PAPER *Special Section on Networking Technologies for Overlay Networks*

A Cooperative Mechanism for Hybrid P2P File-Sharing Networks to Enhance Application-Level QoS

Hongye FU[†], ***Nonmember***, **Naoki WAKAMIYA**[†a)], ***Member***, ***and*** **Masayuki MURATA**[†], ***Fellow***

SUMMARY Overlay networks, such as P2P, Grid, and CDN, have been widely deployed over physical IP networks. Since simultaneous overlay networks 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 proposed a cooperative mechanism for hybrid P2P file-sharing networks, where peers can find more files and exchange files with more peers. Through simulation experiments, we verified the effectiveness of cooperation from view points of application and system.

key words: *overlay networks, P2P (peer-to-peer), cooperative networks, file-sharing*

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 [5] and [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]. We regard an overlay network as an organism. 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. 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 neigh-

Manuscript received December 16, 2005.

Manuscript revised March 27, 2006.

[†] The authors are with the Department of Information Networking, Graduate School of Information Science and Technology, Osaka University, Suita-shi, 565-0871 Japan.

a) E-mail: wakamiya@ise.osaka-u.ac.jp

DOI: 10.1093/ietcom/e89–b.9.2327

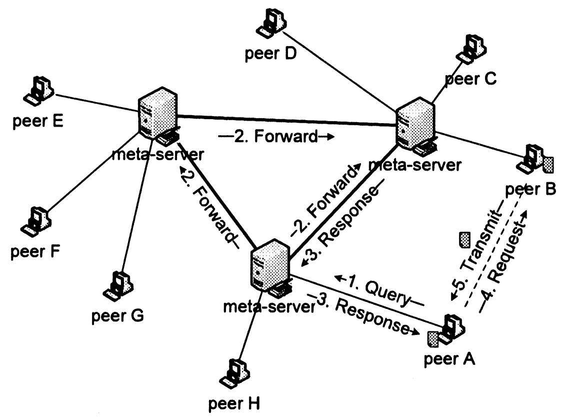


Fig. 1 A hybrid P2P file-sharing network.

boring 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 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 consider 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 [9]. We focus on the former and describe detailed mechanisms for the cooperation to answer the above-mentioned questions. In addition, we consider the benefit of a cooperative peer which plays an important role in the cooperation at the sacrifice of the increased load on itself.

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

2. Approaches for 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 consider two cooperative approaches, they are, a Shared-Peer-Based (SPB) approach where a cooperative program is introduced into those peers

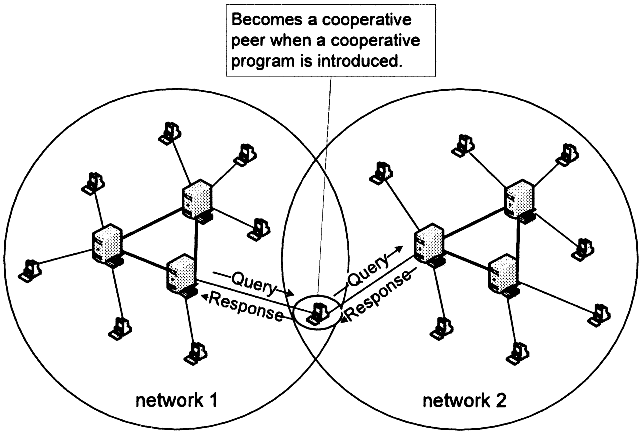

Becomes a cooperative peer when a cooperative program is introduced.
Query
Response
Query
Response
network 1
network 2

Fig. 2 Shared-Peer-Based (SPB) approach.

that participate in multiple P2P networks, and a Server-Chain-Based (SCB) approach where a cooperative program is introduced into meta-servers.

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 (network 2, 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 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. 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 has been shown in [9], 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.

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 conversion to the query message if needed, a cooperative peer transmits it to a meta-server in host P2P networks. 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 without recognizing it.

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 candidates 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.

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

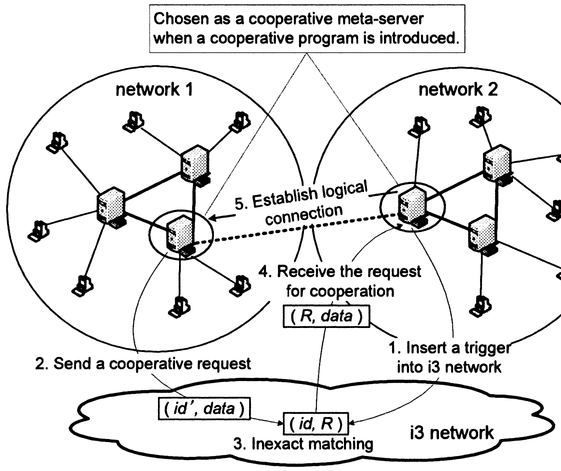

Chosen as a cooperative meta-server when a cooperative program is introduced.
network 1
network 2
5. Establish logical connection
4. Receive the request for cooperation
(R, data)
1. Insert a trigger into i3 network
2. Send a cooperative request
(id', data)
(id, R)
3. Inexact matching
i3 network

Fig. 3 Server-Chain-Based (SCB) approach.

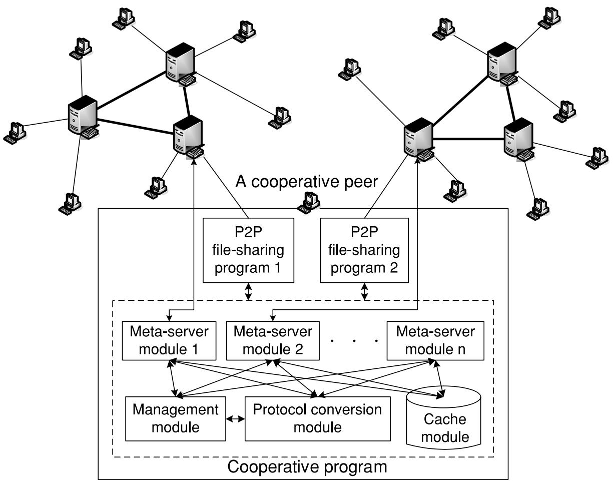


Fig. 4 Modules constituting a cooperative program for SPB approach.

message to all of its neighboring meta-servers in a P2P network by flooding. In addition, it also sends the query message to cooperative meta-servers in host networks via logical links as a peer.

When the desired file is found in a host P2P network, the peer receives a response message through the cooperative meta-server and 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. Cooperative Mechanisms for a Shared-Peer-Based Approach

In the SCB approach, a cooperative program is introduced into a meta-server. Since the number of meta-servers in a P2P network is about 1% of that of peers [12], the load is heavily concentrated around cooperative meta-servers and it causes serious congestions. In addition, for cooperative meta-servers to find another P2P network, an additional mechanism such as i3 must be deployed for the SCB approach. On the other hand, we can expect the distribution of the load over a network with the SPB approach. In addition, a shared peer has already found other P2P networks and does not need any help of additional mechanism Therefore, we take the SPB approach in the paper. We describe details about a cooperative mechanism in this section. Figure 4 illustrates components constituting a cooperative program: a management module, a protocol conversion module, meta-server modules, and a cache module.

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 2.1. 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.

3.2 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

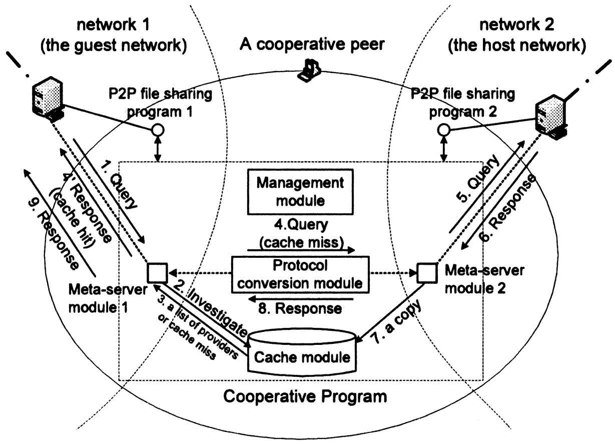


Fig. 5 Processing query and response messages.

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.

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 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. For example, when a query message from a KaZaA [13] network is forward to a Napster [14] network, the message must be modified to conform to the Napster protocol.

When a protocol conversion module 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 protocol conversion is necessary, it takes out information of a query or response from a message. Then a protocol conversion module generates a new 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 3.2. Finally, it forwards the new message to a cooorresponding meta-server module.

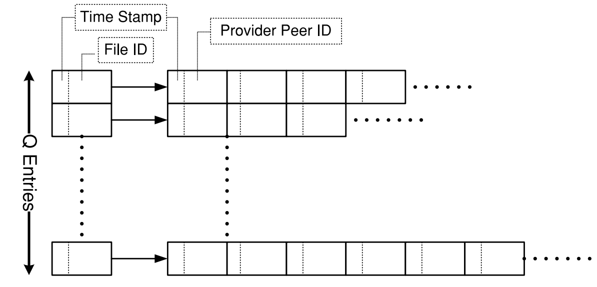


Fig. 6 Structure of a cache.

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, 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. Experimental Results

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 [12], [13], 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

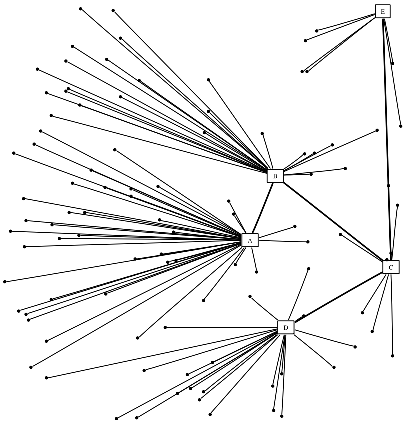


Fig. 7 An example for hybrid P2P file-sharing network topology ($m = 5, n = 100$).

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 file. 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. 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 among F kinds of files is given by the popularity of file f_r. A peer does not issue a query message for a file that it already owns.

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. As stated above, peers generate query messages for files existing in a system. In addition, we do not take into account addition or deletion of any file. However, even with changes in the file availability, results shown in this section do not change much, because such changes have the same influence on both of uncooperative and cooperative cases.

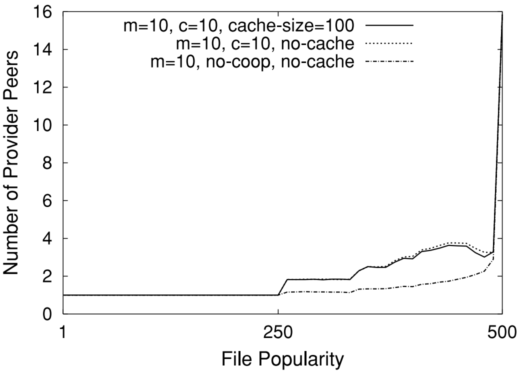


Fig. 8 Number of provider peers against popularity of files.

4.2 Evaluation of Application-Level Behaviors

As application-oriented performance measures, we use the number of provider peers and the response time. 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. The response time is defined as the duration between the emission of a query message and the reception of the first response message. In the following evaluations, we used values as the number of meta-servers m is 10, The number of peers n is 1000, i.e., the ratio of meta-servers to peers is 1% [12]. The number of cooperative peers c is 10. The number of kinds of files F is 500. A cooperative peer has a cache of the capacity of 100 entries.

Figure 8 shows the number of provider peers of successful searches against the file popularity. For 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

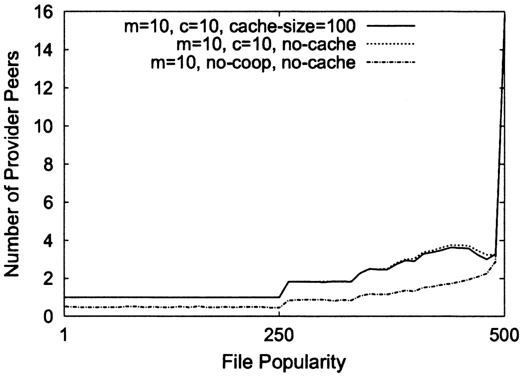


Fig. 9 Expected number of provider peers against popularity of files.

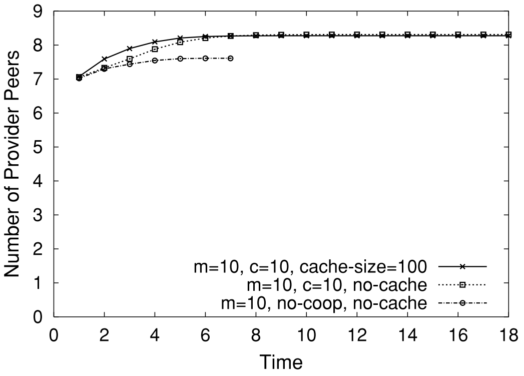


Fig. 10 Distribution of number of provider peers against time.

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. 9. The figure indicates that the cooperation among P2P networks contributes to increasing the number of 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. 10. 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. 8 through Fig. 10, 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.

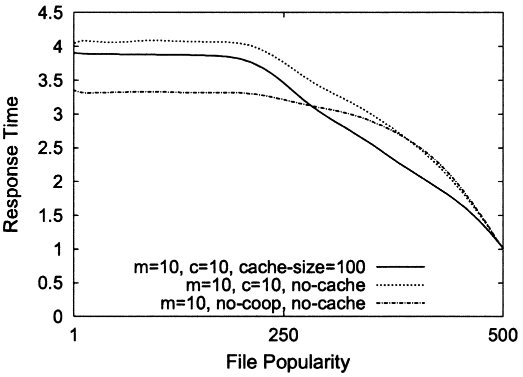


Fig. 11 Response time against popularity of files.

Figure 11 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 [9]. 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 independently of the file popularity.

4.3 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. In the following evaluations, we used values as the number of meta-servers m is 2, 5, and 10, the number of peers n is 1000, and F is 500. A cooperative peer has a cache of the capacity of 100 entries.

Figure 12 shows the average load on meta-servers in network1 where the number of cooperative peers is changed from 0 to 10. 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

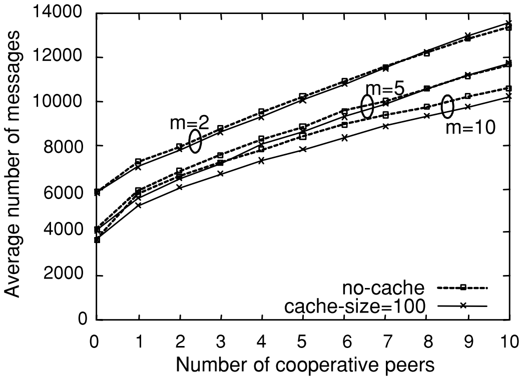


Fig. 12 Load on meta-servers against number of cooperative peers.

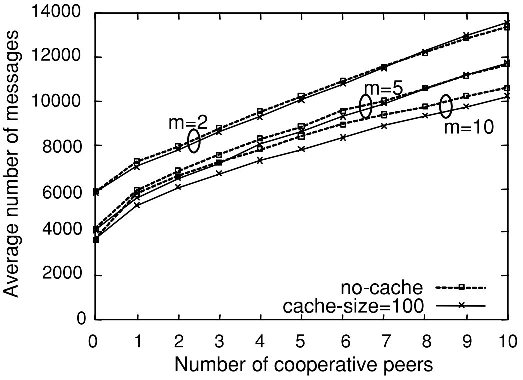


Fig. 13 Load on cooperative peers against number of cooperative peers.

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 a host network 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.

Figure 13 shows the average load on cooperative peers where the number of cooperative peers is changed from 1 to 10. 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 cooperative peers relay query messages independently of others. 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 as the number of cooperative peers increases, the hit rate also increases and the load decreases. However, we find that the decrease in the load on cooperative peers is limited in Fig. 13. 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 came from. They have already processed the query and, thus, such forwarding is redundant and only wastes the network and system resources.

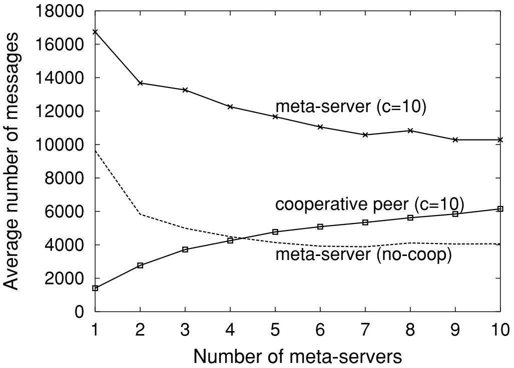


Fig. 14 Influence of number of meta-servers on load.

In Fig. 14, the number of meta-servers in each P2P network 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 a query message does not match any meta-information at the designated meta-server increases. Then, a query message is 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 peers, which is usually less powerful than meta-servers, the cooperation among P2P networks with a small number of meta-servers is desirable.

5. Conclusions and Future Work

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

Through simulation experiments, we have shown that our proposed cooperative mechanisms for the Shared-Peer-Based approach can improve the application-level QoS. A cooperative peer benefits from the cooperation more than normal peers in finding a file thanks to the caching mechanism. The caching mechanism also contributes to reduction of the load on cooperative peers. However, 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. Furthermore, to keep a cache up-to-date to avoid providing a querying peer with out-of-date and useless information, we need more complex mechanism than one proposed in this paper, which puts additional load on cooperative peers. Therefore, we now consider another mechanism to increase the benefit against the cost that a cooperative peer pays. One candidate is a one-way forwarding. In cooperation between two networks, query messages are relayed from network1 to network2 whereas query messages from network2 are always blocked on some cooperative peers.

As several other ongoing researches, we should evaluate the influence of cooperation on physical networks, and investigate a cooperative mechanism which takes into account characteristics of physical networks.

Acknowledgement

This research was partly supported 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.

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, Nov. 2003.

[2] A. Nakao, L. Peterson, and A. Bavier, “A routing underlay for overlay networks,” Proc. ACM SIGCOMM, pp.11–18, Aug. 2003.

[3] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, “Internet indirection infrastructure,” Proc. ACM SIGCOMM, pp.78–88, Aug. 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,” Proc. IEEE ICNP, poster, Nov. 2003.

[6] M. Kwon and S. Fahmy, “Synergy: An overlay internetworking architecture and its implementation,” Tech. Rep., Purdue University, Oct. 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,” Proc. Fifth International Peer-to-Peer Conference (P2P2005), pp.154–155, Aug./Sept. 2005.

[9] H. Fu, N. Wakamiya, and M. Murata, “Proposal and evaluation of a cooperative mechanism for hybrid P2P file-sharing networks,” Proc. Fourth IASTED International Conference on Communications, Internet and Information Technology (CIIT2005), pp.7–13, Oct./Nov. 2005.

[10] J. Konishi, N. Wakamiya, and M. Murata, “Proposal and evaluation of a cooperative mechanism for pure P2P file sharing networks,” Proc. 2nd International Workshop on Biologically Inspired Approaches to Advanced Information Technology (Bio-ADIT2006), pp.33–47, Jan. 2006.

[11] R. Schollmeier, “A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications,” Proc. First International Conference on Peer-to-Peer Computing (P2P2001), pp.101–102, Aug. 2001.

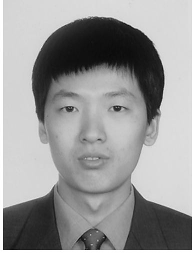
[12] J. Liang, R. Kumar, and K.W. Ross, “Understanding KaZaA,” available at http://cis.poly.edu/ ross/papers/UnderstandingKaZaA.pdf.

[13] “KaZaA Homepage,” available at http://www.kazaa.com.

[14] “Napster Homepage,” available at http://www.napster.com.

[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 microcomputers,” presented at the 3rd International Conference on Open Source, July 2003.



Hongye Fu received the B.L. degree in Japanese Linguistics from Xi’an Jiaotong University, China, in 1999. He is currently the master student at the Graduate School of Information Science and Technology, Osaka University, Japan. His research work is in the area of overlay networking and P2P networks.

Naoki Wakamiya See this issue, p.2319.

Masayuki Murata See this issue, p.2319.