Master's Thesis

Title

Design and Evaluation of a Cooperative Mechanism among Pure P2P File-sharing Networks

Supervisor
Professor Masayuki Murata

Author
Junjiro Konishi

February 15th, 2006

Department of Information Networking
Graduate School of Information Science and Technology
Osaka University

Master's Thesis

Design and Evaluation of a Cooperative Mechanism among Pure P2P File-sharing Networks

Junjiro Konishi

Abstract

To provide application-oriented network services, a variety of overlay networks are deployed over physical IP networks. Since they share and compete for the same physical network resources, their selfish behaviors affect each other and, as a result, their performance deteriorates. Our research group considers a model of overlay network symbiosis, where overlay networks coexist and cooperate to improve their application-level quality of service (QoS) while sustaining influences from the physical network and other overlay networks. In this thesis, we especially focus on Peer-to-Peer (P2P) networks among various overlay networks. We propose a mechanism for pure P2P networks of file-sharing applications to cooperate with each other. In our proposal, cooperative peers establish logical links among two or more P2P networks, so that messages and files are exchanged among cooperative P2P networks through these logical links. For an efficient and effective cooperation, we also propose an algorithm for the selection of cooperative peers and a caching mechanism to avoid putting too much load on cooperative peers and cooperating networks. Simulation results showed that the number of discovered providing peers and the ratio of search hits increased about twice, while the load by the cooperation among P2P networks was reduced about half by caching.

Keywords

Overlay Network
Cooperative Network
Peer-to-Peer (P2P)
File-sharing

Contents

List of Figures

List of Tables

1 Introduction

To provide application-oriented network services, various overlay networks are deployed over physical IP networks. Each overlay network independently measures network conditions such as the available bandwidth and latency through active or passive measurement schemes. Based on its observations, each overlay network controls traffic, chooses routes, and changes topologies in a selfish manner to satisfy its own application-level QoS. Since overlay networks share and compete for the same physical network resources, their selfish behaviors affect each other and their performance deteriorates [1, 2]. For example, to communicate faster with other nodes, a node measures bandwidth and latency to other nodes and changes its neighborship accordingly. As a result, the load in the physical network dynamically changes and consequently the quality of communication perceived by other overlay networks which compete for the same links and routers in the physical networks deteriorates. Those affected overlay networks then adapt data rate, routes, and topologies to satisfy or improve their application-level QoS. This further affects other overlay networks and it causes frequent changes of routing and introduces congestions in the physical network. Finally, the selfish behavior of overlay networks trying to improve their application-level QoS in fact results in the deterioration of application-level QoS.

Recently there are several publications on cooperative overlay networks to enhance their collective performance and efficiently utilize network resources [3, 4, 5, 6, 7, 8]. In [3, 4], the authors investigated a spectrum of the cooperation among competing overlay networks. For example, they proposed the Synergy overlay internetwork which improved routing performance in terms of delay, throughput, and loss of packets by cooperative forwarding of flows. In [5], mechanisms of inter-overlay communications are proposed to exchange information among overlay networks without knowing the destination addresses by using an overlay network called i3 (Internet Indirection Infrastructure) network. The i3 network is a network architecture consisted of some servers. In the i3 network, a receiver sends *trigger* messages with a service identifier and receiver's address to the i3 network. A sender sends *packet* messages with a service identifier to the i3 network. The i3 network transfers *packet* messages to receivers whose *trigger* messages have the same or similar service identifier.

Our research group considers the symbiosis among competing overlay networks [9, 10]. In the model of symbiotic overlay networks, overlay networks in a system evolve, interact with each other, and dynamically change internal structures. Overlay networks meet and communicate with each other in a probabilistic way. Overlay networks that benefit from each other reinforce their relationship, eventually having many inter-overlay links, and merging into one overlay network. Otherwise, they separate from each other. All evolutions, interactions, and internal changes are performed in a self-organizing way. Each node independently decides its behavior based on locally available information. Symbiosis among overlay networks emerges as a consequence of independent and autonomous behaviors of nodes and networks.

For this purpose, we need mechanisms for overlay networks to communicate with each other. In this thesis, we propose a mechanism for an efficient and effective cooperation among P2P networks of file-sharing applications. In a P2P network, hosts called peers directly communicate with each other and exchange information without the mediation of servers. According to user's intention, peers behave on its own decision as an individual does in a group or society. One typical example of P2P applications is a file-sharing system. Napster and WinMX are categorized as hybrid P2P networks where there are so-called meta-servers to maintain meta-information, e.g., list of providing peers to help peers to discover files. In the case of the cooperation among hybrid P2P networks, it is an architectural problem that meta-servers must deal with a large amount of messages since peers always try to obtain meta-information from meta-servers. Other members in our research group proposed a mechanism for the cooperation among hybrid P2P networks and investigated the influence of system conditions such as the number of peers and the number of meta-information in [10]. On the other hand, Gnutella and Winny are pure P2P networks without a server for searching files. A peer has to discover the desired file by itself by emitting a search message into the network. Other peers in the network response to the search message with a response message and relay the search message to their neighbor peers (Fig. 1).

The cooperation among pure P2P networks is accomplished by exchanges of search and response messages among them through logical connections established among so-called cooperative peers. With such cooperation, we can expect that search messages are disseminated more effectively and peers discover desired files more efficiently. Since a peer receives more response messages for a desired file, it can choose a more appropriate peer, i.e., faster and more reliable, from many candidate peers, leading to a higher application-level QoS. Even if P2P networks share different types or categories of files, employ different protocols, or have different architectures, there are benefits in the cooperation. For example, as in [3, 4], the cooperation in routing mes-

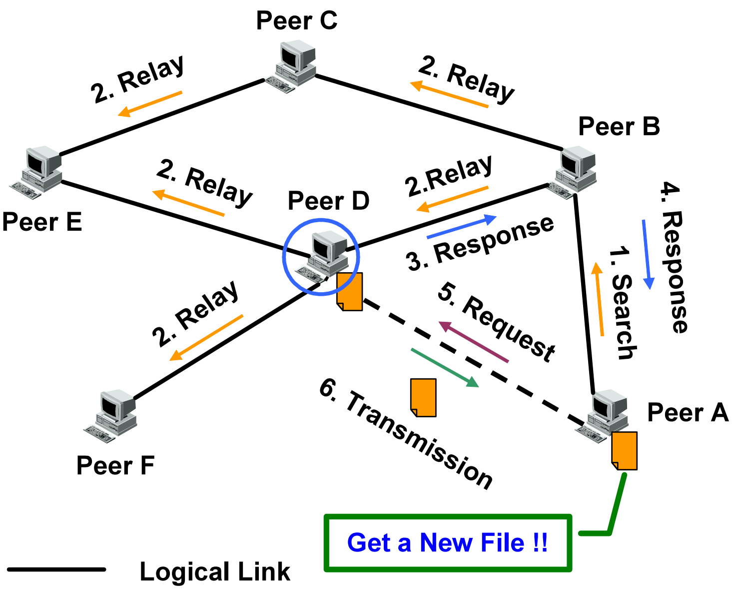


Figure 1: Flooding over a pure P2P file-sharing network

sages provides faster and more reliable transmission of messages over P2P networks. Furthermore, when a P2P network is disconnected by failures or disappearance of peers, search and response messages can propagate among separated parts of the P2P network through cooperative P2P networks. Therefore, the robustness and the resilience of P2P networks are improved by the cooperation as verified in [9].

However, to accomplish the efficient and effective cooperation without introducing much load on logical networks, some careful considerations must be made. For example, if a cooperative peer is located at the edge of a P2P network, it has to set a large Time to Live (TTL) value for search messages to spread over the network. As a result, the number of duplicated search messages to discard increases. They waste network bandwidth and cause network congestions. Therefore, we propose an algorithm to choose appropriate cooperative peers. Cooperative peers are selected from high-degree peers to disseminate search messages effectively. Furthermore, they are placed apart from each other in a P2P network to avoid the concentration of load. We also give some considerations on incentives that a selfish peer begins the cooperation. To reduce the load by the cooperation among P2P networks, we propose a caching mechanism for cooperative peers. We evaluate our proposed mechanism by simulation experiments.

The rest of this thesis is organized as follows. In Section 2, we propose a mechanism for the cooperation among pure P2P networks of file-sharing applications. In Section 3, we evaluate our proposed mechanism through several simulation experiments in terms of the number of discovered providing peers, the search latency, the ratio of cache-hits, and the load on peers. Finally, we conclude the thesis and describe future works in Section 4.

2 Cooperative Mechanism among Pure P2P File-sharing Networks

In this section, we propose a mechanism for pure P2P networks of file-sharing applications to cooperate with each other in an efficient and effective way. In the cooperation among pure P2P networks, a logical link is first established between designated peers, called cooperative peers, which are selected from candidate peers in each P2P network. Candidate peers are those which are willing to play the role for the cooperation to enhance and improve their own QoS. And then, search and response messages are transmitted through the logical link between cooperative peers (Fig. 2).

Our proposed mechanism consists of the following steps. First, a peer in a P2P network is promoted to a candidate peer by running a cooperative program. It joins a candidate network constituting by candidate peers to exchange information for the selection of cooperative peers. Next, a tentative cooperative peer is selected in candidate peers, and then it confirms whether it is appropriate as a cooperative peer or not. After the confirmation, a tentative cooperative peer is promoted to a cooperative peer. Then, a cooperative peer discovers a cooperative peer in another overlay network. When both cooperative peer consider that the cooperation will benefit to themselves, a logical link is established between the cooperative peers. If the link is accepted by the both sides, a cooperative peer finally begins to exchange search and response messages with the cooperative peer at the other end of the logical link. When either end of the logical link considers that it is useless to maintain the link, it is disconnected. We describe in the following the selection of cooperative peers, the preparation before the cooperation of P2P networks, and the behavior of peers in cooperative P2P networks in detail.

2.1 Joining a Candidate Network

When a peer is not satisfied with an application-level QoS received from a P2P network of file-sharing application, it considers to enhance and improve its application-level QoS by its own decision. For example, when a peer cannot discover a desired file at all, when a peer cannot discover enough number of providing peers, or when a peer cannot tolerate the delay in retrieving a file from a providing peer, a peer, i.e., a user should have some frustrations. The peer will consider that it can receive the higher QoS by connecting to another P2P network which provides it with the higher probability of successful search, the larger number of providing peers, and the

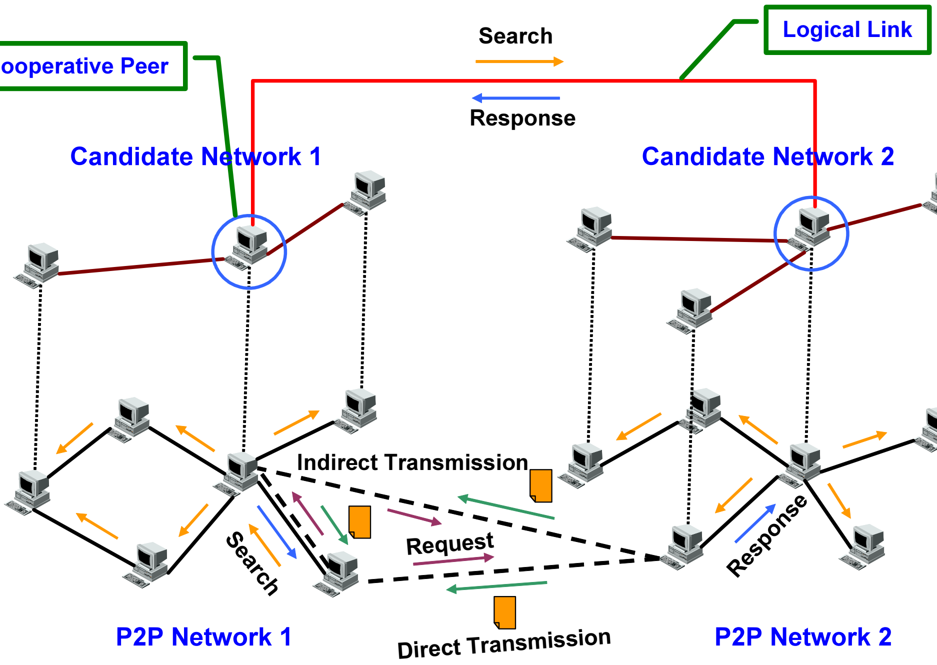


Figure 2: Cooperation among pure P2P file-sharing networks

smaller delay in file retrieval. In such a case, intending to improve its application-level QoS, the peer introduces the cooperation program independently of others. It implies that the peer does not care whether the other peers in the same P2P network will benefit from the cooperation or not. Then, it becomes a candidate peer, i.e., a candidate for cooperative peers. As illustrated in Fig. 2, candidate peers in a P2P network construct a candidate network to communicate with each other to select cooperative peers.

A new candidate peer first discovers another candidate peer in the same P2P network by flooding a special message over the P2P network or using the i3 network [5]. In the latter case, a new candidate peer registers itself to an i3 service repository by sending a *trigger* message containing a service identifier and its address (Fig. 3). Candidate peers in a candidate network send *packet* messages containing a service identifier and its address to the i3 network periodically. A new candidate peer receives one of their *packet* messages and establishes a logical link to the candidate peer. After that, the new candidate peer deletes its *trigger* message from the i3 service repository.

For this purpose, candidate peers must have a similar service identifier in the same P2P network but different from those of other P2P networks. We consider that a service identifier consists

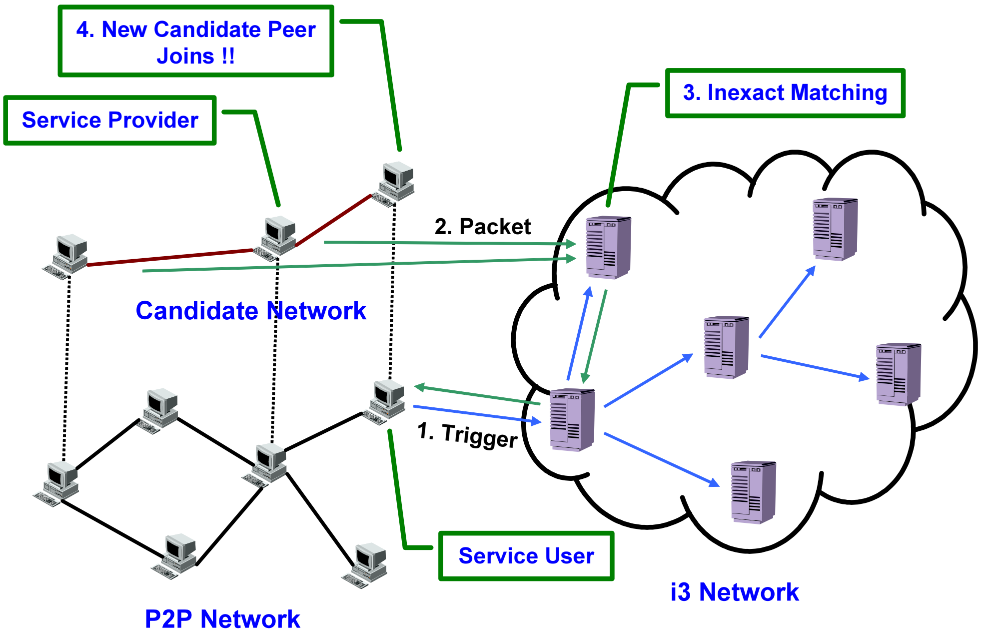


Figure 3: A new candidate peer joins the candidate network

of $l + m + n = 256$ bits. The first l bits are for the cooperation service and common among all cooperation programs. The following m bits correspond to the P2P network. To have the same m bits among candidate peers in the P2P network, we use the IP address of a bootstrapping node. To join a P2P network, a new peer first contacts a bootstrapping node, which should always be available online, to obtain one or more IP addresses of other peers. Since peers in a P2P network know the same bootstrapping node, by applying a hash function to the IP address of the boot strapping node, all candidate peers can have the same network identifier of m bits. We should note here that there is a small possibility that two or more P2P networks have the same m bits identifier. However, we consider that we can avoid the problem without introducing any mediation mechanism. Peers in a P2P network tend to exist close to each other due to a service discovery mechanism of pure P2P applications. Since the i3 network forwards a *packet* message to a node which registers a matching *trigger* message and is close to the sender of the *packet* message, we can expect that a *packet* message is forwarded to another candidate peer of the same P2P network. The last n bits are generated at random. In the i3 network, *inexact matching* is used where the *packet* message has a service identifier matching the longest pattern of bits with the *trigger* message. Therefore, a new candidate peer discovers a randomly chosen candidate peer in the same P2P network.

2.2 Selecting Cooperative Peers based on Degree and Distance

Cooperative peers are selected from the candidate peers on receiving a cooperation request. A new cooperation request is generated by a newly joined candidate peer, generated by a candidate peer on its own decision, or sent from other P2P network.

Cooperative peers must be carefully selected to effectively disseminate search messages in P2P networks and distribute the load on peers and networks. It is shown in recent studies, e.g., [11] that the Internet and many overlay networks have a power-law topology whose degree distribution follows $p(k) \propto k^{-\alpha}$. In [12], it is shown that peers can discover files effectively through high-degree peers. It means that by choosing peers with a large number of neighbor peers as cooperative peers, we can expect effective message dissemination. However, high-degree peers are closely connected with each other and thus such selection leads to the concentration of load and causes congestions.

For the efficient and effective message dissemination, we select cooperative peers that have higher degree and are apart from each other. Details of our proposed selection method of cooperative peers are as follows. First, every candidate peer advertises its degree, i.e., the number of neighbor peers, by flooding a message over a candidate network. Based on obtained information about other candidate peers, each peer ranks candidate peers in descending order of degree. Then, a candidate peer which ranks itself highest advertises a candidacy message to all other candidate peers over a candidate network to become a tentative cooperative peer. On receiving a candidacy message, other candidate peers check the rank of the tentative cooperative peer in their ranking list. If it is not on the first in the list, a candidate peer sends a conflict message to the tentative cooperative peer. A tentative cooperative peer gives up its candidacy and removes itself from the list on receiving more conflict messages than a predetermined threshold T. The threshold T is introduced to consider the case that a candidate peer, who accidentally missed an advertisement of a tentative cooperative peer, will send a conflict message. Otherwise, a tentative cooperative peer floods a confirmation message with a TTL value of k in a P2P network. If any cooperative peer already exists within the range, it sends a reject message to the tentative cooperative peer. On receiving a reject message, a tentative cooperative peer gives up its candidacy and advertises its cancellation to the other candidate peers. The tentative cooperative peer is removed from the list and another selection is conducted again. By this mechanism, cooperative peers are kept apart from each other by more than $k + 1$ hops. When a tentative cooperative peer does not receive any reject message in a given time, it finally becomes a cooperative peer. To select two or more cooperative peers, each candidate peer removes a new cooperative peer from the list and repeats the same procedures.

2.3 Discovering Other P2P Networks

A newly chosen cooperative peer first discovers a candidate peer in other P2P networks by using, for example the i3 network, which mediates communications among overlay networks. A cooperative peer sends a *trigger* message containing a service identifier and its address to the i3 network (Fig. 4). The last $m + n$ bits of the service identifier, which is used as an identifier of a P2P network and a candidate peer, are generated at random to find arbitrary P2P network registered in the i3 network. When a cooperative peer receives a *packet* message which matches the *trigger* message by *inexact matching* where the *packet* message has a service identifier matching the longest pattern of bits with the *trigger* message, it sends a cooperation request to the candidate peer, i.e., the sender of the *packet* message, in another P2P network. Next, the selection of a cooperative peer is initiated by the candidate peer in a newly discovered P2P network. Then, the cooperation request is forwarded from the candidate peer to a new cooperative peer. Finally, a logical link is established between those cooperative peers.

2.4 Decision of Invoking a Cooperation

Through a logical link established in the preceding step, cooperative peers at the both end of the link exchange information to decide whether they cooperate with each other or not. In the cooperation among P2P networks of file-sharing applications, we consider mutualistic symbiosis, where both P2P networks benefit from each other. However, mutualism is accomplished by the selfish decision of cooperative peers. A peer begins the cooperation to enhance and improve its own QoS. A peer maintains an inter-network logical link as far as it considers it is beneficial to itself. When cooperative peers at both ends of a logical link consider it is worth connecting, the link is kept. Therefore, the cooperation among P2P networks is a consequent of selfish behavior of cooperative peers. If P2P networks benefit from each other as a whole, they would be connected by many logical links and behave as a one large P2P network.

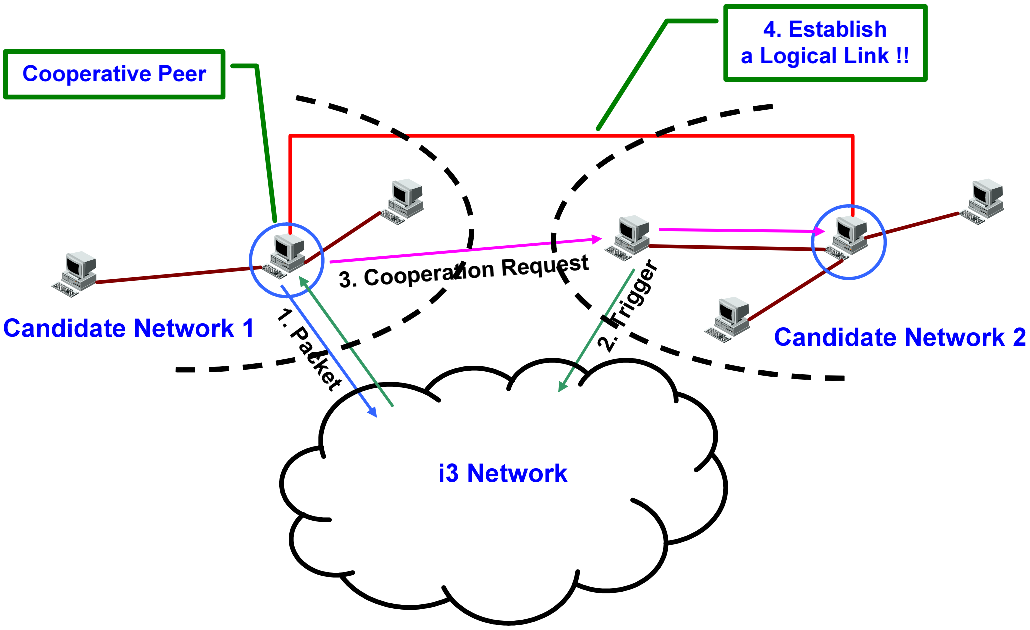


Figure 4: A cooperative peer discovers other P2P networks

The decision of invoking the cooperation is made taking into account some criteria, such as the compatibility between P2P file-sharing protocols, the size of P2P networks such as the number of peers and files, and the type of files shared in P2P networks.

When application protocols are different, cooperative peers must convert one protocol into the other. Therefore, it is desirable that protocols are the same or compatible to reduce the load on cooperative peers. When P2P networks are different in their size, peers in a larger P2P network cannot expect the benefit from the cooperation very much. However, the newly introduced load from a smaller cooperative P2P network is considered not much. On the other hand, peers in a smaller P2P network can share and discover more providing peers by the cooperation, but they receive a considerable amount of search messages from a larger P2P network. Therefore, cooperative peers must consider the trade-off between the benefit in the application-level QoS and the cost in the increased load by the cooperation. When the type and category of files shared in P2P networks are different, the effect of cooperation is rather small from the viewpoint of the application-level QoS. Therefore, it is desirable that P2P networks sharing similar files such as movies, music, and documents cooperate with each other. However, as mentioned in Section 1, it is worth cooperating with a different P2P network from a system-oriented viewpoint.

A cooperative peer obtains those information and defines priorities to each of them. When the weighted sum is beyond a threshold for both cooperative peers, the cooperation is started and continued. We should note that weight values and the threshold are determined by an application and details of its strategy and policy are left as one of future research works.

2.5 Cooperation in Exchanging Messages and Files

In the following, we call a P2P network where a search message is originated a guest network, and another P2P network a host network. In Fig. 2, *P2P network 1* is a guest network served by a host network, i.e., *P2P network 2*. A search message sent from a peer is disseminated over a guest network by a flooding scheme. When a search message reaches a cooperative peer, a cooperation program receives it (Fig. 5). The cooperation program looks up its local cache. Only if meta-information related to a desired file is not discovered in the cache, the search message is forwarded to a cooperative peer in a host network, after protocol conversion is applied if needed. At this time, the TTL value of the search message is decremented by one as in normal forwarding. A cooperative peer in a host network disseminates the search message over the host network by flooding. When there are two or more pairs of cooperative peers among guest and host networks, the same search message would be relayed to a host network. To eliminate the duplication, search messages have the same identifier independently of cooperative peers they traverse even if they are applied protocol conversion. Peers in a host network silently discard duplicated search messages with the same identifier.

If the desired file is discovered in a host network, a response message is generated by a providing peer and it reaches a cooperative peer in a host network along a reverse path of the corresponding search message. A cooperative peer in a host network transmits the response message to a cooperative peer in a guest network via a logical link, after protocol conversion if needed. In the case that a different protocol is used for file retrieval, a cooperative peer in a guest network caches a response message and replaces the address of a providing peer with its own address in the response message. A response message reaches the searching peer of along a reverse path of the search message over a guest network. The searching peer establishes a connection to a providing peer and obtains a file. In the case that a protocol for file retrieval is different, the peer regards a cooperative peer as a providing peer. Then, the cooperative peer retrieves the file from the original providing peer on behalf of the searching peer. Finally, the file is sent to the search-

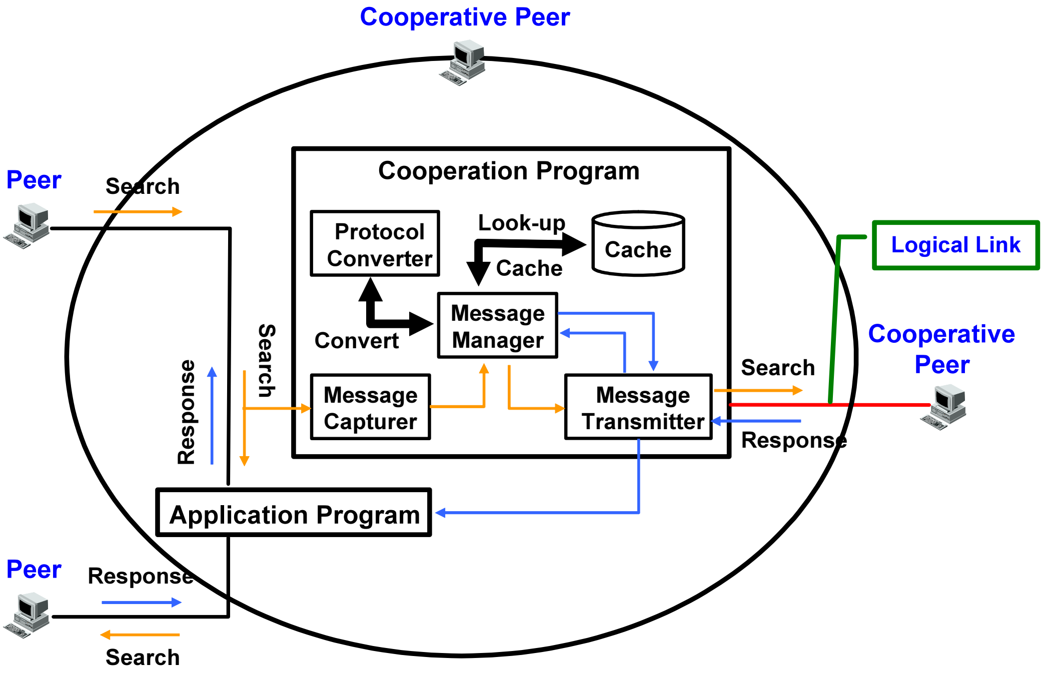


Figure 5: Behavior of a cooperative peer in the cooperation

ing peer. Therefore, peers do not need to recognize such cooperation to receive the benefit of the cooperation.

2.6 Caching Mechanism for Cooperative Peers

When P2P networks cooperate with each other, the load on peers increases because of the increased number of search messages injected by a guest network and that of response messages generated in a host network to answer them. More harmfully, those tremendous amounts of messages concentrate on cooperative peers and logical links established among them. They cause congestion and make cooperative peers and logical links overloaded. Therefore, we introduce a caching mechanism as one of functions of a cooperation program.

There are benefits in caching meta-information of files available in a host network at a cooperative peer of a guest network. First, the load on a host network is decreased, since it does not need to receive and respond search messages that it has already answered. Second, the load on logical links is also decreased, since search messages which hit a local cache at a cooperative peer of a guest network do not traverse the link and cache-hits further suppress the generation of response

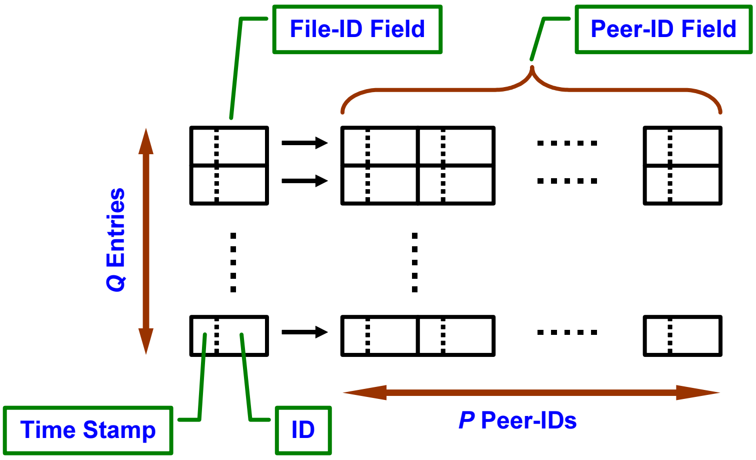


Figure 6: Construction of a caching mechanism

messages. Third, the load on cooperative peers is also decreased. For one search message forwarded to a host network, they would receive a large number of response messages, if the search is for a popular file. Fourth, the response time of search is decreased, since a peer does not need to explore a host network for a file.

A peer has a local cache of the limited capacity. In usual P2P file-sharing systems, each of peers that have a desired file generates a response message to answer the search message. Therefore, a search message for a popular file brings a large number of response messages to a cooperation peer. Consequently, when the whole of the cache is used to deposit meta-information using a LRU algorithm, it will obviously be occupied by meta-information of popular files. However, popular files are also discovered in a guest network. Therefore, to avoid the occupation of a cache with meta-information of popular files, we consider to put a limit on the number of meta-information for each file.

A cache has Q entries. Each entry consists of a file-ID, and a list of P providing peers and a timestamp. Each entry of a list of providing peers also has a timestamp (see Fig. 6). Therefore, the size of the whole cache amounts to $Q \times P$ meta-information. For more obvious discussions and experiments, we only consider a set of a file-ID and peer-IDs as a meta-information, but our scheme can easily extended to the case with other form of meta-information such as attributes and keywords.

When a response message reaches a cooperative peer, the cooperative peer obtains a file-ID and peer-IDs from the message. If there is no entry of the same file-ID in a cache, a new entry is made for the meta-information. When there are already Q entries in a cache, the entry with the oldest timestamp is replaced with the new entry. Timestamps are given to both of the new entry of a file-ID and that of a peer-ID. If the meta-information of the same file-ID is in the cache, the entry is renewed with the current timestamp. Then, a list of providing peers is investigated to see whether there already is the same peer-ID or not. If there is, the current timestamp is given to the entry of the providing peer. Otherwise, the peer-ID is added to the list with the current timestamp, or the oldest peer-ID is replaced with the new peer-ID in a full list.

On receiving a search message from peers in the same P2P network, a cooperative peer first examines its local cache. If there is a match in the cache, it generates a response message constituting a list of providing peers and sends it back to the searching peer via a reverse path that the search message traversed. At the same time, the timestamp of the entry of the file-ID is updated with the current time. Otherwise, the search message is forwarded to a host network.

3 Evaluations by Simulation Experiments

In this section, we conduct several preliminary simulation experiments to evaluate our proposed mechanism. To see what happens when two P2P networks cooperate with each other, we consider two static P2P networks of the same or different size and the same protocol.

3.1 Simulation Conditions

We generate two scale-free networks of 1,000 peers, 5,000 peers, and 10,000 peers based on BA model [13]. We assume that logical links among peers have infinite capacity and zero latency. We consider static and stable networks where there is no change in their topologies due to joins and leaves of peers. In both P2P networks, there are files whose popularity is determined by a Zipf distribution with $\alpha = 1.0$. The number of files also follows a Zipf distribution with $\alpha = 1.0$, where the number of the most popular file is half of the number of all peers, and the number of the least popular file is 1. For example, in two P2P networks of 10,000 peers, there are 10,000 types of 93,668 files. Figure 7 illustrates the cumulative distribution of the number of files against the file popularity. In the figure, $n_1 : n_2$ indicate the cooperation between a P2P network of n_1 peers and a P2P network of n_2 peers. Files are placed on randomly chosen peers. A search message is generated at a randomly chosen peer for a file determined in accordance with the popularity. In each case, the number of search messages is the same number of all peers. It is disseminated by flooding within the range limited by a TTL value of 7, the default value of Gnutella. To keep the distribution of files to follow a Zipf, a peer does not retrieve a file in our evaluation.

In our simulation experiments, we assume that cooperative peers are selected from all peers, that is, all peers are candidate peers. The number of cooperative peers is set at 10. Therefore, there are 10 logical links among P2P networks. A cache of a cooperative peer has the capacity of $Q = 50$ entries of file-IDs, each of which maintains a list of up to $P = 10$ peer-IDs. We carry out simulation experiments 40 times in each case.

3.2 Performance Evaluations

Metrics of our evaluation are the number of discovered providing peers, the search latency, and the load on peers. The number of discovered providing peers is defined as the average number of providing peers discovered in P2P networks per search message. The search latency corresponds

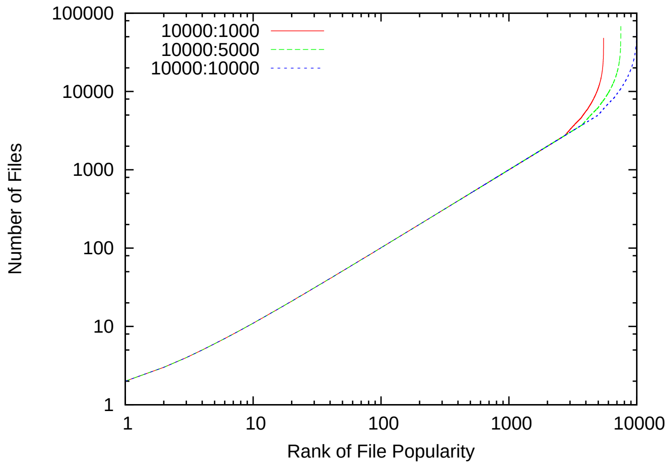


Figure 7: CDF of the number of files against the file popularity

to the number of hops between a searching peer and the nearest providing peer in P2P networks. The load on peers is the average number of messages that a peer sends and receives.

For comparison purposes, we conducted simulation experiments for different methods to select cooperative peers. “Descending Order of Degree” in the following figures corresponds to the degree-dependent selection where cooperative peers are chosen in a descending order of degree. In “random”, cooperative peers are chosen at random. “Uncooperative” indicates the result of the case where there is no cooperation among P2P networks. “Proposal (Distance $>= d$)” shows performance of our proposal where cooperative peers are chosen in descending order of degree and they are apart from each other by at least d hops. In this case, a TTL value of a confirmation message is set at $d - 1$.

3.2.1 Number of Discovered Providing Peers

Figures 8, 9, and 10 illustrate the relationship between the number of discovered providing peers and the file popularity for the cooperation among P2P networks of various sizes. For comparison purposes, we also show results of the case that cooperative peers do not have a local cache. The X-axis corresponds to the rank of file popularity, and the Y-axis does the number of discovered providing peers. The reason of step-shaped lines is that the number of files, which follows a Zipf distribution, takes integer values based on the popularity. It is shown that by connecting two P2P networks by the degree-dependent selection methods such as “Descending Order of Degree” and

“Proposal”, a peer can discover more providing peers than that of “Random”. In addition, it can be seen that the number of discovered providing peers of “Random” is almost the same as that of “Uncooperative”. It means that the cooperation among P2P networks by degree-dependently chosen cooperative peers improve the application-level QoS, but that of randomly chosen cooperative peers does not do so. Since the majority is low-degree peers in a power-law network, a random selection method often chooses low-degree peers as cooperative peers which cannot effectively disseminate search messages over a host P2P network.

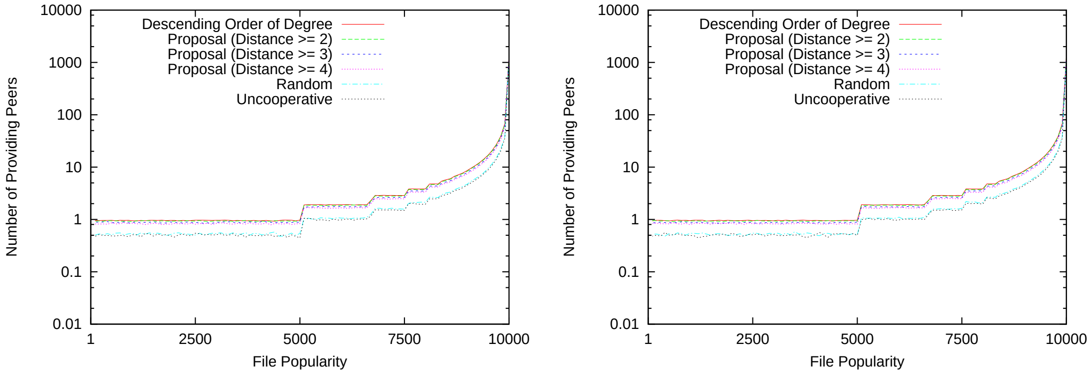
When the size of cooperative P2P networks is different, the number of discovered providing peers of “Random” is obviously different between P2P networks. It is shown that peers cannot discover providing peers in another P2P network since P2P networks do not cooperate with each other efficiently and effectively. However, those of “Descending Order of Degree” and “Proposal” are almost the same between P2P networks. If cooperative peers are chosen appropriately, every peer in P2P networks can discover providing peers in a similar manner regardless of the size of P2P networks. Comparing the case of cooperative peers with a local cache with the case of them without a local cache, the number of discovered providing peers is almost the same.

3.2.2 Search Latency

Figures 11, 12, and 13 illustrate the relationship between the number of hops between a searching peer and the nearest providing peer and the file popularity. The case that searching peers cannot discover any providing peers is not taken into account. The X-axis corresponds to the rank of file popularity, and the Y-axis does the number of hops. The numbers of hops of “Descending Order of Degree” and “Proposal” are larger than those of “Random” and “Uncooperative” in all cases.

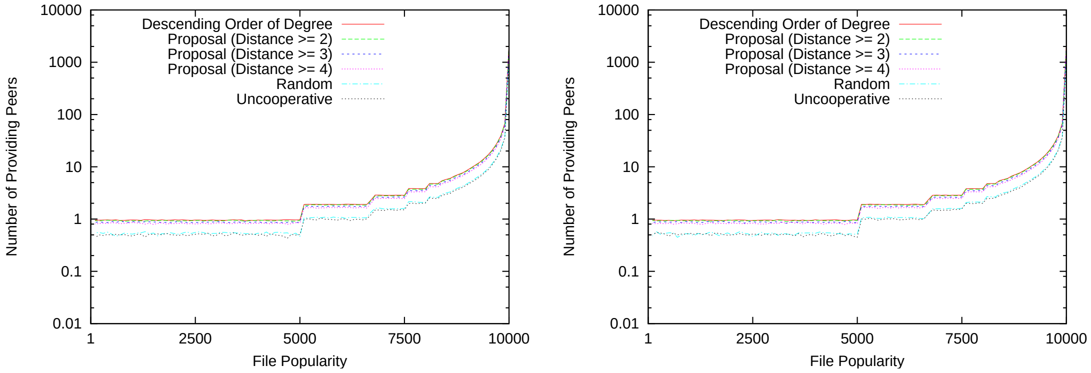
It can be seen that caching at cooperative peers does not contribute to faster search. When a peer searches for a popular file, it can discover the nearest providing peer in a guest network. Even if there is a match in a local cache, the cooperative peer is not necessarily the nearest providing peer. On the other hand, when a peer searches for a unpopular file, it can discover the desired file in a host network in cooperating networks especially with degree-dependent selection methods. In such cases, the number of hops to the nearest providing peer becomes large. On the other hand, a peer cannot discover a unpopular file and, as a result, the number of hops is small in “Random” and “Uncooperative” cases.

When the size of cooperative P2P networks is different, the difference among selection meth-



(a) Network1

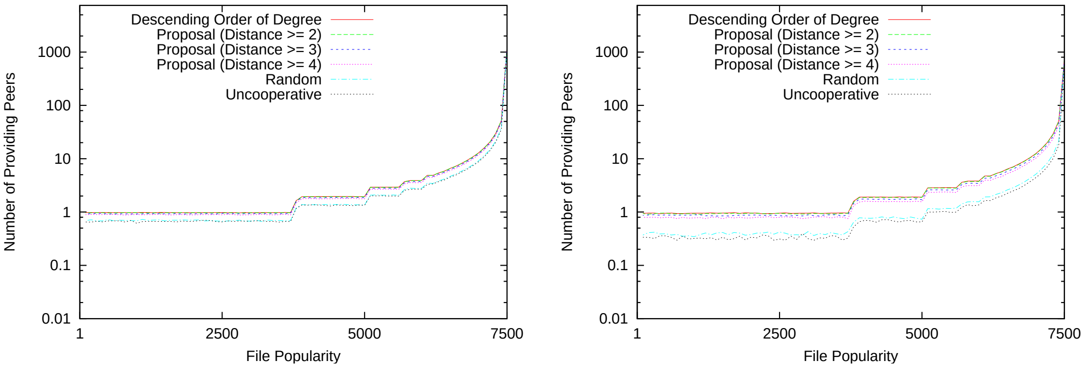
(b) Network2



(c) Network1 (without cache)

(d) Network2 (without cache)

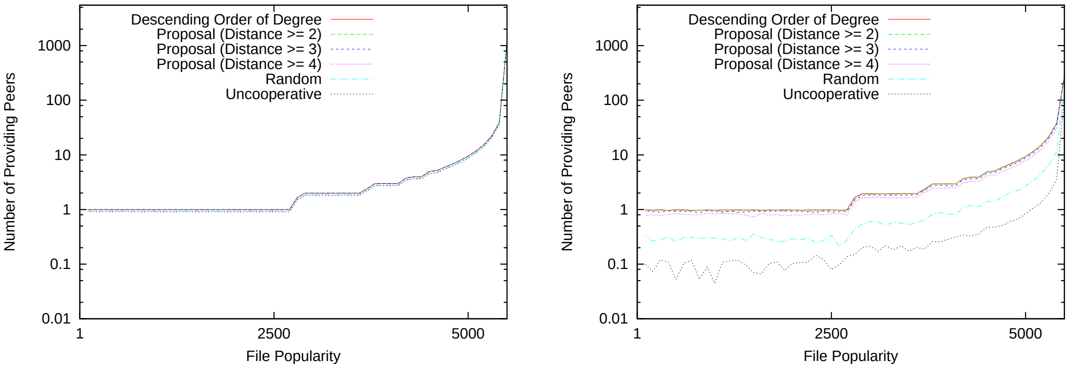
Figure 8: Relationship between the number of discovered providing peers and the file popularity (10,000:10,000)



(a) Network1

(b) Network2

Figure 9: Relationship between the number of discovered providing peers and the file popularity (10,000:5,000)



(a) Network1

(b) Network2

Figure 10: Relationship between the number of discovered providing peers and the file popularity (10,000:1,000)

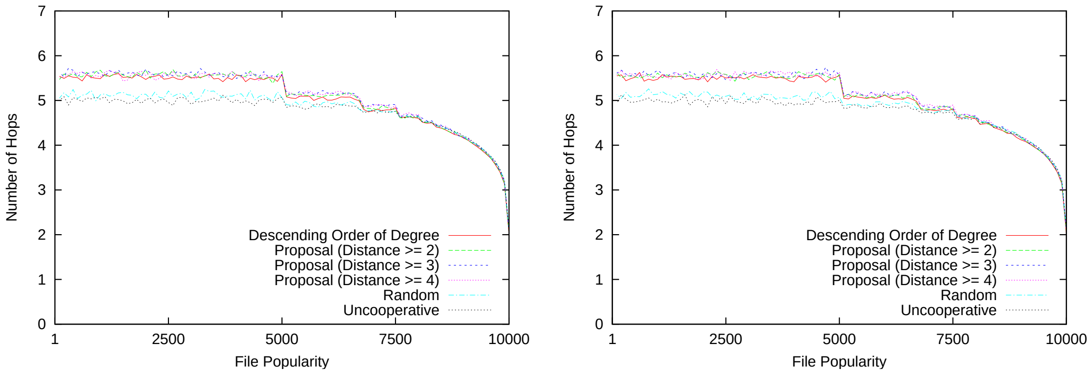
ods is small in a larger P2P network. Especially in Fig. 13(a), there is no change in the number of hops in a larger P2P network, because a peer in a larger P2P network can discover a desired file in its own P2P network. On the other hand, the number of hops in a smaller P2P network increases more when it cooperates with a larger P2P network.

3.2.3 Load on Peers

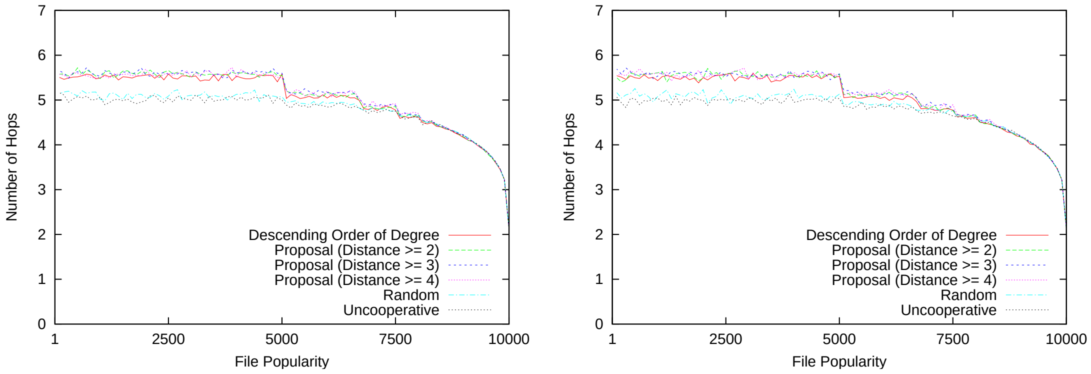
Figures 14, 15, and 16 show how the load on a peer increases by the cooperation. The ratio of increase on Y-axis is derived as the ratio of the number of duplicated search messages that a peer receives in cooperating P2P networks to that in uncooperative P2P networks. The duplicated search messages are redundant and waste physical network resources and the processing power of peers. In comparison with “Descending Order of Degree”, our proposed selection method can reduce the number of duplicated search messages at most of peers. In P2P networks used in simulation experiments, most of high-degree peers are connected with each other and form the core of a P2P network. Since cooperative peers are selected purely based on their degree in “Descending Order of Degree”, they quickly flood the core of a P2P network with copied and duplicated search messages. On the other hand, in “Proposal”, cooperative peers are apart from each other in a P2P network. Then, concentration of the load on high-degree peers are avoided at the sacrifice of slight increase of the load on medium-degree peers which are chosen as cooperative peers.

As the difference of the sizes of P2P networks becomes larger, a smaller P2P network suffers more from duplicated search messages. The reason is that a larger P2P network introduces many search messages into a smaller P2P network as a result of message exchanges.

On the other hand, Figs. 17, 18, and 19 show the ratio of increase in the number of messages that a peer sends and receives including duplicated messages in cooperating P2P networks to that in uncooperative P2P networks. The load on high-degree peers, which are chosen as cooperative peers, increases as the number of hops among cooperative peers increases in our proposed selection method. However, the load on lower-degree peers decreases than that of “Descending Order of Degree”. In a power-law network, there are high-degree peers at the core of network. Most of cooperative peers are selected from low-degree peers as the number of hops among cooperative peers increases with our proposed method. Thus, the highest-degree cooperative peer, which is first selected as a cooperative peer, can disseminate search messages more effectively than the

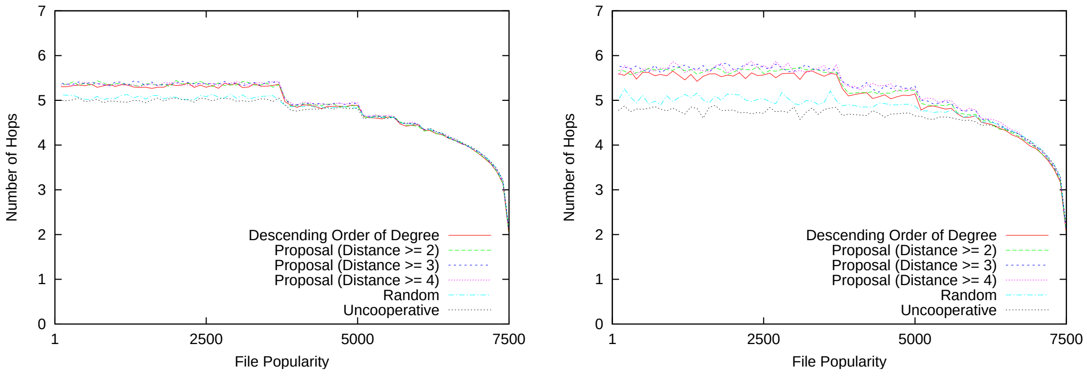


(a) Network1 (b) Network2



(c) Network1 (without cache) (d) Network2 (without cache)

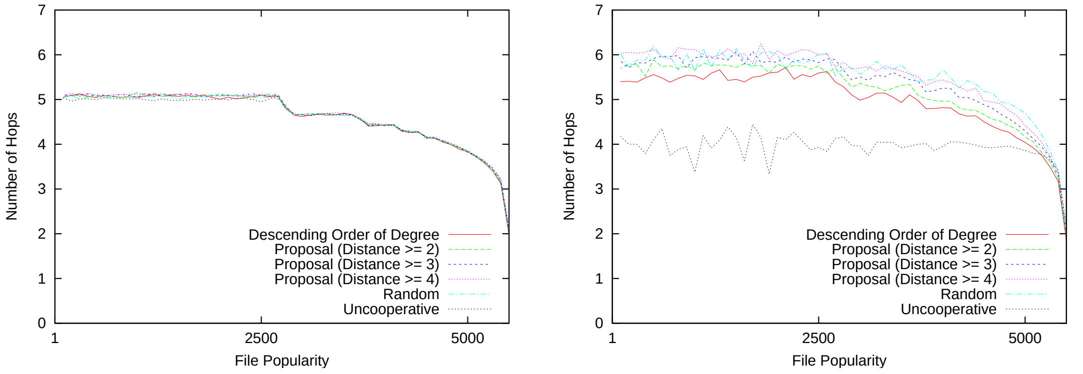
Figure 11: Relationship between the number of hops between a searching peer and the nearest providing peer and the file popularity (10,000:10,000)



(a) Network1

(b) Network2

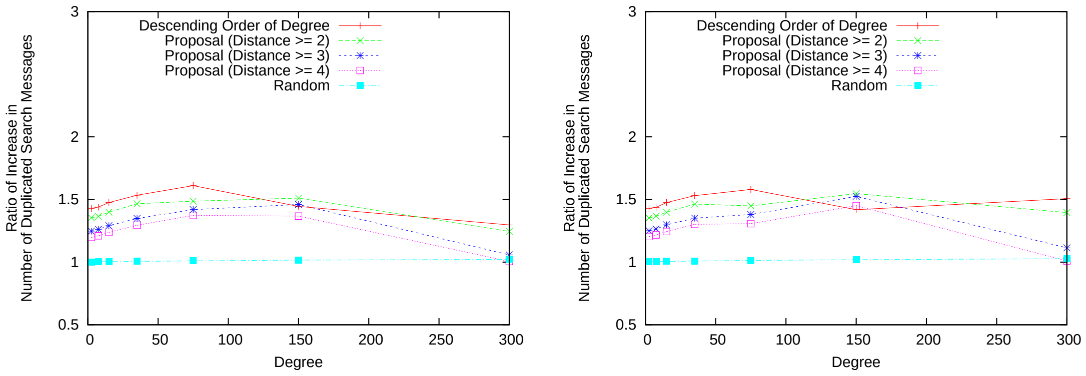
Figure 12: Relationship between the number of hops between a searching peer and the nearest prividing peer and the file popularity (10,000:5,000)



(a) Network1

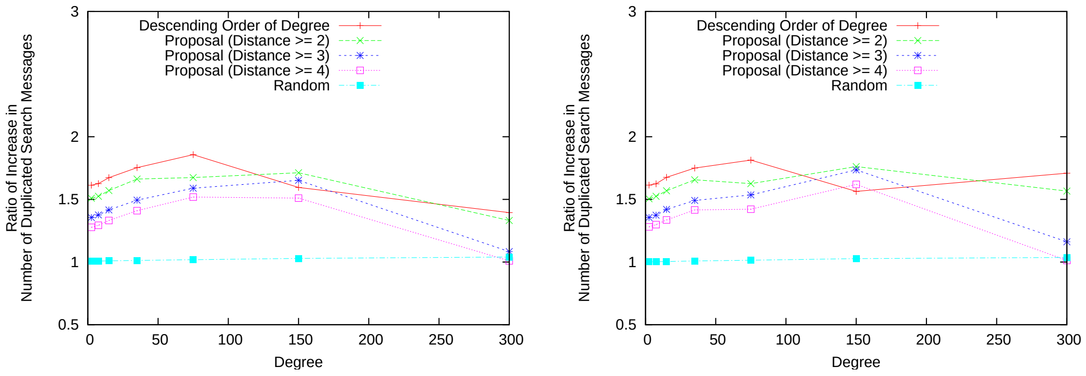
(b) Network2

Figure 13: Relationship between the number of hops between a searching peer and the nearest prividing peer and the file popularity (10,000:1,000)



(a) Network1

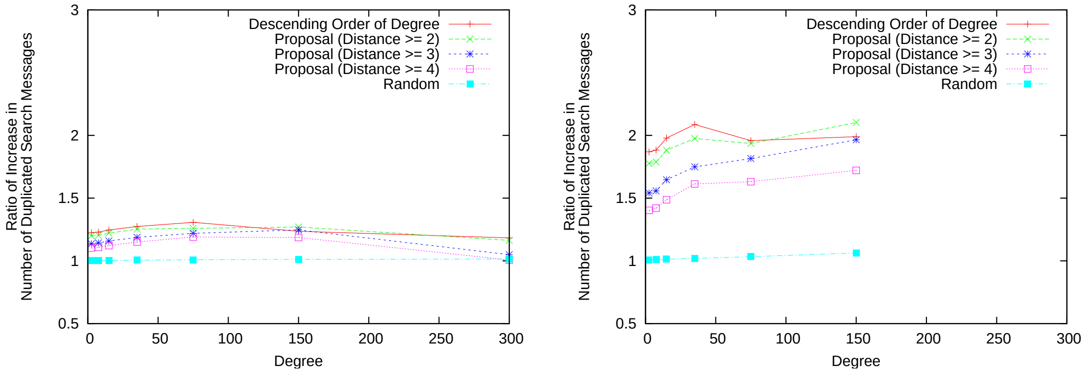
(b) Network2



(c) Network1 (without cache)

(d) Network2 (without cache)

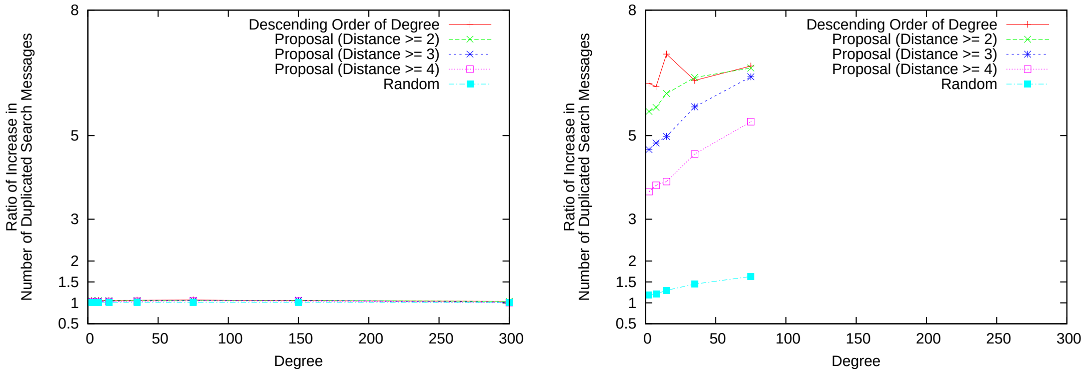
Figure 14: Distribution of the ratio of increase in the number of duplicated search messages (10,000:10,000)



(a) Network1

(b) Network2

Figure 15: Distribution of the ratio of increase in the number of duplicated search messages (10,000:5,000)



(a) Network1

(b) Network2

Figure 16: Distribution of the ratio of increase in the number of duplicated search messages (10,000:1,000)

other cooperative peers. As a result, it must relay a large number of response messages for a search message. However, by introducing a caching mechanism into cooperative peers, the load on the highest-degree cooperative peer becomes below the half as shown in Fig. 17. A cache can reduce the number of response messages, which are the majority of messages that a peer handles, since a search message that hits a cache does not bring any response messages.

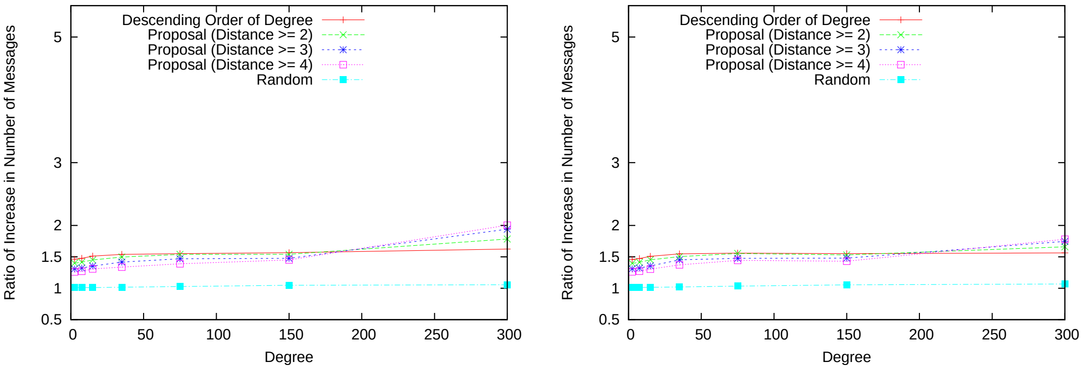
Even when we consider all messages, a peer in a smaller network suffers more from the cooperation as shown in Figs. 18 and 19. However, it benefits very much in discovering more providing peers as shown in Figs. 9 and 10. For example, a peer in a smaller network can discover ten times as many providing peer as in an uncooperative network when the network cooperates with a ten-times larger network.

3.2.4 Advantage and Disadvantage of Cooperative Peers

In our proposed mechanism, a cooperative peer examines its local cache to discover meta-information to answer search messages it receives. In addition, it also uses a cache for search messages generated by itself. If there is a match, it can discover providing peers without spending time and consuming bandwidth in exploring P2P networks.

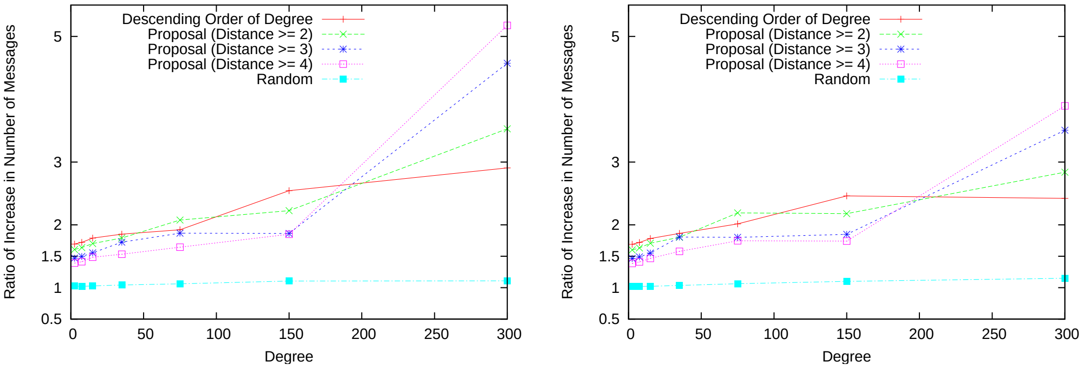
Table 1 summarizes the average ratio of cache-hits. The ratio of cache-hits ranges 32–42% depending on settings. The ratio becomes higher as the number of hops among cooperative peers increases. As the distance becomes larger, lower-degree peers are to be chosen as cooperative peers. A low-degree cooperative peer only receives search and response messages for popular files whereas a high-degree cooperative peer receives many messages for both of popular and unpopular files. Consequently, a low-degree cooperative peer has meta-information of popular files only. Since desired files are chosen following a Zipf distribution, a low-degree cooperative peer offers a higher hit ratio than a high-degree cooperative peer. In addition, we can observe that the ratio of cache-hits is higher for the cooperation among P2P networks of different sizes. This is because that the number of files is proportional to the number of peers. Then, a cooperative peer can cache a relatively large number of meta-information in a small-sized P2P network.

Now let us consider the processing capacity that a peer needs. In Gnutella, a search message consists of the header of 23 bytes and the payload of more than 2 bytes. A response message consists of the header of 23 bytes and the payload of more than 27 bytes. When we assume that the average message size is 50 bytes and each peer generates a search message per minute, the



(a) Network1

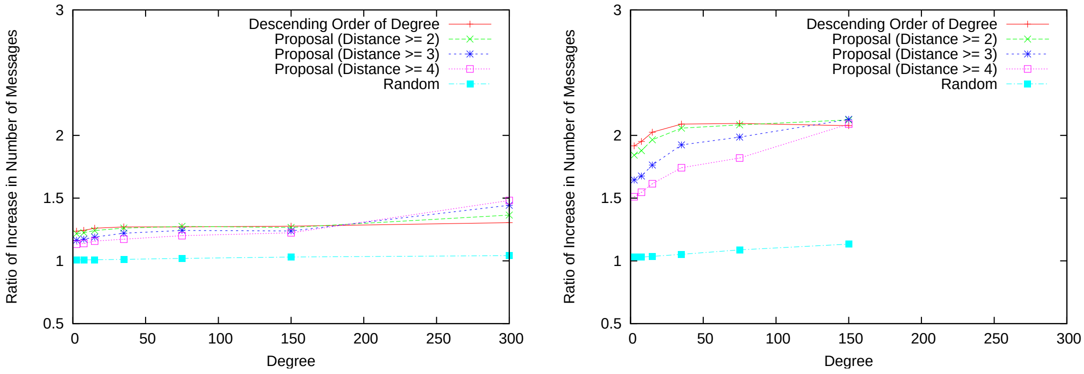
(b) Network2



(c) Network1 (without cache)

(d) Network2 (without cache)

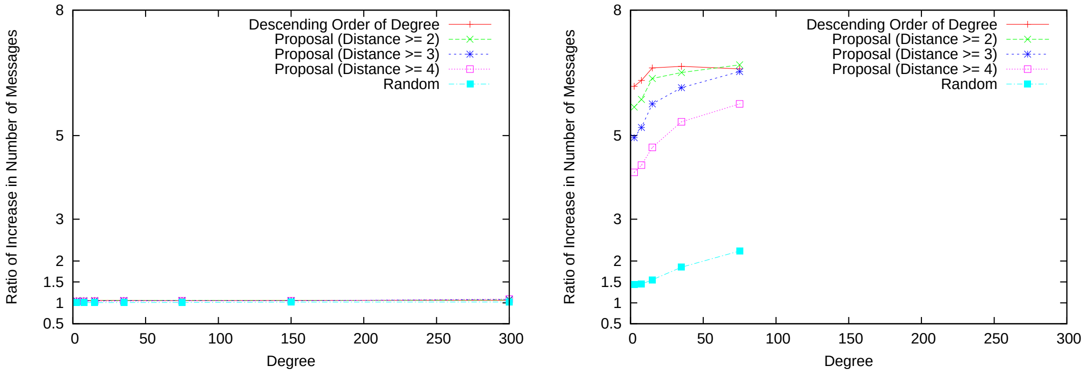
Figure 17: Distribution of the ratio of increase in the number of messages (10,000:10,000)



(a) Network1

(b) Network2

Figure 18: Distribution of the ratio of increase in the number of messages (10,000:5,000)



(a) Network1

(b) Network2

Figure 19: Distribution of the ratio of increase in the number of messages (10,000:1,000)

average load on peers and cooperative peers are calculated as in tables 2, 3, and 4. In the cooperation between two P2P networks of 10,000 peers, the processing capacity required for peers of "Descending Order of Degree" is larger by about 1.5 times than that of "Uncooperative". Furthermore, the processing capacity required for cooperative peers is larger by about 45 times than that of all peers. When there is the difference among the size of P2P networks, the ratio of increase in the required processing capacity in a smaller P2P network becomes higher than that in a larger P2P network. Therefore, to accomplish the cooperation among pure P2P file-sharing networks, all peers, especially cooperative peers, need much processing capacity. However, if cooperative peers are placed apart from each other by our proposed selection method, the required processing capacity of them can be reduced to about the half.

Table 1: Ratio of cache-hits at cooperative peers [%]

Selection Method	10,000:1,000		10,000:5,000		10,000:10,000	
	Net.1	Net.2	Net.1	Net.2	Net.1	Net.2
Descending Order of Degree	40.7	36.9	35.5	35.0	32.8	33.5
Proposal (Distance $>=$ 2)	41.4	37.8	36.7	35.5	34.3	34.3
Proposal (Distance $>=$ 3)	41.6	38.0	37.9	37.0	35.8	35.8
Proposal (Distance $>=$ 4)	41.1	38.8	39.7	38.9	38.2	38.2
Random	40.3	38.4	38.3	37.7	37.2	37.1

Table 2: Required processing capacity of peers and cooperative peers (10,000:10,000) [KB/s]

Selection Method	Network1		Network2	
	All Peers	Co. Peers	All Peers	Co. Peers
Descending Order of Degree	80.5	3614	80.3	3800
Proposal (Distance $>=$ 2)	77.5	2532	77.4	2578
Proposal (Distance $>=$ 3)	72.7	1550	72.8	1561
Proposal (Distance $>=$ 4)	69.9	918	70.1	1140
Random	54.9	78.8	55.0	85.0
Uncooperative	54.2	-	54.1	-

Table 3: Required processing capacity of peers and cooperative peers (10,000:5,000) [KB/s]

Selection Method	Network1		Network2	
	All Peers	Co. Peers	All Peers	Co. Peers
Descending Order of Degree	67.8	2966	53.7	1477
Proposal (Distance $>=2$)	66.8	2054	52.0	1119
Proposal (Distance $>=3$)	64.2	1242	47.0	737
Proposal (Distance $>=4$)	62.4	714	43.1	453
Random	54.8	59.2	28.2	51.5
Uncooperative	54.4	-	27.3	-

Table 4: Required processing capacity of peers and cooperative peers (10,000:1,000) [KB/s]

Selection Method	Network1		Network2	
	All Peers	Co. Peers	All Peers	Co. Peers
Descending Order of Degree	57.6	2481	34.6	425
Proposal (Distance $>=2$)	57.4	1691	32.4	260
Proposal (Distance $>=3$)	57.0	1003	29.0	171
Proposal (Distance $>=4$)	56.6	559	24.2	99.1
Random	54.9	42.3	8.52	20.0
Uncooperative	54.6	-	5.42	-

4 Conclusion

In this thesis, in a context of the overlay network symbiosis, we proposed a mechanism for pure P2P networks of file-sharing applications to cooperate with each other. Through several simulation experiments, it was shown that application-level QoS in term of the number of discovered providing peers was improved by selecting high-degree peers as cooperative peers. Furthermore, it was shown that by keeping cooperative peers apart from each other, the redundant load on the P2P network was reduced. A caching mechanism for cooperative peers was shown to be effective in reducing the load on cooperative peers, but it did not contribute to faster search.

As future research works, we will investigate behaviors of the cooperation among dynamic P2P networks, which change their topology by joins, moves, and leaves of peers. In this case, search messages cannot be disseminated enough because a P2P network would be separated into more than two parts. If P2P networks cooperate with each other, they can be disseminated among separated parts of the P2P network through cooperative P2P networks. Furthermore, we should evaluate influences of the cooperation to physical networks.

Acknowledgements

I would like to express my gratitude to my supervisor, Prof. Masayuki Murata of Osaka University, for his extensive and continuous support and invaluable comments on my studies.

I would like to express my appreciation to Associate Prof. Naoki Wakamiya of Osaka University who has always given me appropriate guidance and valuable advice.

I am deeply grateful to Profs. Koso Murakami, Makoto Imase, Teruo Higashino, Hirotaka Nakano, and Tetsuji Satoh, Associate Prof. Go Hasegawa, Research assistants Shin'ichi Arakawa and Masahiro Sasabe of Osaka University, who gave me helpful comments and feedback.

Finally, I heartily thank my seniors and colleagues in the Department of Information Networking, Graduate School of Information Science and Technology of Osaka University for their help.

References

[1] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker, “On Selfish Routing in Internet-Like Environments,” in *Proceedings of ACM SIGCOMM Conference 2003*, pp. 151–162, Aug. 2003.

[2] M. Seshadri and R. H. Katz, “Dynamics of Simultaneous Overlay Network Routing,” Tech. Rep. UCB/CSD-03-1291, Electrical Engineering and Computer Sciences (EECS), University of California Berkeley (UCB), Nov. 2003.

[3] M. Kwon and S. Fahmy, “Toward Cooperative Inter-overlay Networking,” in *Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP)*, poster paper, Nov. 2003.

[4] M. Kwon and S. Fahmy, “Synergy: An Overlay Internetworking Architecture and its Implementation,” in *Proceedings of the 14th IEEE International Conference on Computer Communications and Networks (ICCCN)*, pp. 401–406, Oct. 2005.

[5] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, “Internet Indirection Infrastructure,” in *Proceedings of ACM SIGCOMM Conference 2002*, pp. 73–88, Aug. 2002.

[6] S. Zhuang, K. Lai, I. Stoica, R. Katz, and S. Shenker, “Host Mobility Using an Internet Indirection Infrastructure,” in *Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 129–144, May 2003.

[7] A. Nakao, L. Peterson, and A. Bavier, “A Routing Underlay for Overlay Networks,” in *Proceedings of ACM SIGCOMM Conference 2003*, pp. 11–18, Aug. 2003.

[8] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, “Resilient Overlay Networks,” in *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP)*, pp. 131–145, Oct. 2001.

[9] N. Wakamiya and M. Murata, “Toward Overlay Network Symbiosis,” in *Proceedings of the 5th IEEE International Conference on Peer-to-Peer Computing (P2P2005)*, pp. 154–155, Aug. 2005.

[10] H. Fu, N. Wakamiya, and M. Murata, “Proposal and Evaluation of a Cooperative Mechanism for Hybrid P2P File Sharing Networks,” in *Proceedings of the 4th IASTED International*

Conference on Communications, Internet, and Information Technology (CIIT2005), pp. 7–13, Oct. 2005.

[11] M. E. J. Newman, “The Structure and Function of Complex Networks,” *SIAM Review*, vol. 45, 2, pp. 167–256, 2003.

[12] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, “Search in Power-law Networks,” *Physical Review E*, vol. 64, 046135, Sept. 2001.

[13] A. L. Barabasi and R. Albert, “Emergence of Scaling in Random Networks,” *Science*, vol. 286, pp. 509–512, Oct. 1999.