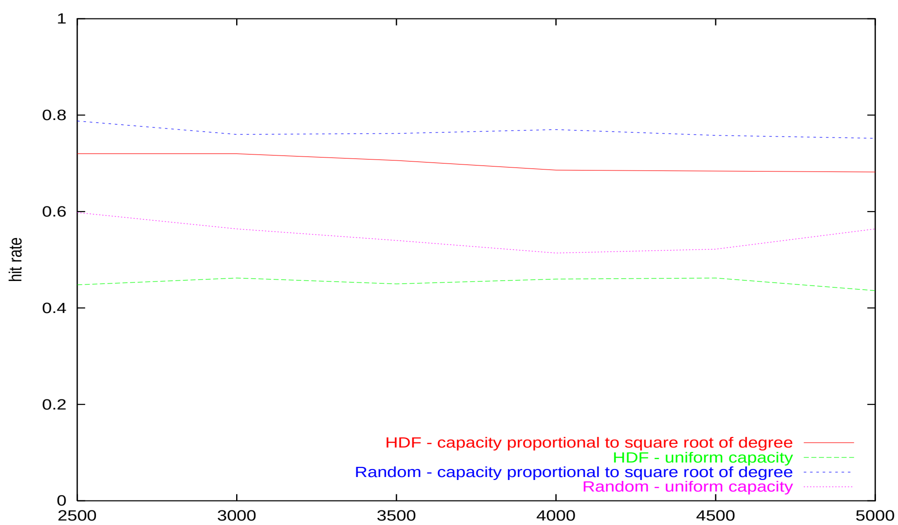

Figure 23: Effect of Square-Root Proportional Replica Capacity Adjustment (Hit Ratio)

6 Conclusion

In this thesis, we have focused on the effect of Power-Law property on the replication in the P2P services.

We first clarified the influence of Power-Law property in replication methods. In the Power-Law network, a few nodes have a large number of degrees while most other nodes have only a small number of degrees. Our observation have shown that 5% of peers have high degrees larger than 10, and 50% of peers only have a single degree. We have also shown that these 5% of peers reply 95% of all queries. We have found that the characteristic of network topology strongly affects the effect of replication (e.g., search speed, utilization of replicas).

Based on above observations, we next have proposed new query forwarding methods which takes degrees of nodes into considerations. We have proposed HDF (High Degree Forwarding) and LDF (Low Degree Forwarding) for the query forwarding methods. From our simulation results, we have observed that HDF can decrease the search delay instead of the degradation of the hit ratio. If we simply apply HDF to the query forwarding, the peer continuously tries to forward the query message to already visited peers with large degrees many times, and the peer becomes hard to receive the reply message when the degree of peer having the target service is small. To solve this problem, we have proposed some extentions (revisit avoidance, weighted function, and hybrid forwarding) to apply HDF and LDF. We have also shown that our extentions can solve the problem in HDF achieving the performance improvement of HDF.

There are some future research topics listed below. 1) We need to evaluate the effect of LDF in more detail. 2) Choosing the appropriate value of TTL is also important to obtain the maximum improvement in the P2P search. 3) Defining the weighted function of HDF and the combination of hybrid forwarding are also important for further improvement of replication.

Acknowledgements

I would like to express my sincere and deep gratitude to my supervisor, Professor Masayuki Murata of Osaka University, who has guided me through this work. This thesis owes much to his thoughtful and continuous advice and I appreciate greatly.

Special thanks go to Research Associate Shingo Ata of Osaka City University, for his valuable comments and suggestions. The accomplishment of this thesis cannot be without his many supports.

I would like to express my gratitude to Professor Hideo Miyahara, Associate Professor Naoki Wakamiya, Associate Professor Hiroyuki Ohsaki, Associate Professor Go Hasegawa, and Research Associate Shin'ichi Arakawa of Osaka University, for their comments and suggestions on the research activities.

Finally, I express my gratitude to a number of friends and colleagues in the colleagues in the Department of Informatics and Mathematical Science, and the Department of Information Science and Technology of Osaka University for their generous help, enlightening and valuable suggestions.

References

[1] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proceedings of 16th ACM International Conference on Supercomputing (ICS'02)*, June 2002.

[2] J. Ritter, "Why gnutella can't scale. no, really.." `http://www.darkridge.com/~jpr5/doc/gnutella.html`.

[3] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[4] J. Kleinberg, "The small-world phenomenon: An algorithm perspective," in *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pp. 163–170, ACM Press, 2000.

[5] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, "Search in power-law networks," *Physical Review E, 64 46135*, 2001.

[6] "The gnutella protocol specification v0.4." `http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf`.

[7] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "A measurement study of peer-to-peer file sharing systems," Tech. Rep. UW-CSE-01-06-02, Department of Computer Science and Engineering University of Washington, 2001.

[8] P. Keyani, B. Larson, and M. Senthil, "Peer pressure: Distributed recovery from attacks in peer-to-peer systems," in *Proceedings of The IFIP Workshop on Peer-to-Peer Computing*, 2002.

[9] "Brite: Boston university representative internet topology generator." `http://www.cs.bu.edu/brite/`.

[10] R. Albert, A. Barabsi, and H. Jeong, "Mean-field theory for scale-free random networks," *Physica A 272*, pp. 173 – 187, 1999.

[11] K. Sripanidkulchai, "The popularity of gnutella queries and its implications on scalability.." `http://www-2.cs.cmu.edu/~kunwadee/research/p2p/gnutella.html`.

[12] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *Proceedings of the IEEE INFOCOM '99*, Mar. 1999.