Master's Thesis

Title

Design and Evaluation of a Cooperative Mechanism of Hybrid P2P File-Sharing Networks to Enhance Application-Level QoS

Supervisor

Professor Masayuki Murata

Author

Hongye Fu

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 of Hybrid P2P File-Sharing Networks to Enhance Application-Level QoS

Hongye Fu

Abstract

Overlay networks, such as P2P, Grid, and CDN, have been widely deployed over physical IP networks. Since simultaneous overlay networks share and compete for network resources, their selfish behaviors to improve their application-oriented QoS disrupt each other. To enhance the collective performance and improve the QoS at the application-level, we consider so-called the overlay network symbiosis where overlay networks cooperate with each other.

In this paper, we propose two cooperative approaches for hybrid P2P file-sharing networks by which two or more hybrid P2P file-sharing networks can efficiently cooperate with each other and peers can find more files and exchange files with more peers. Then, we propose a cooperative mechanism for the Shared-Peer-Based approach in detail. In the SPB approach, a peer belonging to multiple P2P networks becomes a cooperative peer and relays query and response messages among cooperative P2P networks.

Through simulation experiments, we verify the effectiveness of cooperation from view points of application and system. We show that the application-level QoS is improved in terms of the number of available files, the hit ratio, and the number of provider peers. Furthermore, a caching mechanism is useful at improving the response time. As the benefit of cooperative peers, the response time of cooperative peers is decreased by 15–33% than normal peers. On the other hand, however, our results also indicate that the system load would be increased by cooperation of networks. We discuss systematic setup for effective cooperation.

Keywords

Overlay Networks

P2P (Peer-to-Peer)

Cooperation

File-Sharing

Contents

List of Figures

List of Tables

1 Introduction

With emerging needs for application-oriented network services, overlay networks, such as P2P, Grid, and CDN, have been widely deployed over physical IP networks. To satisfy their own application-level QoS, these overlay networks first probe physical networks, for example using *ping* and *traceroute*, to learn and monitor underlaying physical topologies and dynamic attributes such as available bandwidth and delay. Then, based on those observations, overlay networks dynamically construct more efficient overlay topologies, select more efficient paths, and conduct traffic engineering independently. Since those behaviors are selfish and greedy, these overlay networks compete for limited physical resources and disrupt each other [1].

To enhance the collective performance of competing overlay networks and efficiently utilize network resources, several research papers on cooperation among overlay networks have been published in recent years [1-6]. In [2], they proposed a new architectural element—a shared routing underlay—that sits between overlay networks and the underlying Internet. Overlay networks query the routing underlay when making application-specific routing decisions. The underlay, in turn, extracts and aggregates topology information from the underlying Internet. In [3, 4], they proposed an overlay-based Internet Indirection Infrastructure (i3) that offers a rendezvous-based communication abstraction. In the i3 network, instead of explicitly sending a packet to a destination, each packet is associated with an identifier, this identifier is then used by the receiver to obtain delivery of the packet. A receiver R that inserts a trigger (id, R) in the i3 infrastructure receives all packets with identifier id. Based on the i3 network architecture, a wide variety of fundamental communication services are efficiently supported, such as mobility, multicast, anycast, and so on. In [5, 6], they investigated a spectrum of cooperation among coexisting overlay networks. As an example, they proposed an architecture where overlay networks cooperated with each other in inter-overlay routing where a message from one overlay network was forwarded to another which provided a shorter path to the destination.

The analysis on coexistence of competitors has been investigated in the field of biology. In the ecosystem, organisms in the same environment live together with direct and/or indirect interactions with each other. In [7], they established the mathematical model of the metabolic pathways of bacterial strains to elucidate mechanisms of coexistence of living organisms of closely related species. They revealed that the coexistence emerged not only from interactions among competitors, but also from changes of their internal states.

Taking inspirations from biology and based on the model in [7], our research group consider the symbiosis among competing overlay networks [8-10]. In the model of symbiotic overlay networks, overlay networks built on the same system evolve, interact with each other, and dynamically change internal structures, as living organisms in the same environment do. Overlay networks meet and communicate with each other in a probabilistic way. Overlay networks that benefit from each other reinforce their relationship, eventually have many inter-overlay links, and become one. Otherwise, they part from each other. All of 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 network emerges as a consequence of independent and autonomous behaviors of nodes and networks.

For cooperation among overlay networks, an effective and efficient cooperative mechanism is important and necessary, by which overlay networks efficiently communicate with each other. Although a variety of overlay networks are deployed over physical IP networks, in this paper, we focus on the cooperation among hybrid P2P file-sharing networks. Both in terms of the number of users and traffic volume, a P2P file-sharing network is one of the most important overlay networks in the Internet today. P2P networks are categorized into two, those are, a hybrid P2P network with central entities such as meta-servers, and a pure P2P network without any servers [11]. Whereas a pure P2P network consists of only peers, a typical hybrid P2P file-sharing network involves peers and one or more meta-servers which maintain a directory of available files on a P2P network to assist peers in finding files.

As shown in Fig. 1, a hybrid P2P file-sharing network has a two-tiers topology, where meta-servers compose a core network and peers are connected to meta-servers to form star-shaped networks. When a client joins a hybrid P2P network, it connects with a meta-server and then registers meta-information about files to share with the other peers. In a hybrid P2P application, a peer sends a query message to a connected meta-server to find a file. If meta-information about the desired file exists in its directory, the meta-server sends a response message to the querying peer. Otherwise, the meta-server forwards the query message to other neighboring meta-servers. On receiving a query message, a meta-server investigates its directory. If it does not have any meta-information matching a query message, it forwards the query message to its neighboring meta-servers except that one from which it received the query message. This method is generally called flooding. If

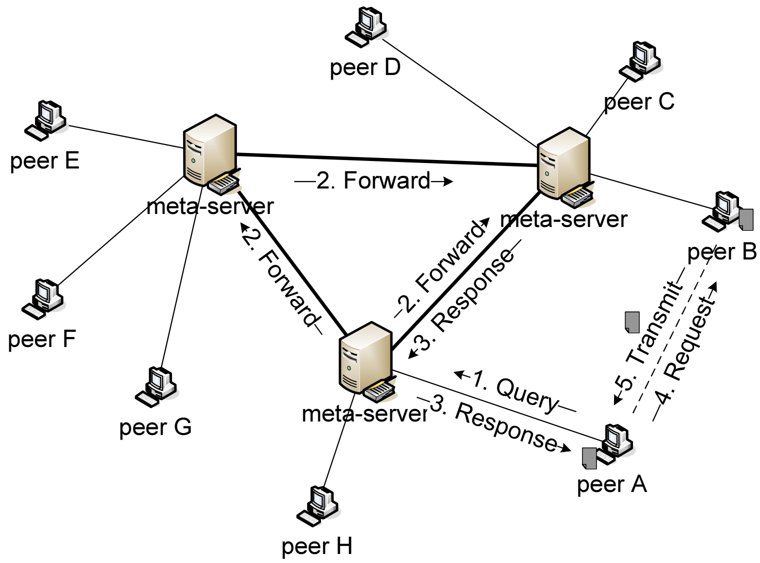

Figure 1: A Hybrid P2P File-Sharing Network

a meta-server has any matching meta-information, it generates a response message and sends it to the neighboring meta-server from which it received the query message. A response message is relayed among meta-servers following a reverse path of the corresponding query message and finally it reaches the querying peer. Once a peer receives a response message, it can directly retrieve a file from a provider peer, which appears in the response message.

There are several benefits of cooperative P2P file-sharing networks from various aspects of application and system. In a P2P file-sharing application, when a peer failed in finding a file, it often repeatedly retries searches by changing keywords. Such repetitive and redundant query messages increase the load on meta-servers and waste physical network resources. In addition, when a meta-server halts for the overload or links among meta-servers are disconnected, meta-information on a failed or isolated meta-server is lost or becomes inaccessible. As a result, the possibility of successful search decreases and the application-level QoS deteriorates. If multiple P2P file-sharing networks cooperate with each other and share their files among them by exchanging query and response messages, a peer can find more files at more peers and the possibility of successful search increases. Consequently, the number of repetitive and redundant query messages decreases and resultant extra load also decreases. In addition, a peer can choose the best, i.e., the fastest or the most reliable peer among many provider peers found in a search. Cooperation among

P2P networks also brings benefits even if they share different types or categories of files. Relaying message among inter-connected overlay networks provides faster and more reliable message transmission [5, 6]. Furthermore, it improves the robustness and the resilience of P2P networks as verified in [8]. Even when a P2P network is disconnected due to failures of meta-servers or links, meta-servers and peers in separated P2P networks are still able to communicate with each other since their message are relayed through cooperating P2P networks.

The cooperation among hybrid P2P networks is achieved by exchanging query and response messages among them. To accomplish effective and efficient cooperation in a transparent way where other meta-servers and peers are unaware of the cooperation, several difficulties must be faced and considered. First, how do P2P networks discover each other and decide to cooperate with each other? Next, how do cooperative P2P networks exchange query and response messages among them? For example, peers in hybrid P2P networks do not either receive any query messages nor send any response messages. Therefore, when we consider a peer as a point of cooperation, we need a mechanism for a peer to receive and relay query and response messages. Finally, when a desired file is found in a cooperative P2P network where a different protocol for file retrieval is employed, how does a querying peer retrieve its desired file? To solve these difficulties, we consider that a peer or a meta-server introduces a cooperative program to achieve the cooperation among hybrid P2P file-sharing networks.

In this paper, we propose two cooperation approaches. One is called a *Shared-Peer-Based Approach*, where peers participating in multiple P2P networks play a role of cooperative points or gateway nodes, and the other is called a *Server-Chain-Based Approach*, in which hybrid P2P networks cooperate through inter-meta-server connections. We focus on the former and describe detailed mechanisms for a Shared-Peer-Based approach. Through simulation experiments, we verified that our proposed cooperative mechanisms for the Shared-Peer-Based approach can improve the application-level QoS in terms of the number of available files, the hit ratio, and the number of provider peers. Furthermore, to improve the response time and reduce the system load by cooperation of P2P networks, we consider to introduce cache system into cooperative peers. A similar idea can be found in [12], where they introduced forward caching and reverse caching into the gateway to reduce query and reply traffic across the gateway.

The rest of the paper is organized as follows. We describe our proposed cooperation approaches for hybrid P2P file-sharing networks, i.e., shared-peer-based approach and server-chain-based approach, in Section 2. The details of cooperative mechanisms for a shared-peer-based approach are presented in Section 3. Next, in Section 4, we evaluate our proposed mechanisms for a shared-peer-based approach through simulation experiments. Finally, we conclude the paper and describe future works in Section 5.

2 Overview of Cooperation among Hybrid P2P File-Sharing Networks

In the context of the overlay network symbiosis, we assume that, intending to improve its application-level QoS, a peer or a meta-server introduces a cooperative program to achieve the cooperation among hybrid P2P file-sharing networks. We should note here that introduction of a cooperative program is determined by a peer or a meta-server itself, independently from the others. By a cooperative program, a P2P network, i.e., a peer or a meta-server can discover other P2P networks, decide whether P2P networks cooperate with each other or not, and cooperate.

In this section, we propose two cooperative approaches, that is, a Shared-Peer-Based (SPB) approach where a cooperative program is introduced into those peers that participate in multiple P2P networks, and a Server-Chain-Based (SCB) approach where a cooperative program is introduced into meta-servers. In the following subsections, we will briefly describe them.

2.1 Shared-Peer-Based Approach

In the SPB approach, a peer which participates in two or more P2P networks is called a shared peer. It becomes a point of cooperation, called a cooperative peer, by introducing a cooperative program (Fig. 2). Through cooperative peers, multiple P2P networks exchange their query and response messages among them. Then, peers in a P2P network can obtain files from other P2P networks. We hereafter call a P2P network from which a query message is originated as a guest network (network1, in Fig. 2) and a P2P network to which a query message is forwarded as a host network (network2, in Fig. 2).

A new cooperative peer, which newly introduced a cooperative program, first decides whether or not to initiate cooperation among P2P networks to which it belongs. The decision is made according to some criteria. We will describe details in Section 3.2.

In the SPB approach, a cooperative peer behaves not only as a peer, but also as a meta-server. When a meta-server disseminates a query message by flooding, the query message is also forwarded to a cooperative peer since it is regarded as one of neighboring meta-servers. On receiving a query message, a cooperative peer investigates its local cache. If meta-information about the desired file exists in its local cache, the cooperative peer generates a response message and sends it to the querying peer through intermediate meta-servers. Otherwise, after applying protocol con-

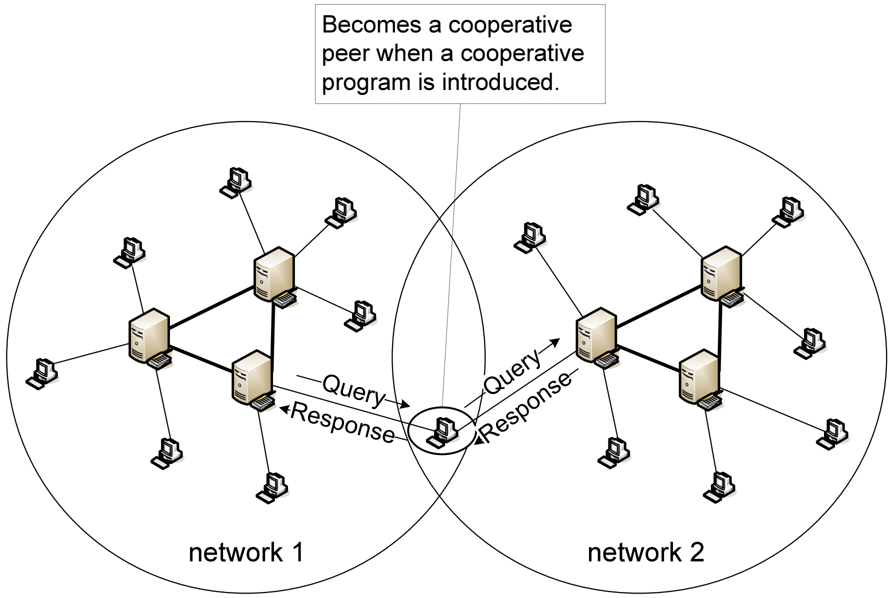

Figure 2: Shared-Peer-Based (SPB) Approach

version to the query message if needed, a cooperative peer transmits it to a meta-server in host P2P networks as shown in Fig. 2. A query message is disseminated in a host network as usual to find a desired file. When two P2P networks are connected by more than two cooperative peers, the same query message will be forwarded to a host network. To detect the duplication, each query message has a unique identifier. A meta-server discards redundant and duplicated query messages with the same identifier.

If a desired file is found in a host network, a response message is generated by a meta-server in a host network and it is sent back to the corresponding cooperative peer through a reverse path of the query message. Preparing for future query messages for the same or similar file, a cooperative peer deposits the meta-information in a response message into its local cache. After protocol conversion on necessity, a cooperative peer forwards a response message to its neighboring meta-server as a normal meta-server does. A response message eventually arrives at the peer which emitted the corresponding query message. It retrieves a desired file directly from a provider peer in a host network. If a host network employs a different protocol for file retrieval, a cooperative peer replaces information about a provider peer with itself in a response message. Then, a querying peer establishes a connection to a cooperative peer to retrieve a file. A cooperative peer obtains a file in place of a querying peer on request. Consequently, a peer can benefit from the cooperation

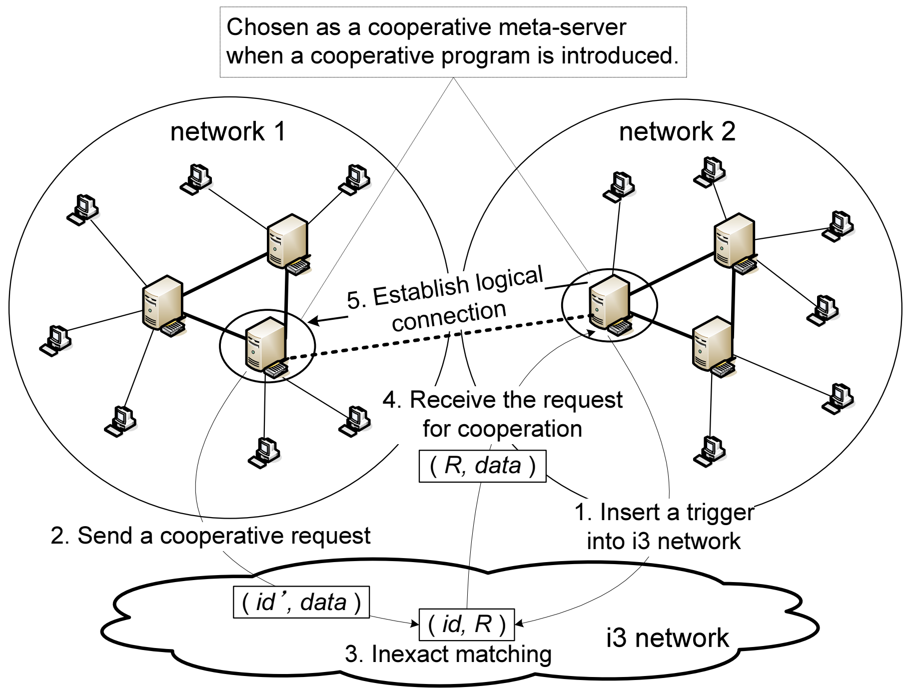

Figure 3: Server-Chain-Based (SCB) Approach

without recognizing it. We will describe details of cooperative mechanisms for this approach in Section 3.3.

2.2 Server-Chain-Based Approach

In the SCB approach, P2P networks cooperate with each other through logical connections established between meta-servers as shown in Fig. 3. The SCB approach is similar to a scheme proposed in [10] for cooperation among pure P2P file-sharing networks.

A meta-server which introduces a cooperative program becomes a candidate of cooperative meta-servers. One among them is chosen as a cooperative meta-server taking into account several criteria such as the number of peers connecting with it, the number of meta-information in its directory, and the distance to the other meta-servers in a P2P network. Then, a cooperative meta-server finds cooperative meta-servers of other P2P networks by using, for example, i3 [3, 4]. A logical connection is established between cooperative meta-servers to exchange query and response messages with each other.

After introducing a cooperative program into a meta-server, becoming a candidate of cooperative meta-server, it sends a query message with a special keyword to find other candidates. The special keyword is too long to hit in any normal meta-servers but in candidates of cooperative meta-server. On receiving the query message with a special keyword, the cooperative meta-server (if exist) and other candidates send back a response message which includes its own IP address, the number of peers connecting with it, the number of meta-information in its directory, the maximal delay to other meta-servers, and IP address of the current cooperative meta-server. When a current cooperative meta-server isn't being yet, all candidates take a vote to choose one taking into account the information within response messages received from all other candidates. To distribute the load on a cooperative meta-server, furthermore, when the current cooperative meta-server halts or leaves, a new one will be chosen by the same way described above.

A cooperative meta-server in another P2P network can be found by using i3. To establish an inter-meta-server connection, a cooperative meta-server emits a request message for cooperation to a cooperative meta-server in a host P2P network. On receiving the request message, a cooperative meta-server in a host P2P network decides whether or not to accept the request taking into account the benefit and cost. If it accepts the request, a logical connection is established between the cooperative meta-servers. Otherwise, it sends a response message to refuse the request.

In the SCB approach, a cooperative meta-server behaves not only as a meta-server in its own P2P network, but also as a peer against cooperative meta-servers in host networks. When a cooperative meta-server receives a query message and has no corresponding meta-information both in its local directory and in its local cache, it forwards the query message to all of its neighboring meta-servers in a P2P network by flooding. In addition, after applying protocol conversion if needed, it also sends the query message to cooperative meta-servers in host networks via logical links as a peer. The query message is treated as a normal query message in host P2P networks to find a desired file.

When the desired file is found in a host P2P network, a response message is generated by a meta-server in the host network and it is sent back to the cooperative meta-server in the guest network. Preparing for future query messages, a cooperative meta-server deposits the meta-information in a response message into its local cache. After applying protocol conversion and replacing information about a provider peer if needed, a response message is sent back to the querying peer following a reverse path of the query message. The peer obtains the desired file directly from a provider peer in a host network or with the mediation of a cooperative meta-server.

Thus, also in the SCB approach, the cooperation among hybrid P2P networks is achieved in a transparent way where other meta-servers and peers are unaware of the cooperation.

3 Details of Cooperation for a Shared-Peer-Based Approach

In this section, we describe details about a cooperative mechanism for the SPB approach described in Section 2.1.

3.1 Introduction of a Cooperative Program into Shared Peers

A shared peer introduces a cooperative program with the purpose of enhancing its own application-level QoS.

A user of a shared peer can find its desired files by choosing one or more appropriate P2P networks among those that it participates in, using corresponding P2P file-sharing applications, and submitting query messages for each of P2P networks. If a user fails in finding the file, other P2P networks are chosen and the same procedures are repeated. It apparently inefficient and troublesome. Once a shared peer becomes a cooperative peer by introducing a cooperative program, a user only needs to issue one query message on a P2P file-sharing application. The query message is automatically forwarded to multiple P2P networks that are worth searching by a cooperative program. Without recognizing which P2P networks are searched for the file, a user can find and obtain the desired file with a help of a cooperative program. In addition, the response time in finding a file decreases on a cooperative peer. To reduce the load of cooperation, a cooperative program deposits meta-information of response messages that passed the cooperative peer. If a query message matches any of meta-information in a local cache, there is no need of emitting a query message to find the file and the response message can be immediately obtained.

3.2 Decision of Cooperation

A new cooperative peer, which newly introduced a cooperative program, first decides whether or not to initiate cooperation among P2P networks to which it belongs. The decision is made according to some criteria such as the compatibility of protocols, the number meta-servers in each P2P networks, the size of P2P network, categories of files that peers are interested in and shared in P2P networks, and the number of cooperative peers in cooperative P2P networks.

If P2P networks offer files of the same or similar categories, peers can find and obtain more files by cooperation.

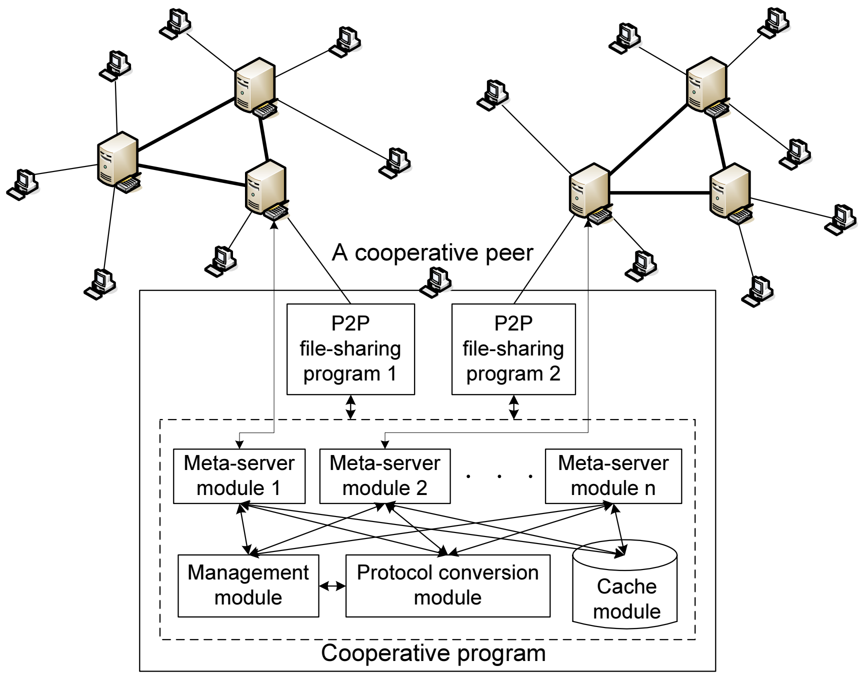

Figure 4: Modules Constituting a Cooperative Program for the SPB Approach

When P2P networks use different protocols for searching or retrieving files, the load on a cooperative peer increases for protocol conversion and relaying files.

If P2P networks are different in size, as will be shown later, a network with smaller number of peers benefits more in application-level QoS, while a network with larger number of peers cannot benefit very much. However, on the other hand, at the system-level, the newly introduced load from a smaller network to a larger network is not much.

The load on cooperative peers increases as the number of meta-servers increases. Therefore, the cooperation among P2P networks with a small number of meta-servers is desirable. On the other hand, the load on meta-servers increases with the increase of the number of cooperative peers while the load on cooperative peers does not change much. So the number of cooperative peers among cooperative P2P networks should be limited at a small number.

3.3 Structure of a Cooperative Program

Figure 4 illustrates components constituting a cooperative program: a management module, a protocol conversion module, meta-server modules, and a cache module.

3.3.1 Management Module

A management module allows a cooperative program to communicate with P2P file-sharing programs, manages the other modules, and decides whether or not to make P2P networks cooperate with each other.

Once a cooperative program is initiated, a management module detects P2P file-sharing programs working in a cooperative peer. Then it decides whether or not to perform cooperation taking into account several criteria described in 3.2. To start cooperation, a management module initiates meta-server modules corresponding to each of cooperative P2P networks.

If the protocol conversion is necessary for file retrieval, a querying peer sends a request to a corresponding P2P file-sharing application in a guest network. A management module intercepts the request, finds the information about the original provider peer in a cache, and then makes a P2P file-sharing program of a host network retrieve the desired file from the original provider peer. Finally, a management module makes a P2P file-sharing program of a guest network send the desired file to the querying peer.

3.3.2 Meta-Server Module

A meta-server module can perform some of meta-server's functions, including relaying query and response messages, accessing meta-information in a local cache through a cache module, and generating a response message. Each meta-server module connects to a meta-server in a corresponding P2P network.

When it receives a query message, a network from which the query came is considered as a guest network. A meta-server module for a guest network first investigates a local cache by forwarding the message to a cache module. If meta-information about the desired file exists in a local cache of a cooperative peer, it receives a list of provider peers from a cache module. Then, it generates a response message and sends the message to the querying peer through intermediate meta-servers. Otherwise, a meta-server module sends the query message to a protocol conversion module to forward it to host networks. A protocol conversion module manipulates a query message to fit to protocols used in host networks and sends it to meta-server modules of host networks. A query message is sent to a meta-server in a host network and disseminated by flooding. The flow of processing a query message corresponds to arrows 1 to 5 in Fig. 5.

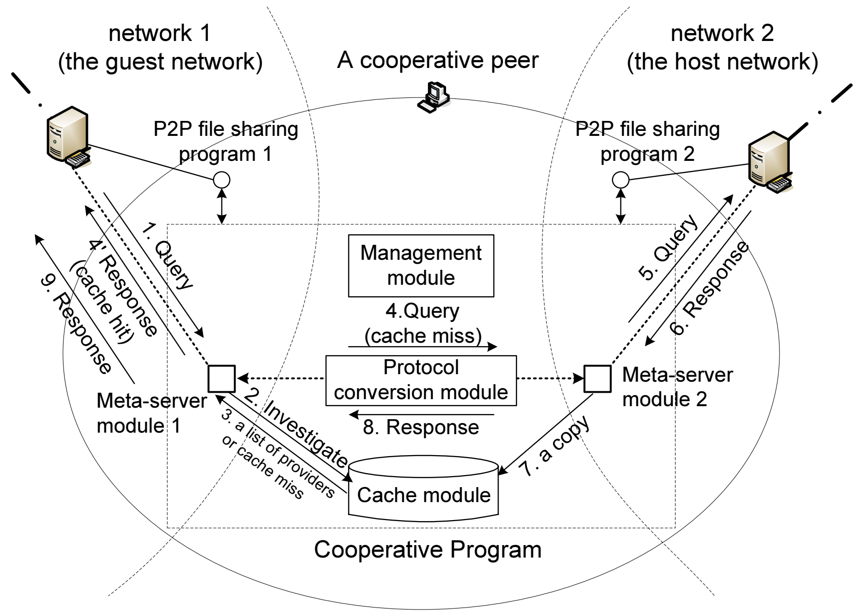

Figure 5: Processing Query and Response Messages

If the corresponding meta-information for a desired file is found on a meta-server in a host network, a response message is generated and it reaches the corresponding meta-server module for the host network through the same path, but in the reversed direction, that the query message traversed. A meta-server module forwards a copy of a response message to a cache module, and then relays the response message to a protocol conversion module. If needed, a protocol conversion module manipulates a response message and replaces a provider peer with itself. Then it forwards the response message to a meta-server module for a guest network. Finally, a meta-server module for a guest network sends a response message to a neighboring meta-server in the guest network so that it is relayed to a querying peer. These steps are shown in Fig. 5 by arrows 6 to 9.

3.3.3 Protocol Conversion Module

A protocol conversion module is used to manipulate a message to fit to a protocol used in a network which the message will be forwarded to.

When it receives a query message or a response message, it first evaluates the compatibility of protocols used in a host network and a guest network. If it considers the protocol conversion

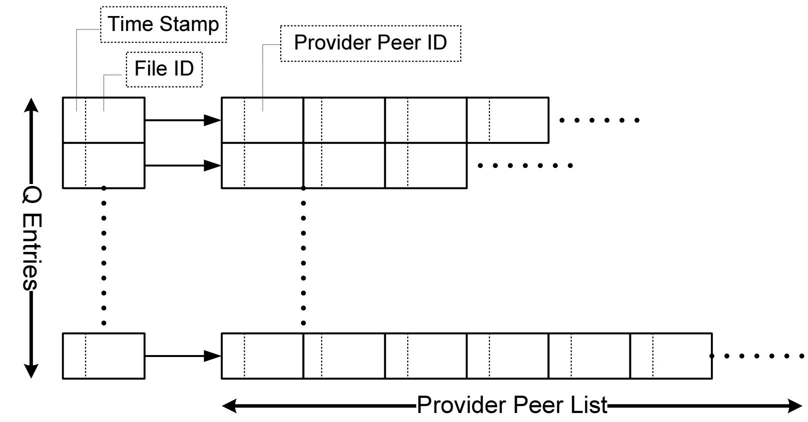

Figure 6: Structure of a Cache

is necessary, it takes out information of a query or response from a message. Then a protocol conversion module generates a new query message or response message in the form of the protocol used in a host or guest network, respectively. In the case that a protocol for file retrieval is different among P2P networks, it replaces the address of a provider peer in a response message with that of the cooperative peer for further processing of file retrieval as explained in 3.3.2. Finally, it forwards the new message to a corresponding meta-server module.

3.3.4 Cache Module

A cache module maintains a local cache buffer at a cooperative peer. Each entry of a cache consists of a file identifier field with fixed length and a provider peer list with variable length as illustrated in Fig. 6. The file identifier field is used as indexes of cache entries. Each of entries and each of provider peers in a list has a timestamp.

When a cache module receives a query message from a meta-server module, it investigates a cache buffer to find a desired file. If there exists corresponding meta-information, it makes a list of provider peers and send the list back to the meta-server module. At the same time, a timestamp of the entry is updated by the current time. Otherwise, it notifies the meta-server module of a cache miss.

On receiving a response message from a meta-server module, it investigates its local cache. If a file identifier in the response message exists in its local cache, a meta-server module first updates the timestamp of corresponding cache entry. Then, it updates the timestamp of the provider peer in a provider peer list if there, or adds information of new provider peers into the list with a timestamp of the current time. Otherwise, a cache module makes a new entry to deposit new meta-information into its cache. When the cache reaches maximum capacity and there is no room for a new entry, according to LRU replacement policy, the entry with the oldest timestamp is replaced with the new one. We should note here that the maximum capacity of a cache is determined by a user itself independently from others.

4 Evaluation of Cooperation

In this section, we conduct simulation experiments to evaluate our proposed cooperative mechanism for the SPB approach from viewpoints of application-level and system-level performance.

4.1 Simulation Model

Referring to KaZaA's topology [13] [14], we simulate a hybrid P2P file-sharing network by following steps. First, m meta-servers and n peers are randomly placed in a two-dimensional region. Next, a randomly chosen meta-server is connected to the closest meta-server to construct an initial seed of a meta-server network. Then, a meta-server is randomly chosen one by one and is connected to the closest meta-server in a meta-server network. Finally, peers are connected to the closest meta-server. An example of a generated hybrid P2P network is shown in Fig. 7, where squares correspond to meta-servers and dots stand for normal peers. In our simulation, two hybrid P2P networks are generated in the same manner.

When we consider cooperative P2P networks, c cooperative peers are randomly placed in the two-dimensional region. Each cooperative peer is connected to two meta-servers each of which is the closest to the peer in each of P2P networks. In order to keep the number of peers, c peers are randomly chosen and removed in each network.

F kinds of files are available in two networks. Each file has an identifier f_r $(1 \leq r \leq F)$, where f_F is for the most popular file and f_1 is for the least popular one. The popularity of files follows a Zipf distribution with $\alpha = 1.0$. The number of each file existing in networks also follows a Zipf distribution with $\alpha = 1.0$, where the number of the least popular file is 1 and the number of the most popular file is F. As an example, the number of files of each kind of files against popularity of files is shown in Fig. 8 where the number of kinds of files F is 500. Files are assigned to peers at random. Peers register meta-information about assigned files to its designated meta-server.

We conducted simulation experiments based on the query-cycle model [15]. Peers generate query messages following the poisson process whose rate λ is randomly chosen from 0 to 0.5 at the uniform distribution. It means that the probability that a peer issues x query messages in one query cycle becomes $p(x) = \frac{e^{-\lambda} \cdot \lambda^x}{x!}$. The probability that a peer generates query message q_r for file f_r is given by the popularity of file f_r. A peer does not issue a query message for a file that it already owns.

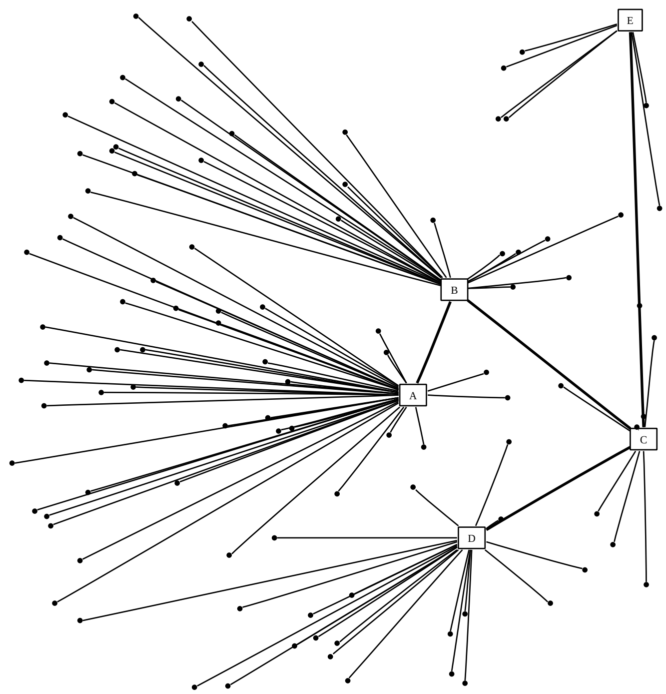

Figure 7: An Example for Hybrid P2P File-Sharing Network Topology ($m = 5, n = 100$)

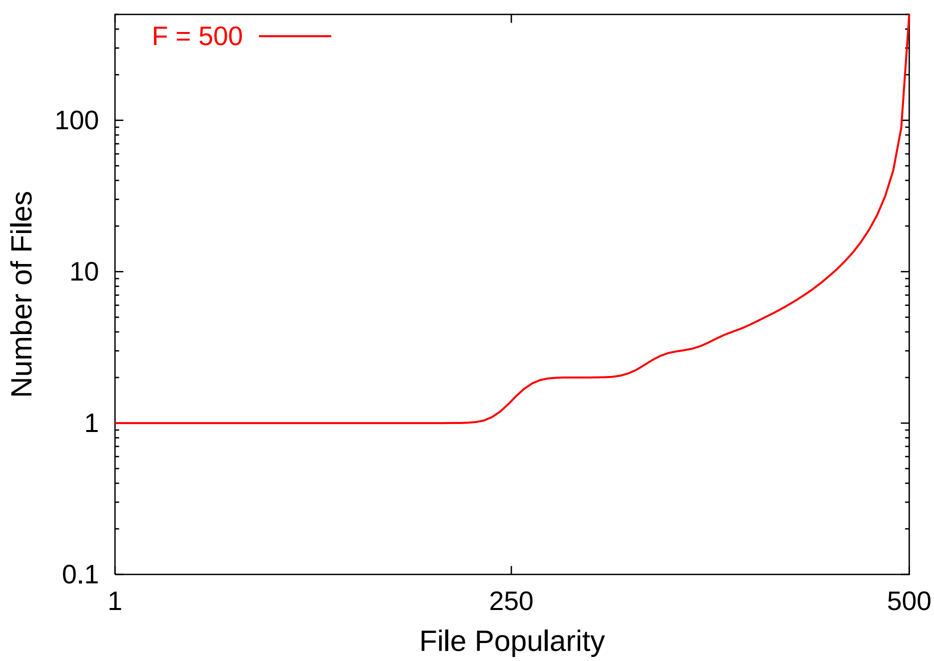

Figure 8: Number of Files against Popularity of Files ($F = 500$)

In our simulation experiments, we assume that two networks use the same protocol. We conducted 100 set of simulations of 20 query cycles. Only results obtained after all caches are filled with entries are used and their averaged values are shown in the following results. We should note here that a TTL value for a query message is not defined as in a pure P2P file-sharing application, since the most of hybrid P2P file-sharing application do not use a TTL mechanism. A meta-server always forwards a query message to all of its neighboring meta-servers except one that the query came from, unless it has the corresponding meta-information for the query. It means that a query message can find a file as long as it exists in a P2P network.

4.2 Evaluation of Application-Level QoS of Normal Peers

As application-level performance measures of normal peers, we use the file availability, the hit ratio, the number of provider peers, and the response time.

4.2.1 File Availability and Hit Ratio

When two P2P networks cooperate with each other, peers can find files not only in their own network but also in the other. The file availability is defined as the ratio of the number of kinds of files in one network to the total number of kinds of files available in two networks, that is, F. We also define the hit ratio as the ratio of the number of query messages whose desired files are found to the total number of query messages. By the definition, the hit ratio is one when two P2P networks cooperate. In the following results, we assume that 0.25 kinds of files are shared in every peer on average.

Table 1 summarizes the file availability and the hit ratio for two P2P networks, i.e., network1 and network2, of the same number of peers. In the table, $n1 : n2$ stands for the number of peers in network1 and network2, respectively. We find that the file availability of an independent network is only about 68–70%, and it increases by about 30% when they cooperate with each other. Furthermore, the hit ratio also increases by cooperation regardless of the network size and the degree of increase is higher with smaller networks.

In table 2, we set the number of peers in network2 at 1000 while setting the number of peers in network1 at 100 and 500, respectively. It is shown that the effect is higher for cooperation among imbalance networks, and a network with less number of peers benefits more. For example, the file

Table 1: File Availability and Hit Ratio (same number of peers)

		File Availability	Hit Ratio
100:100	network1	69.9%	89.6%
	network2	68.4%	89.1%
1000:1000	network1	69.2%	93.5%
	network2	69.4%	93.5%
10000:10000	network1	69.0%	95.3%
	network2	69.7%	95.4%

Table 2: File Availability and Hit Ratio (different number of peers)

		File Availability	Hit Ratio
100:1000	network1	23.9%	69.3%
	network2	95.2%	99.1%
500:1000	network1	54.9%	88.4%
	network2	81.0%	96.1%

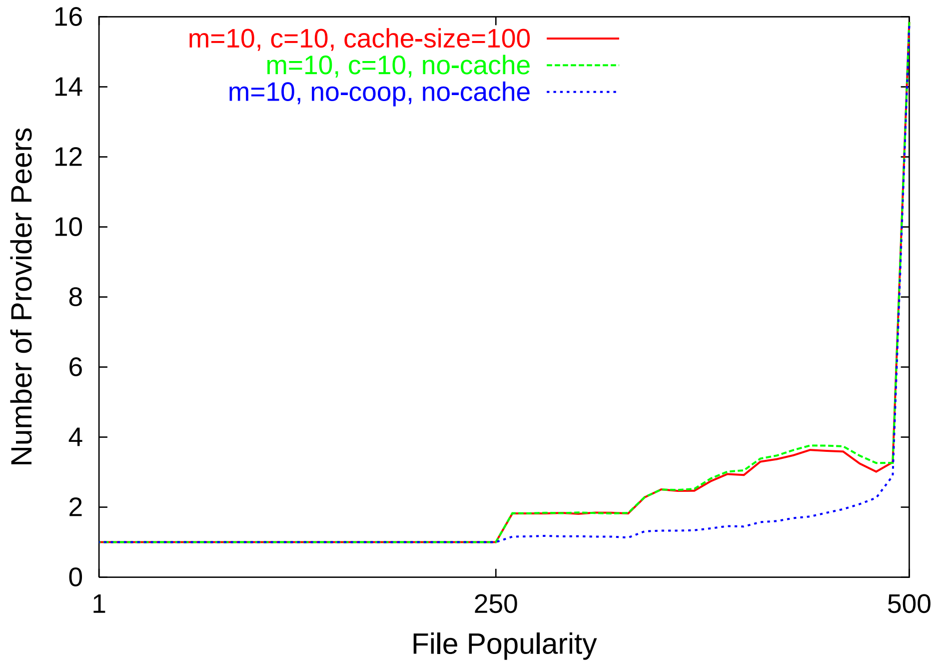

Figure 9: Number of Provider Peers against Popularity of Files

availability of network1 of 100 peers increases by about 76% when it cooperates with network2, i.e., a ten times larger network.

4.2.2 Number of Provider Peers

The number of provider peers of successful search is calculated against the popularity of files and the time. The former is defined as the average number of found provider peers among successful searches for files of the same popularity. The latter is defined as the average of the cumulative number of found provider peers in one time unit, where propagation delay of a logical link is assumed identical and 0.5 time unit. In the following evaluations, we use values as the number of peers n is 1000 and the number of kinds of files F is 500.

Figure 9 shows the number of provider peers of successful searches against the file popularity. In the figure, m stands for the number of meta-servers and c stands for the number of cooperative peers. "no-coop" means that two P2P networks do not cooperate with each other and "no-cache" means that a cooperative peer does not deposit meta-information into its local cache. "cache-size = 100" stands for a cooperative peer has a cache of the capacity of 100 entries. It is shown that, for

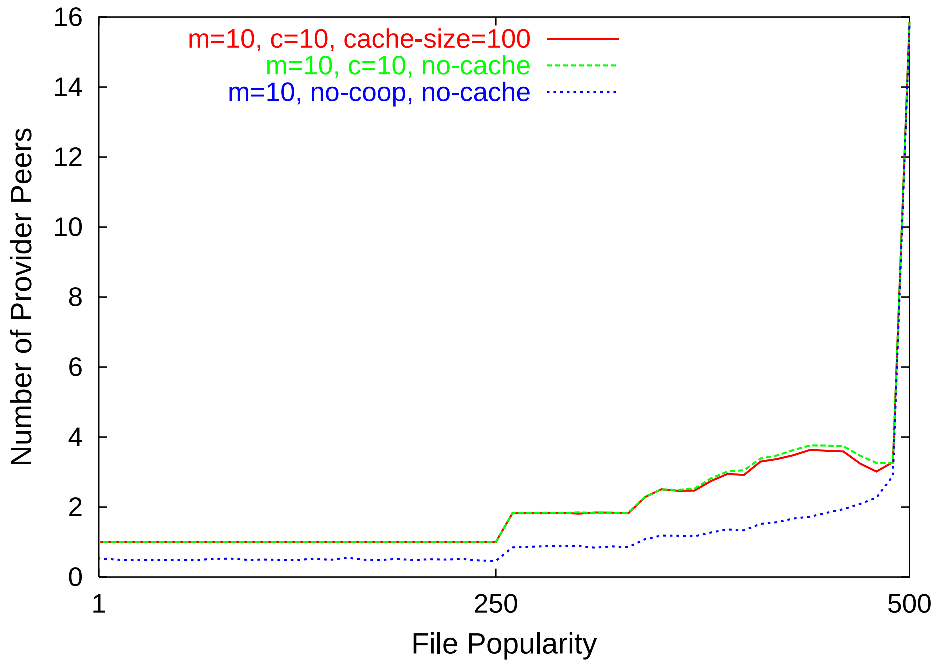

Figure 10: Expected Number of Provider Peers against Popularity of Files

unpopular files, only one provider peer is found. When two P2P networks cooperate, the number of provider peers found is increased for files with high and moderate popularity as shown in the figure. However, introducing a cache slightly decreases the number of provider peers, whereas theoretically, an entry in a cache has a list of all provider peers that can be found by a search. The reason is as follows. When a meta-server in a host network directly connected to a cooperative peer cannot answer a query received from the cooperative peer, the query message is forwarded to other meta-servers. In such a case, response messages arrive at the cooperative peer one by one and a corresponding cache entry gradually grows as response messages arrive. Since a cooperative peer answers a query message without forwarding it to the other meta-server when there is any matching entry in its cache as normal meta-servers do, a querying peer whose query message arrives at a cooperative peer during the growth of the entry only receives a partial list of provider peers.

When we take into account failed searches by setting their number of found provider peers as zero, the expected number of provider peers per search becomes as shown in Fig. 10. The figure indicates that the cooperation among P2P networks contributes to increasing the number of

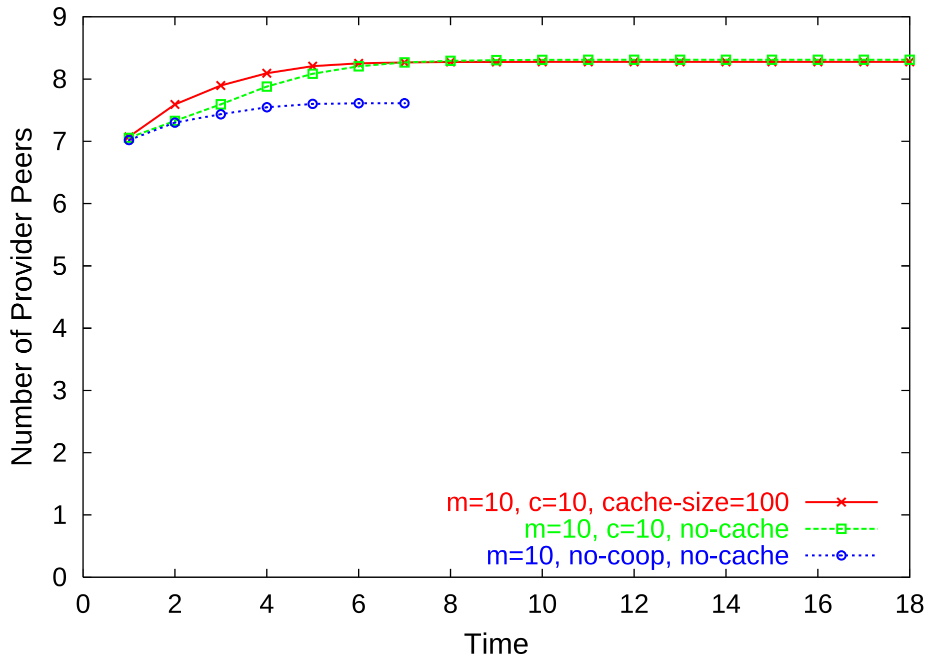

Figure 11: Distribution of Number of Provider Peers against Time

provider peers independently of the existence of a cache.

The expected number of provider peers found by a query message increases as time passes as shown in Fig. 11. However, the difference between mechanisms with and without caching is small. This is because that query messages for highly popular files find desired meta-information at the designated meta-server or before reaching cooperative peers. Therefore, only query messages for moderately popular or unpopular files reach cooperative peers. Since the distribution of the number of files existing in P2P networks follows a Zipf distribution, the number of provider peers of those files is small. Consequently, the difference in the number of provider peers between mechanisms with and without caching is small. From Fig. 9 through Fig. 11, we conclude that caching at cooperative peers do not improve the application-level QoS in terms of the number of provider peers, although cooperation itself is beneficial for peers.

4.2.3 Response Time

The response time is defined as the duration between the emission of a query message and the reception of the first response message, where propagation delay of a logical link is assumed

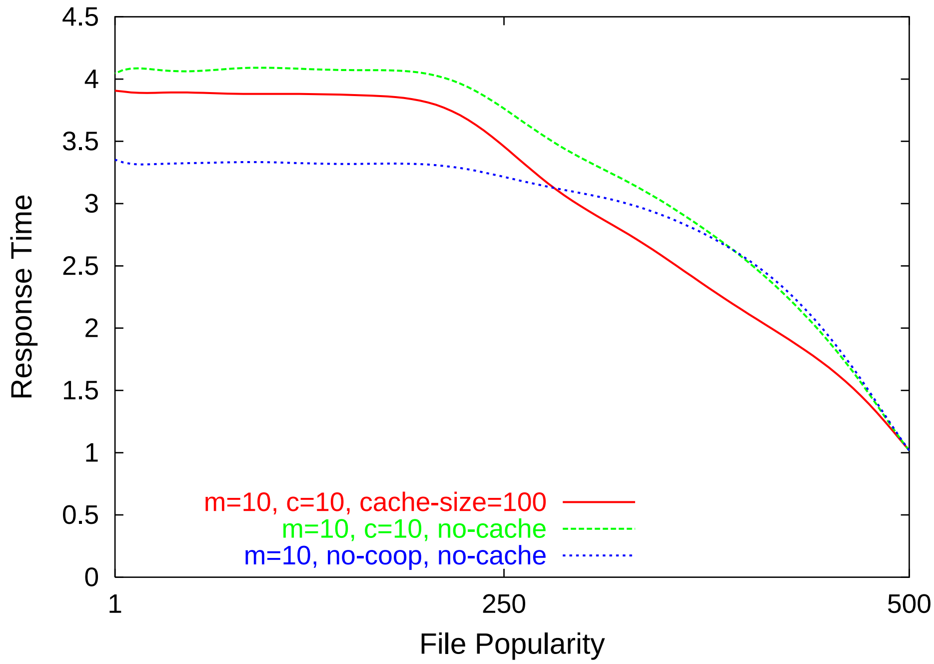

Figure 12: Response Time against Popularity of Files (normal peer)

identical and 0.5 time unit. In the following results, we use values as the number of peers n is 1000 and the number of kinds of files F is 500.

Figure 12 shows the response time of successful searches against the file popularity. The reason that the result of the no-cooperative case is lower than the others is that we only consider the delay of successful search in the figure. When P2P networks are independent, a peer can find only about 70% of files available in two P2P networks due to the placement of files. On the other hand, because of a query flooding mechanism in a hybrid P2P network, a peer can find all files in the system when two P2P networks cooperate with each other. However, the response time increases to explore a cooperating P2P network for unpopular files. As shown in the figure, the response time is slightly improved by a cache especially for files with moderate and low-lying popularity.

4.3 Evaluation of Application-Level QoS of Cooperative Peers

In this section, as application-level performance measures of cooperative peers, we evaluate our proposed cooperative mechanisms for the SPB approach in terms of the cache hit ratio of cooper-

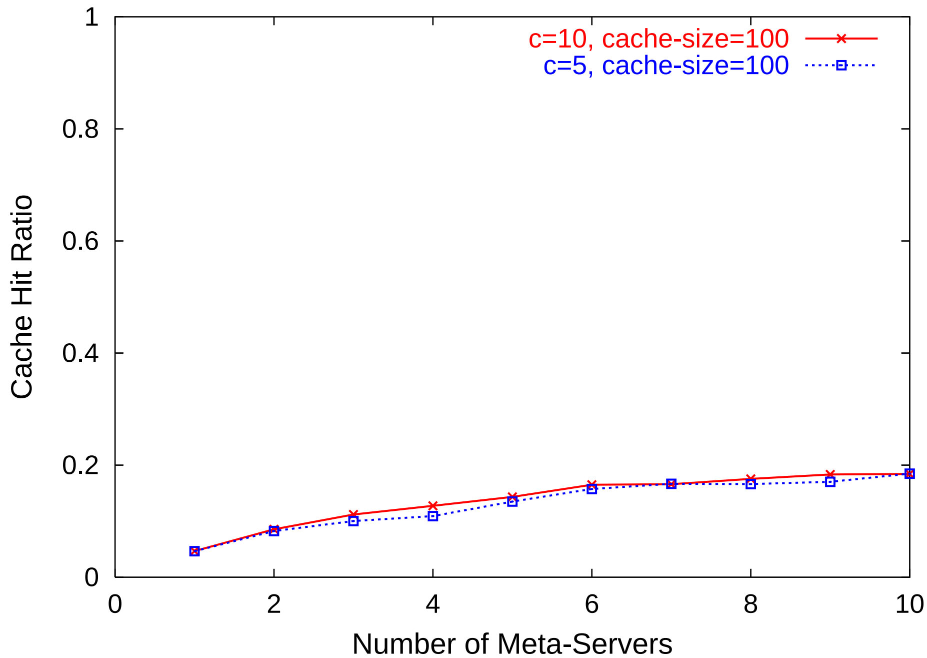

Figure 13: Cache Hit Ratio of Cooperative Peers

ative peers and the response time.

4.3.1 Cache Hit Ratio of Cooperative Peers

The cache hit ratio of cooperative peers is defined as the ratio of the number of query messages whose desired meta-information are found in its local cache to the total number of query messages, where only query messages generated by cooperative peers are taken into account.

Figure 13 shows the cache hit ratio of cooperative peers where we set the number of peers n at 1000 while changing the number of meta-servers m from 1 to 10 in both networks. The number of kinds of files F is 500. It is shown that the cache hit ratio of cooperative peers increases with the increase of the number of meta-servers independently of the number of cooperative peers. The reasons are as follows. In our simulation, the number of peers and files is fixed while changing the number of meta-servers. Therefore, as the number of meta-servers increases, the number of peers connecting with each meta-server decreases. Then, the number of meta-information registered at each meta-server decreases. Consequently, the probability that a query message matches corresponding meta-information at the designated meta-server decreases (see table 3) and more query

Table 3: Hit Ratio at the Designated Meta-Server

Number of Meta-Servers	Hit Ratio
m=1	93.5%
m=2	86.1%
m=5	74.7%
m=10	66.2%

Table 4: Comparison of Average Response Time of Normal Peers and Cooperative Peers

Number of Meta-Servers	Normal Peer	Cooperative Peer
m=1	1.11	0.95
m=2	1.21	0.98
m=5	1.69	1.23
m=10	2.10	1.41

messages are forwarded to neighboring meta-servers and cooperative peers.

Figure 14 shows the cache hit ratio of cooperative peers against popularity of files. As the number of meta-servers increases, the popularity of a file with the highest hit ratio increases. It is because that query messages for popular files do not necessarily match the repository of a designated meta-server with larger number of meta-servers. Consequently, more query messages for more popular files are forwarded to cooperative peers and the popularity of meta-information cached in cooperative peers increases. Since the number of query messages for popular files is more than that of unpopular ones, the cache hit ratio of cooperative peers increases.

4.3.2 Response Time

Figure 15 shows the response time that a cooperative peer experiences against the file popularity where we set the number of peers *n* at 1000 in both P2P networks. It is shown that the response time is improved by a cache especially for files with moderate popularity.

Table 4 summarizes the average response time of normal peers and cooperative peers, where

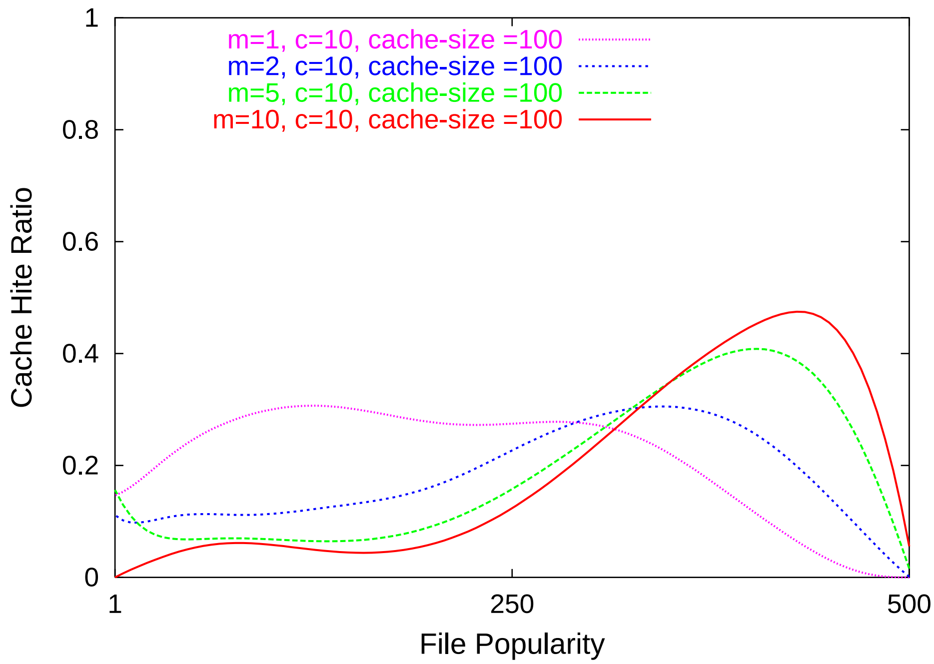

Figure 14: Cache Hit Ratio of Cooperative Peers against Popularity of Files

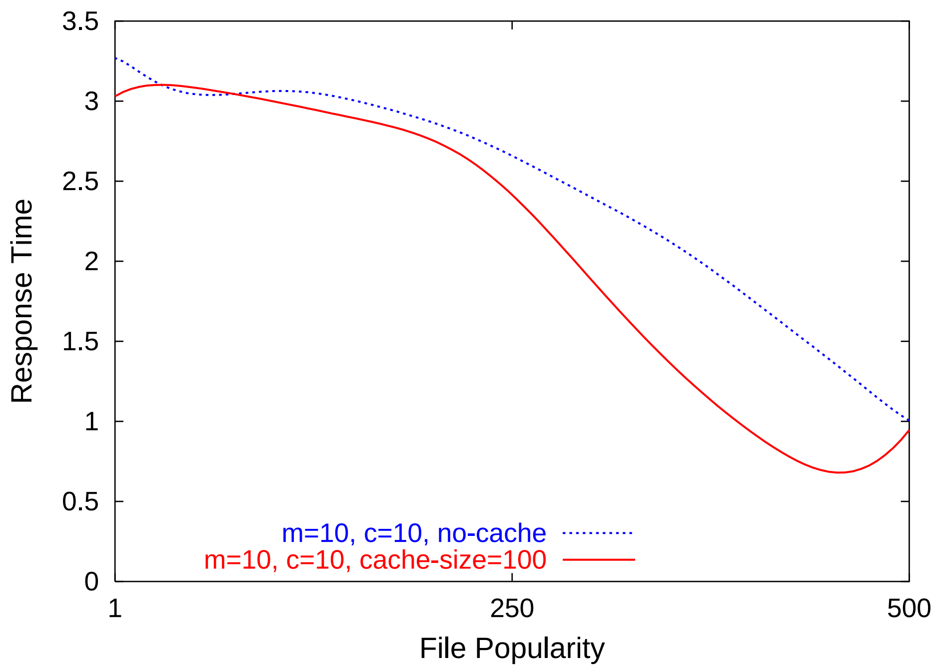

Figure 15: Response Time against Popularity of Files (cooperative peer)

the number of cooperative peers c is 10. The average response time of normal peers is derived by $\sum p_i h_i$, where p_i stands for the probability that a peer generates a query message for f_i and the average response time for file f_i of normal peers is h_i. On the other hand, the average response time of cooperative peers is given by $\sum p_i(1-a_i)k_i$, where a_i stands for the cache hit ratio for file f_i at cooperative peers and k_i stands for the average response time that cooperative peers experience without caching. It is shown that the average response time of cooperative peers is smaller than normal peers by 15–33%.

4.4 Evaluation of System-Level Behaviors

As system-level measures, we use the load of cooperative peers and meta-servers to evaluate the load introduced by cooperation. We define the load on cooperative peers as the average number of messages that a cooperative peer received and sent. Analogously, the load on meta-servers is defined as the average number of query messages that a meta-server received and sent and response messages that a meta-server generated, received, and sent.

4.4.1 Load on Meta-Servers against the Number of Cooperative Peers

In this section, we evaluate the influence of number of cooperative peers on the load of meta-servers.

Figure 16 shows the average load on meta-servers in network1 where the number of cooperative peers is changed from 0 to 10. The number of peers n is 1000, and F is 500. The number of cooperative peers 0 means that two P2P networks are uncooperative. It is shown that the load on meta-servers almost linearly increases with the increase of the number of cooperative peers independently of the existence of caching. This is because that all cooperative peers relay the query messages independently of others. Consequently, the same query messages are injected into a host P2P network through multiple cooperative peers. This increases the load on meta-servers.

We also find that the cache system introduced into cooperative peers does not reduce the load on meta-servers very much. The reasons are as follows. A query message will be disseminated among meta-servers if the meta-server to which a querying peer is connected does not have meta-information of the desired file. They eventually reach cooperative peers in a guest network. Even if there is a cache hit at any of them, query messages are forwarded to meta-servers in a host network

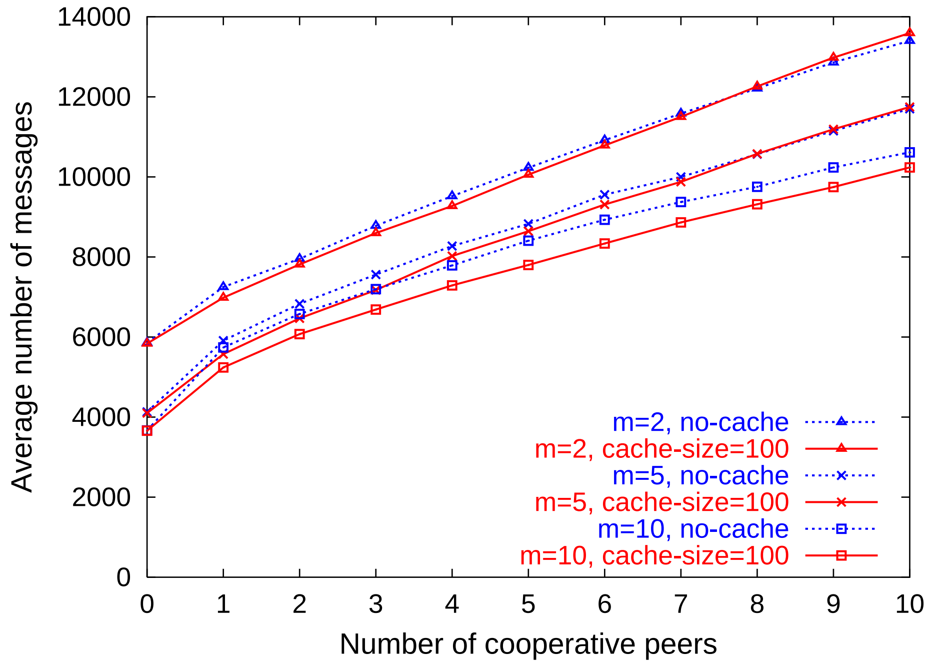

Figure 16: Load on Meta-Servers against the Number of Cooperative Peers

by the other cooperative peers. This case often occurs for unpopular files. Another reason is the load of response messages. If a desired file is popular or moderately popular, many cooperative peers generate response messages due to high cache hit ratio. Since response messages for the same query message are not discarded at meta-servers, as a result, the number of response messages increases than in the case of no-cache cooperations. This trend becomes obvious especially when the number of cooperative peers is larger than meta-servers. The other reason is the load of useless response messages. Since the same cache buffer is used for two P2P networks at a cooperative peer, it contains information of both of a guest network and a host network. Therefore, the probability that a query message finds the meta-information at a cooperative peer, which is also found at meta-severs in a guest network, increases as the number of cooperative peers increases. These response messages are redundant.

4.4.2 Load on Cooperative Peers against the Number of Cooperative Peers

In this section, we evaluate the influence of number of cooperative peers on the load of cooperative peers.

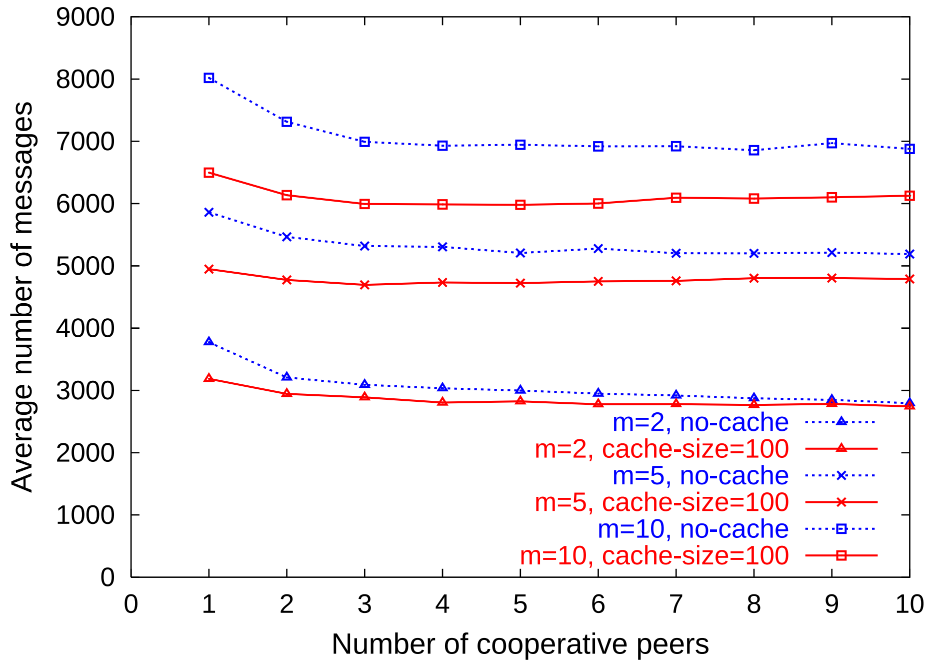

Figure 17: Load on Cooperative Peers against the Number of Cooperative Peers

Figure 17 shows the average load on cooperative peers where the number of cooperative peers is changed from 1 to 10. The number of peers n is 1000, and F is 500. It is shown that, as the number of cooperative peers increases, the load on cooperative peers does not change much independently of the existence of caching. This is because that all cooperative peers relay the query messages independently of others. In addition, when a query message hits a cache of a cooperative peer, the cooperative peer does not forward the message to a host network. Then, response messages which the cooperative peer has to relay from a host network to a guest network can be suppressed. It means that the load on cooperative peers decreases.

However, we find that the decrease in the load on cooperative peers is limited in Fig. 17, especially in the case that the number of meta-servers is far less than the number of cooperative peers. Now, consider the case that a meta-server in a host network has multiple cooperative peers with a guest network. Even if only one cooperative peer among them cannot answer a query, a query message is forwarded to the meta-server. Since the meta-server considers cooperative peers as neighboring meta-servers, it forwards the query message to all cooperative peers except one that the query originated from. They have already processed the query and, thus, such forwarding

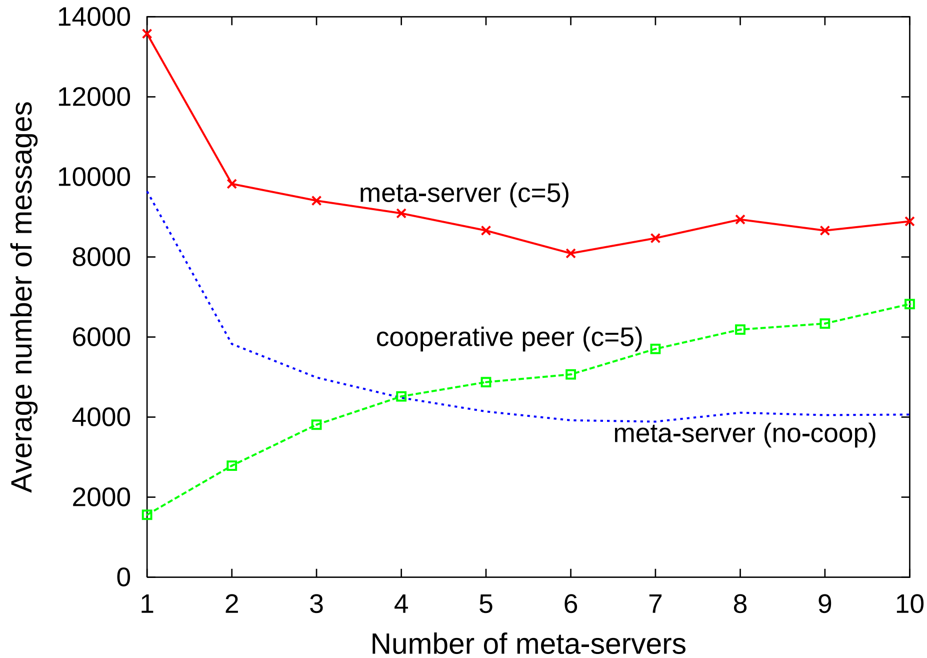

Figure 18: Influence of Number of Meta-servers on Load

is redundant and only wastes the network and system resources.

4.4.3 Influence of Number of Meta-servers on Load

In this section, we evaluate the influence of number of meta-servers on the load of cooperative peers and meta-servers. We use values as the number of peers n is 1000, and F is 500. A cooperative peer does not deposit meta-information into its local cache.

In Fig. 18, the number of meta-servers in each P2P networks changes from 1 to 10. It is shown that the load on meta-servers decreases with the increase of the number of meta-servers and it is the same for both in cooperative and non-cooperative networks. On the other hand, the load on cooperative peers increases as the number of meta-servers increases. When the number of meta-servers increases, the number of peers per meta-server decreases. Thus, the amount of meta-information deposited in a meta-server decreases. Consequently, the probability that query message does not match any meta-information at the designated meta-server increases. Then, query messages are forwarded to neighboring meta-servers and the number of query messages that a cooperative peer receives increases. Therefore, from a view point of the load on cooperative

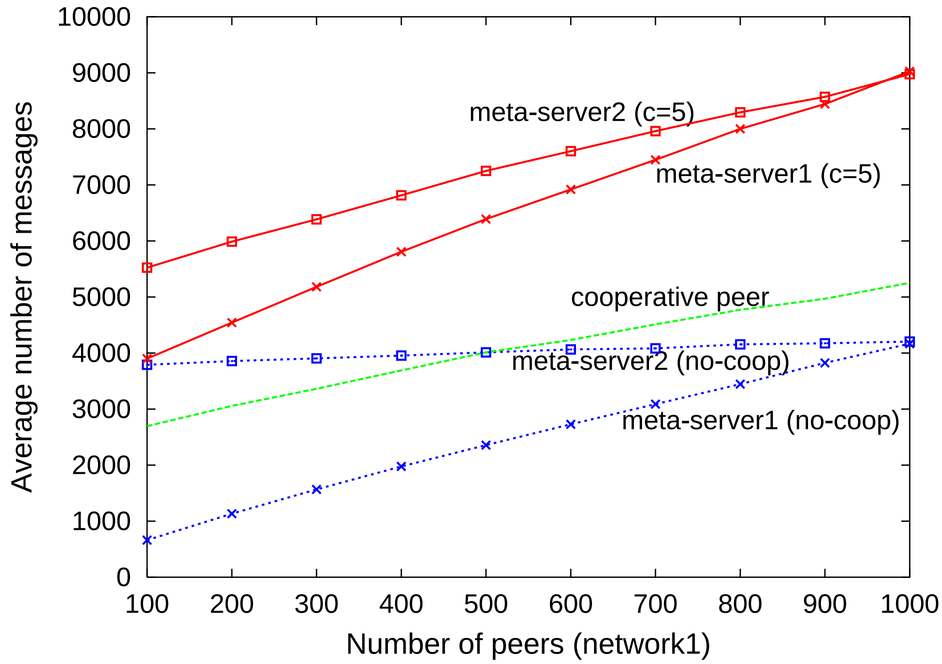

Figure 19: Influence of Size of P2P Network on Load

peers, which is usually less powerful than meta-servers, the cooperation among P2P networks with a small number of meta-servers is desirable.

4.4.4 Influence of Size of P2P Network on Load

In this section, we evaluate the influence of number of peers on the load of cooperative peers and meta-servers. We use values as the number of meta-servers *m* is 5, and assume that 0.25 kinds of files are shared in every peer on average. A cooperative peer does not deposit meta-information into its local cache.

Figure 19 depicts variations of load with changes in the number of peers in network1 from 100 to 1000. The number of peers in network2 is fixed at 1000. It is obvious that the load on both cooperative peers and meta-servers increases as the number of peers of network1 increases. We also find that the difference in load on meta-servers becomes small with the decrease of difference in size among P2P networks.

To compare the increase of load on meta-servers in networks of different size, Fig. 20 shows the normalized load, derived as the ratio of the number of messages after cooperation to that

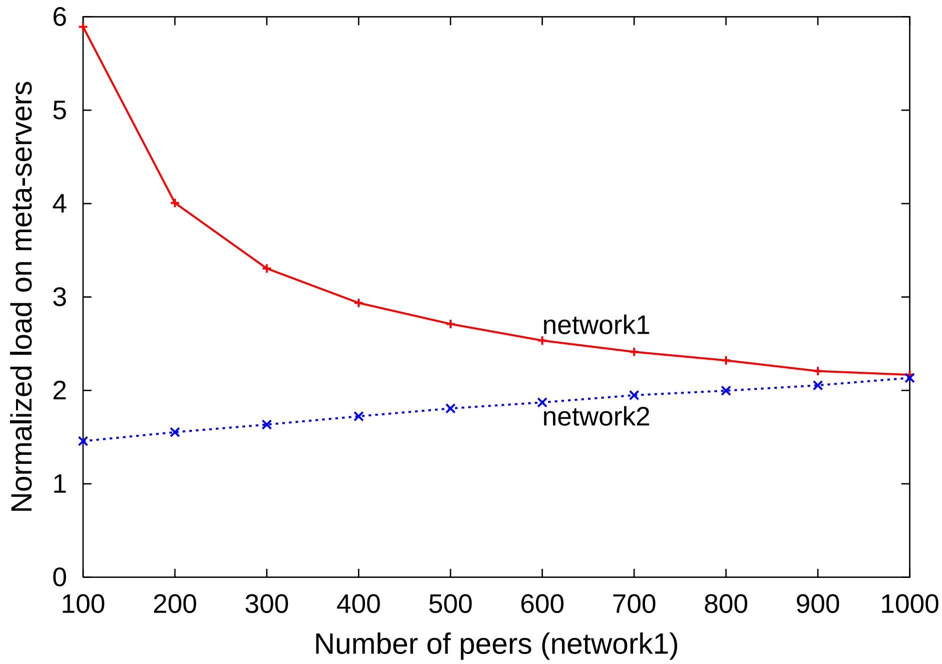

Figure 20: Normalized Load on Meta-servers

before cooperation. It can be seen that the load on meta-servers in network1 increases more than that in network2, since more query messages are forwarded from network2 into network1. On the other hand, table 2 indicates that the benefit of network2, i.e., a larger network in cooperation is smaller than that of network1. It means that the most of query messages injected into network1 by network2 are redundant and meaningless.

5 Conclusions and Future Work

In this paper, we proposed two cooperative approaches for hybrid P2P file-sharing networks by which two or more hybrid P2P file-sharing networks can efficiently cooperate with each other to improve their collective application-level QoS.

By introducing a cooperative program into a shared peer or meta-servers, P2P networks can discover each other and decide whether or not to cooperate with each other taking into account the benefit and cost. A cooperative peer or a cooperative meta-server behaves not only as a peer, but also as a meta-server, so that cooperative P2P networks can exchange query and response messages through it. Furthermore, when a desired file is found in a host P2P network, a peer can obtain the desired file directly from a provider peer in a host network or with the mediation of a cooperative peer or a cooperative meta-server.

Through simulation experiments, we have shown that our proposed cooperative mechanisms for the Shared-Peer-Based approach can improve the application-level QoS in terms of file availability, the hit ratio, and the number of provider peers, at the sacrifice of the increased load on meta-servers and cooperative peers. We also investigated the influence of network configurations such as the number of peers and meta-servers. Although slight improvements are observed for the response time and the load on cooperative peers, the cache system introduced into cooperative peers does not help much to improve the number of provider peers and reduce the load on meta-servers. It should be noticed that the capacity of a cache in our simulation experiments amounted to one-fifth of the total number of kinds of files available in P2P networks. Considering the fact that there are about 100,000,000 files shared in a P2P network [16], such assumption is unrealistic. However, our results also indicated that the load on cooperative peers increased as the number of meta-servers increased. Therefore, when P2P networks with a large number of meta-servers cooperate with each other, introducing caching into cooperative peers could be helpful to decrease the load on cooperative peers to some extent. As the benefit of cooperative peers, the response time of cooperative peers is smaller than normal peers by 15–33%.

As several ongoing researches, we plan to investigate behaviors of cooperation among dynamic P2P networks, where peers dynamically join/leave and their topologies dynamically change as consequences of cooperation. Furthermore we should evaluate the influence of cooperation among P2P networks on physical networks, and investigate a cooperative mechanism which takes into account characteristics of physical networks.

Acknowledgements

I would like to express my deepest gratitude to my supervisor Professor Masayuki Murata of Osaka University, who introduced me to the area of computer networks including the subjects in this thesis and support through my studies. His excellent guidance and thoughtful advice have provided an invaluable experience that will help me in my career.

All works of this thesis would not have been possible without the helpful suggestions and support of Associate Professor Naoki Wakamiya of Osaka University. He has opened the way for the accomplishment of this thesis.

I am most grateful to Professor Koso Murakami, Professor Makoto Imase, Professor Teruo Higashino, Professor Hirotaka Nakano, and Professor Tetsuji Satoh of Osaka University for their appropriate guidance and invaluable firsthand advice.

I am also indebted to Associate Professor Go Hasegawa, Associate Professor Hideki Tode, Associate Professor Akio Nakata, Associate Professor Hiroyuki Ohsaki, Assistant Professor Ichinoshin Maki, Assistant Professor Shin'ichi Arakawa, and Assistant Professor Masahiro Sasabe of Osaka University who gave me helpful comments and feedbacks. Furthermore, I thank all my colleagues in the Advanced Network Architecture Research Group for their generous help, enlightening and valuable suggestions. Especially I received much useful advices from Junjiro Konishi and wish to thank him.

And this research was supported in part by "New Information Technologies for Building a Networked Symbiosis Environment" in The 21st Century Center of Excellence Program of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

Finally, I wish to express my warmest thanks to my wife and my parents for their unconditional love, support, patience and understanding.

References

[1] M. Seshadri and R. H. Katz, "Dynamics of Simultaneous Overlay Network Routing," Tech. Rep. UCB//CSD-03-1291, EECS Department, University of California, Berkeley, November 2003.

[2] A. Nakao, L. Peterson, and A. Bavier, "A Routing Underlay for Overlay Networks," in *Proceedings of ACM SIGCOMM*, pp. 11–18, August 2003.

[3] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet Indirection Infrastructure," in *Proceedings of ACM SIGCOMM*, pp. 78–88, August 2002.

[4] "Internet Indirection Infrastructure (i3) Homepage." available at `http://i3.cs.berkeley.edu/`.

[5] M. Kwon and S. Fahmy, "Toward Cooperative Inter-overlay Networking," in *Proceedings of IEEE ICNP*, November 2003. Poster.

[6] M. Kwon and S. Fahmy, "Synergy: An Overlay Internetworking Architecture and its Implementation," tech. rep., Purdue University, October 2005.

[7] T. Yomo, W.-Z. Xu, and I. Urabe, "Mathematical Model Allowing the Coexistence of Closely Related Competitors at the Initial Stage of Evolution," *Researches on Population Ecology*, vol. 38, no. 2, pp. 239–247, 1996.

[8] N. Wakamiya and M. Murata, "Toward Overlay Network Symbiosis," in *Proceedings of the Fifth International Peer-to-Peer conference (P2P2005)*, pp. 154–155, August–September 2005.

[9] H. Fu, N. Wakamiya, and M. Murata, "Proposal and Evaluation of a Cooperative Mechanism for Hybrid P2P File-Sharing Networks," in *Proceedings of The Fourth IASTED International Conference on Communications, Internet and Information Technology (CIIT2005)*, pp. 7–13, October–November 2005.

[10] J. Konishi, N. Wakamiya, and M. Murata, "Proposal and Evaluation of a Cooperative Mechanism for Pure P2P File Sharing Networks," in *Proceedings of the 2nd International Workshop on Biologically Inspired Approaches to Advanced Information Technology (Bio-ADIT2006)*, pp. 33–47, January 2006.

[11] R. Schollmeier, "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications," in *Proceedings of the First International Conference on Peer-to-Peer Computing (P2P2001)*, pp. 101–102, August 2001.

[12] A. R. Butt, N. Malhotra, S. Patro, and Y. C. Hu, "On the Equivalence of Forward and Reverse Query Caching in Peer-to-Peer Overlay Networks," in *Proceedings of the 9th International Workshop on Web Content Caching and Distribution (WCW'04)*, pp. 169–181, October 2004.

[13] "KaZaA Homepage." available at `http://www.kazaa.com`.

[14] J. Liang, R. Kumar, and K. W. Ross, “Understanding KaZaA.” available at http://cis.poly.edu/~ross/papers/UnderstandingKaZaA.pdf.

[15] M. T. Schlosser and S. D. Kamvar, “Simulating A File-Sharing P2P Network.” available at http://www.stanford.edu/~sdkamvar/papers/simulator.pdf.

[16] C.-H. Wu, “Peer-to-peer Systems: Macro-computing with Micro-computers.” presented at the 3rd International Conference on Open Source, July 2003.