A Cooperation Mechanism for Pure P2P File-Sharing Networks to Improve Application-Level QoS

Naoki Wakamiya[a], Junjiro Konishi[a], and Masayuki Murata[a]

[a]Graduate School of Information Science and Technology, Osaka University, Osaka, Japan

ABSTRACT

To provide application-oriented network services, a variety of overlay networks are deployed over physical IP networks. Since they share and compete for the same physical network resources, their selfish behaviors affect each other and, as a result, their performance deteriorates. In this paper, we propose a mechanism for pure P2P networks of file-sharing applications to cooperate with each other. In our proposal, a cooperative peer first finds another P2P network and establishes a logical link to a cooperative peer in the found network. Both ends of the logical link decide whether they cooperate or not from a viewpoint of the mutualism. When they consider they benefit from the cooperation, messages and files are exchanged among cooperative P2P networks through the logical link. For an efficient and effective cooperation, our mechanism has an algorithm for the selection of cooperative peers and a caching mechanism to avoid putting too much load on cooperative peers and cooperating networks. Simulation results showed that the number of discovered providing peers and the ratio of search hits increased about twice, while the load by the cooperation among P2P networks was reduced about half by caching.

Keywords: P2P network, file sharing, cooperation, symbiosis

1. INTRODUCTION

To provide application-oriented network services, various overlay networks are deployed over physical IP networks. To satisfy their application-level QoS, each overlay network independently measures network conditions such as the available bandwidth and latency through active or passive measurement schemes and then controls traffic, chooses routes, and changes topologies in a selfish manner. Such selfish behaviors affect each other through shared and competed physical network resources. Consequently, aiming to achieve the desired level of QoS leads to deterioration of the perceived QoS.[1, 2]

Our research group considers the symbiosis among competing overlay networks.[3] In the model of symbiotic overlay networks, overlay networks in a system evolve, interact with each other, and dynamically change internal structures. Overlay networks meet and communicate with each other. Overlay networks that benefit from each other reinforce their relationship, eventually having many inter-overlay links, and finally merging into one overlay network. Otherwise, they separate from each other. All evolutions, interactions, and internal changes are performed in a self-organizing way. Each node independently decides its behavior based on locally available information. Symbiosis among overlay networks emerges as a consequence of independent and autonomous behaviors of nodes and networks.

The cooperation among pure P2P networks is accomplished by exchanging search and response messages through logical connections established among so-called cooperative peers[4] (see Fig. 1). With such cooperation, we can expect that search messages are disseminated more effectively and a peer finds file more efficiently. Since a peer receives more response messages for a file, it can choose a more appropriate peer, i.e., faster and more reliable, among many candidate peers, leading to a higher application-level QoS. Even if P2P networks share different types or categories of files, employ different protocols, or have different architectures, there are benefits in cooperation. For example, as in,[5, 6] cooperation in routing messages provides faster and more reliable transmission of messages over P2P networks. Furthermore, when a P2P network is disconnected by failures or disappearance of peers, search and response messages can propagate among separated parts of the P2P networks

---

wakamiya@ist.osaka-u.ac.jp; phone +81-6-6879-4541; fax +81-6-6879-4544

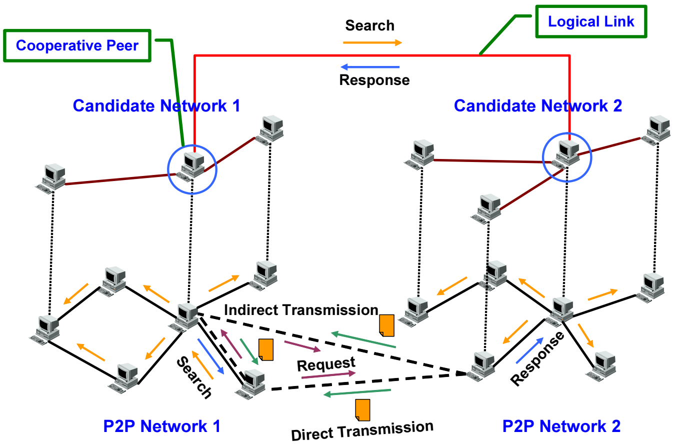


Figure 1. Cooperation among pure P2P file-sharing networks

through cooperative P2P networks. Therefore, the robustness and the resilience of P2P network are improved by cooperation as verified in.[3]

However, to accomplish the efficient and effective cooperation without introducing much load on logical networks, some careful considerations must be made. First of all, cooperative peers should be located around the center of a P2P network. If a cooperative peer is located at the edge, a large TTL value has to be set for a search message and it leads to the waste of bandwidth and network congestion due to extra and redundant copies by a flooding scheme. In addition, for effective message dissemination, it is better to choose a peer with a large number of neighbors, that is, a high-degree peer, as a cooperative peer. However, since high-degree peers neighbor each other in a P2P network with a power-law topology, we need an additional algorithm to place cooperative peers apart from each other to avoid the concentration of load. To reduce the load by the cooperation among P2P networks, a caching mechanism for cooperative peers is helpful. Our cooperation mechanism tackles the above mentioned issues for efficient and effective cooperation to improve the application-level QoS.

The rest of this paper is organized as follows. In Section 2, we give details of our cooperation mechanism. In Section 3, we evaluate our proposed mechanism through several simulation experiments in terms of the number of discovered providing peers, the search latency, the ratio of cache-hits, and the load on peers. Finally, we conclude the paper and describe future works in Section 4.

2. COOPERATIVE MECHANISM AMONG PURE P2P FILE-SHARING NETWORKS

Our proposed mechanism consists of the following steps. First, a peer in a P2P network is promoted to a candidate peer by running a cooperative program by user's intention. Candidate peers are those which are willing to play the role for the cooperation to enhance and improve their own QoS. It joins a candidate network constituting by candidate peers to exchange information for the selection of cooperative peers. Next, a tentative cooperative peer is selected in candidate peers, and then it confirms whether it is appropriate as a cooperative peer or not. After the confirmation, a tentative cooperative peer is promoted to a cooperative peer. Then, a cooperative peer discovers a cooperative peer in another overlay network. When both cooperative peer consider that the cooperation will benefit to themselves, a logical link is established between the cooperative peers. If the link is accepted by the both sides, a cooperative peer finally begins to exchange search and response messages with the cooperative peer at the other end of the logical link. When either end of the logical link considers that it is useless to maintain the link, it is disconnected.

2.1. Joining a Candidate Network

When a peer cannot discover a desired file at all, when a peer cannot discover enough number of providing peers, or when a peer cannot tolerate the delay in retrieving a file from a providing peer, a peer, i.e., a user should

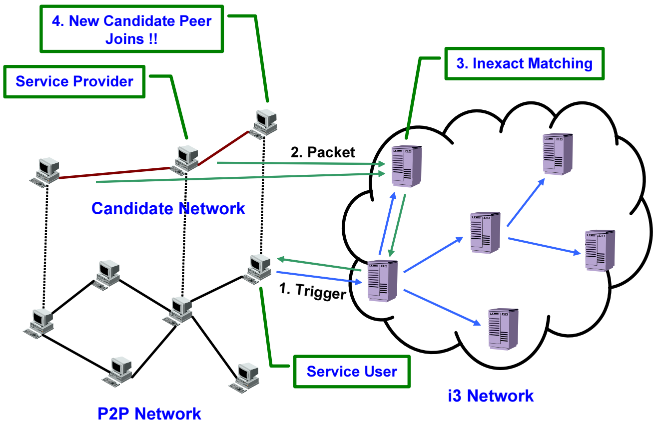


Figure 2. A new candidate peer joins the candidate network

have some frustrations. Then, the peer will consider that it can receive the higher QoS by connecting to another P2P network which provides it with the higher probability of successful search, the larger number of providing peers, and the smaller delay in file retrieval. In such a case, intending to improve its application-level QoS, the peer introduces the cooperation program on its own decision. It implies that the peer does not care whether the other peers in the same P2P network will benefit from the cooperation or not.

Then, it becomes a candidate peer, i.e., a candidate for cooperative peers. As illustrated in Fig. 1, candidate peers in a P2P network construct a candidate network to communicate with each other to select cooperative peers. A new candidate peer first discovers another candidate peer in the same P2P network by flooding a special message over the P2P network or using the i3 network.[7] For details of the mechanism using the i3, refer to the paper.[4]

2.2. Selecting Cooperative Peers based on Degree and Distance

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

It is shown in recent studies, e.g.,[8] that the Internet and many overlay networks have a power-law topology whose degree distribution follows $p(k) \propto k^{-\alpha}$. In,[9] it is shown that peers can discover files effectively through high-degree peers. Therefore, for the efficient and effective message dissemination, we select cooperative peers that have higher degree and are apart from each other to avoid the concentration of load.

First, every candidate peer advertises its degree, i.e., the number of neighbor peers, by flooding a message over a candidate network. Based on obtained information about other candidate peers, each peer ranks candidate peers in descending order of degree.

Next, a candidate peer which ranks itself highest advertises a candidacy message to all other candidate peers over a candidate network to become a tentative cooperative peer. On receiving a candidacy message, other candidate peers check the rank of the tentative cooperative peer in their ranking list. If it is not on the first in the list, a candidate peer sends a conflict message to the tentative cooperative peer. A tentative cooperative peer gives up its candidacy and removes itself from the list on receiving more conflict messages than a predetermined threshold T. The threshold T is introduced to consider the case that a candidate peer, who accidentally missed an advertisement of a tentative cooperative peer, will send a conflict message.

Then, if the number of conflict messages is smaller than T, a tentative cooperative peer floods a confirmation message with a TTL value of k in a P2P network. If any cooperative peer already exists within the range, it sends a reject message to the tentative cooperative peer. On receiving a reject message, a tentative cooperative peer gives up its candidacy and advertises its cancellation to the other candidate peers. The tentative cooperative

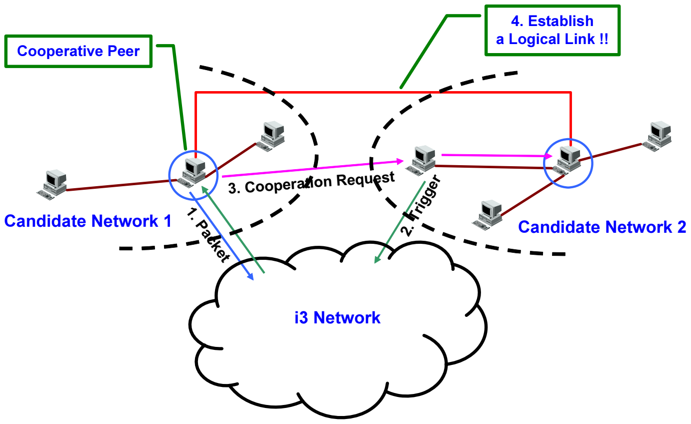


Figure 3. A cooperative peer discovers other P2P networks

peer is removed from the list and another selection is conducted again. By this mechanism, cooperative peers are kept apart from each other by more than $k + 1$ hops.

When a tentative cooperative peer does not receive any reject message in a given time, it finally becomes a cooperative peer. To select two or more cooperative peers, each candidate peer removes a new cooperative peer from the list and repeats the same procedures.

2.3. Discovering Other P2P Networks

A newly chosen cooperative peer first discovers a candidate peer in other P2P networks by using, for example the i3 network, which mediates communications among overlay networks.[7] A cooperative peer sends a *trigger* message containing a service identifier and its address to the i3 network (Fig. 3). The first bits of the service identifier correspond to the discovery service and the latter corresponding to an identifier of a P2P network and a candidate peer are generated at random to find arbitrary P2P network registered in the i3 network. For details of the mechanism using the i3, refer to the paper.[4]

When a cooperative peer finds a peer of other P2P network, for example, by receiving a *packet* message, it sends a cooperation request to the candidate peer, i.e., the sender of the *packet* message. Next, the selection of a cooperative peer is initiated by the candidate peer in a newly discovered P2P network. Then, the cooperation request is forwarded from the candidate peer to a new cooperative peer in the found P2P network. Finally, a logical link is established between those cooperative peers.

2.4. Decision of Starting Cooperation

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

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

When application protocols are different, cooperative peers must convert one protocol into the other. Therefore, it is desirable that protocols are the same or compatible to reduce the load on cooperative peers. When P2P

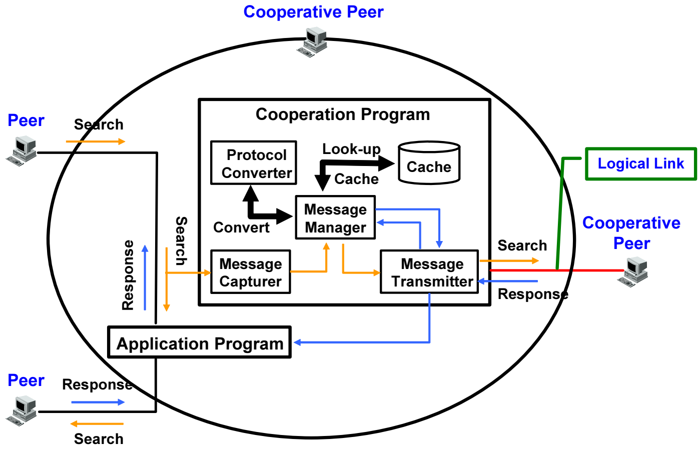


Figure 4. Behavior of a cooperative peer in the cooperation

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

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

2.5. Cooperation in Exchanging Messages and Files

In the following, we call a P2P network where a search message is originated a guest network, and another P2P network a host network. In Fig. 1, *P2P network 1* is a guest network served by a host network, i.e., *P2P network 2*. A search message sent by a peer is disseminated over a guest network by a flooding scheme.

When a search message reaches a cooperative peer, a cooperation program receives it (Fig. 4). The cooperation program looks up its local cache. Only if meta-information related to a desired file is not discovered in the cache, the search message is forwarded to a cooperative peer in a host network, after protocol conversion is applied if needed. At this time, the TTL value of the search message is decremented by one as in normal forwarding. A cooperative peer in a host network disseminates the search message over the host network by flooding. When there are two or more pairs of cooperative peers among guest and host networks, the same search message would be relayed to a host network. To eliminate the duplication, search messages have the same identifier independently of cooperative peers they traverse even if they are applied protocol conversion. Peers in a host network silently discard duplicated search messages with the same identifier.

If the desired file is discovered in a host network, a response message is generated by a providing peer and it reaches a cooperative peer in a host network along a reverse path of the corresponding search message. A cooperative peer in a host network transmits the response message to a cooperative peer in a guest network via a logical link, after protocol conversion if needed. In the case that a different protocol is used for file retrieval, a cooperative peer in a guest network caches a response message and replaces the address of a providing peer with its own address in the response message. A response message reaches the searching peer of along a reverse

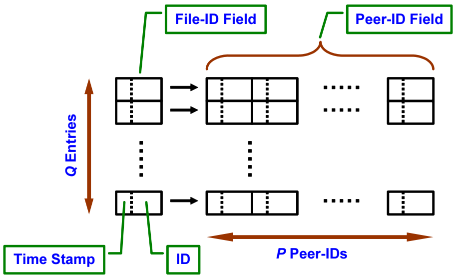


Figure 5. Structure of a caching mechanism

path of the search message over a guest network. The searching peer establishes a connection to a providing peer and obtains a file. In the case that a protocol for file retrieval is different, the peer regards a cooperative peer as a providing peer. Then, the cooperative peer retrieves the file from the original providing peer on behalf of the searching peer. Finally, the file is sent to the searching peer. Therefore, peers do not need to recognize such cooperation to receive the benefit of the cooperation.

2.6. Caching Mechanism for Cooperative Peers

There are benefits in caching meta-information of files available in a host network at a cooperative peer of a guest network. First, the load on a host network is decreased, since it does not need to receive and respond search messages that it has already answered. Second, the load on logical links is also decreased, since search messages which hit a local cache at a cooperative peer of a guest network do not traverse the link and cache-hits further suppress the generation of response messages. Third, the load on cooperative peers is also decreased. For one search message forwarded to a host network, they would receive a large number of response messages, if the search is for a popular file. Fourth, the response time of search is decreased, since a peer does not need to explore a host network for a file.

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

A cache has Q entries. Each entry consists of a file-ID, and a list of P providing peers and a timestamp. Each entry of a list of providing peers also has a timestamp (see Fig. 5). Therefore, the size of the whole cache amounts to $Q \times P$ meta-information. For more obvious discussions and experiments, we only consider a set of a file-ID and peer-IDs as a meta-information, but our scheme can easily extended to the case with other form of meta-information such as attributes and keywords.

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

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

2.7. Decision of Finishing Cooperation

Cooperation of P2P networks is terminated by disconnection of all logical links established between all pairs of cooperative peers. A logical link is maintained by the soft-state principle. When no message is transmitted through a logical link for a predetermined duration of time S, it is disconnected. In addition, a peer intentionally disconnects a logical link when it considers that it pays too much for the cooperation.

As a consequent of the cooperation, which was initiated by a peer itself, the peer helps peers in a cooperating network in finding files by relaying search and response messages. Taking into account the trade-off between the benefit and the cost of the cooperation, a peer decides whether it maintains the link or not. For example, a cooperative peer monitors the number of outgoing messages and that of incoming messages, then compare their ratio to the threshold R, which is determined by an application or a user. We should note here that details of criteria are left as one of future research topics.

3. EVALUATIONS BY SIMULATION EXPERIMENTS

In this section, we conduct several simulation experiments to evaluate the performance of our proposed mechanism in terms of the number of discovered providing peers, the search latency, the ratio of cache-hits, and the load on peers.

3.1. Simulation Conditions

We generate two networks of 10,000 peers based on BA model.[10] We assume that logical links among peers have infinite capacity and zero latency. We consider static and stable networks where there is no change in their topologies due to joins and leaves of peers.

In both P2P networks, there are files whose popularity is determined by a Zipf distribution with $\alpha = 1.0$. The number of files also follows a Zipf distribution with $\alpha = 1.0$, where the number of the most popular file is half of the number of all peers, and the number of the least popular file is 1. Therefore, there are 10,000 types of 93,668 files in the system. Files are placed on randomly chosen peers.

A search message is generated at a randomly chosen peer for a file determined in accordance with the popularity. It is disseminated by flooding within the range limited by a TTL value of 7, the default value of Gnutella. To keep the distribution of files to follow a Zipf, a peer does not retrieve a file in our evaluation.

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

3.2. Performance Evaluations

Metrics of our evaluation are the number of discovered providing peers, the search latency, and the load on peers. The number of discovered providing peers is defined as the average number of providing peers discovered in P2P networks per search message. The search latency corresponds to the number of hops between a searching peer and the nearest providing peer in P2P networks. The load on peers is the average number of messages that a peer sends and receives.

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

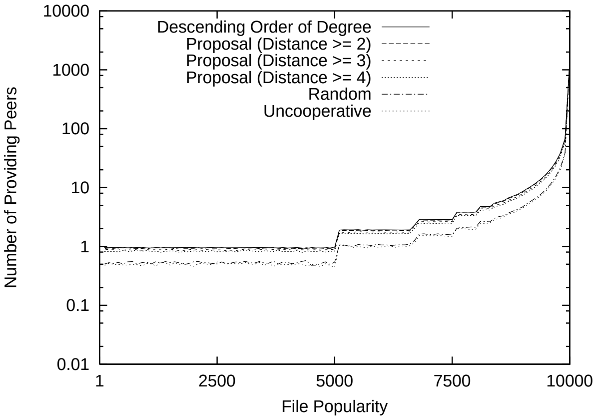


Figure 6. Relationship between the number of discovered providing peers and the file popularity

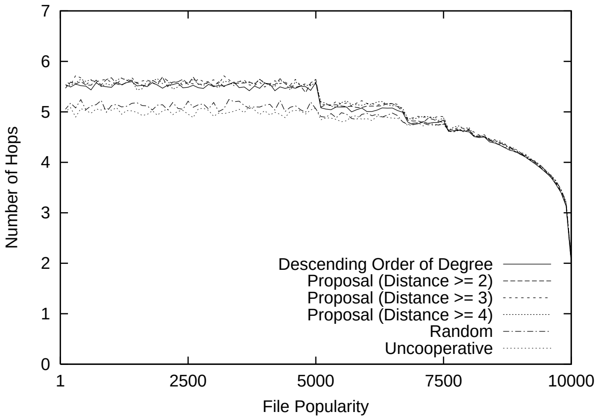


Figure 7. Relationship between the number of hops to the nearest providing peer and the file popularity

3.2.1. Number of Discovered Providing Peers

Figure 6 illustrates the relationship between the number of discovered providing peers and the file popularity. Although the figure is shown for one P2P network, the other showed similar results. The X-axis corresponds to the rank of file popularity, and the Y-axis does the number of discovered providing peers. The reason of step-shaped lines is that the number of files, which follows a Zipf distribution, takes integer values based on the popularity.

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

3.2.2. Search Latency

Figure 7 illustrates the relationship between the number of hops from a searching peer to the nearest providing peer and the file popularity. The case that searching peers could not discover any providing peers is not taken into account. The X-axis corresponds to the rank of file popularity, and the Y-axis does the number of hops.

The numbers of hops of "Descending Order of Degree" and "Proposal" are larger than those of "Random" and "Uncooperative" in all cases, because degree-dependent cooperation mechanisms can provide searching peer with a file of a distant node due to efficient message dissemination.

3.2.3. Load on Peers

Figure 8 shows how the load on a peer increases by cooperation. The ratio of increase on Y-axis is derived as the ratio of the number of duplicated search messages that a peer receives in cooperating P2P networks to that in uncooperative P2P networks. The duplicated search messages are redundant and waste physical network resources and the processing power of peers.

In comparison with "Descending Order of Degree", our proposed selection method can reduce the number of duplicated search messages at most of peers. In P2P networks used in simulation experiments, most of high-degree peers are connected with each other and form the core of a P2P network. Since cooperative peers are selected purely based on their degree in "Descending Order of Degree", they quickly flood the core of a P2P network with copied and duplicated search messages. On the other hand, in "Proposal", cooperative peers are

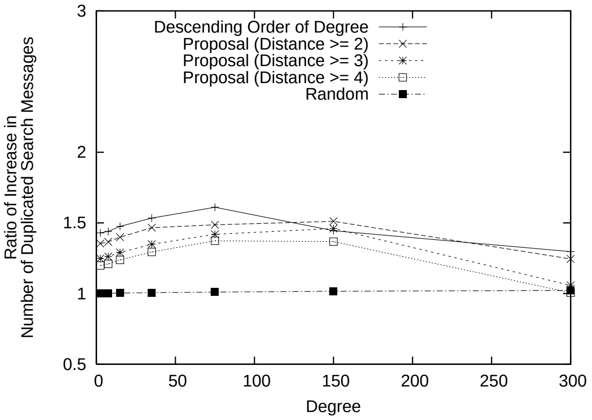
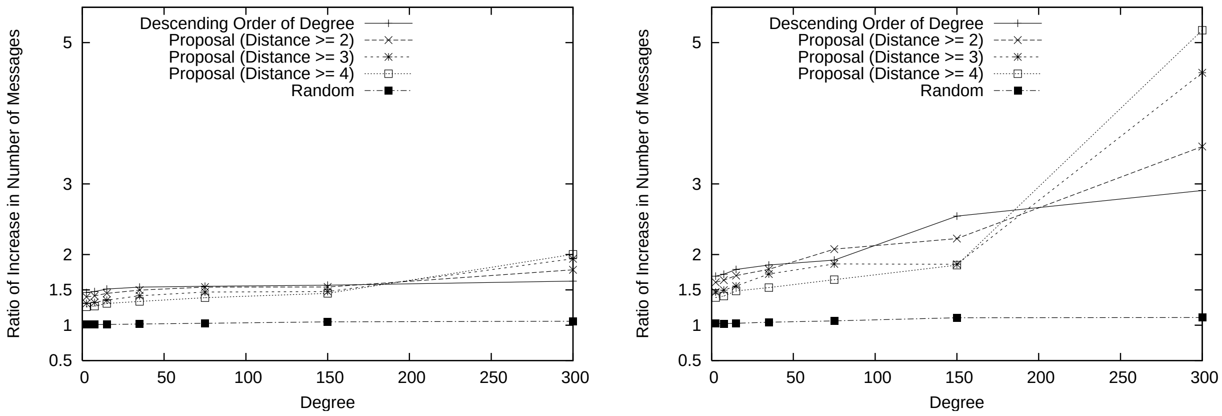


Figure 8. Distribution of the ratio of increase in the number of duplicated search messages



(a) with cache

(b) without cache

Figure 9. Distribution of the ratio of increase in the number of messages

apart from each other in a P2P network. Then, concentration of the load on high-degree peers are avoided at the sacrifice of slight increase of the load on medium-degree peers which are chosen as cooperative peers.

On the other hand, Fig. 9 shows the ratio of increase in the number of messages that a peer sends and receives including duplicated messages.

The load on high-degree peers, which are chosen as cooperative peers, increases as the number of hops among cooperative peers increases in our proposed selection method. However, the load on lower-degree peers decreases than that of “Descending Order of Degree”. In a power-law network, there are high-degree peers at the core of network. Most of cooperative peers are selected from low-degree peers as the number of hops among cooperative peers increases with our proposed method. Thus, the highest-degree cooperative peer, which is first selected as a cooperative peer, can disseminate search messages more effectively than the other cooperative peers. As a result, it must relay a large number of response messages for a search message.

However, by introducing a caching mechanism into cooperative peers, the load on the highest-degree cooperative peer becomes below the half as shown in Fig. 9. A cache can reduce the number of response messages, which are the majority of messages that a peer handles, since a search message that hits a cache does not bring

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

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

Table 2. Required processing capacity of peers and cooperative peers [KB/s]

Selection Method	10,000:1,000				10,000:5,000				10,000:10,000			
	Net.1		Net.2		Net.1		Net.2		Net.1		Net.2	
	All	Coop	All	Coop	All	Coop	All	Coop	All	Coop	All	Coop
Descending Order	57.6	2481	34.6	425	67.8	2966	53.7	1477	80.5	3614	80.3	3800
Proposal ($d = 2$)	57.4	1691	32.4	260	66.8	2054	52.0	1119	77.5	2532	77.4	2578
Proposal ($d = 3$)	57.0	1003	29.0	171	64.2	1242	47.0	737	72.7	1550	72.8	1561
Proposal ($d = 4$)	56.6	559	24.2	99.1	62.4	714	43.1	453	69.9	918	70.1	1140
Random	54.9	42.3	8.52	20.0	54.8	59.2	28.2	51.5	54.9	78.8	55.0	85.0
Uncooperative	54.6	-	5.42	-	54.4	-	27.3	-	54.2	-	54.1	-

any response messages.

3.2.4. Advantage and Disadvantage of Cooperative Peers

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

Table 1 summarizes the average ratio of cache-hits. $n : m$ stands for the number of peers in one network (Net.1) is n and that in the other network (Net.2) is m. The ratio of cache-hits ranges 32–42%. The ratio becomes higher as the number of hops among cooperative peers increases with our proposal. As the distance becomes larger, lower-degree peers are to be chosen as cooperative peers. A low-degree cooperative peer only receives search and response messages for popular files whereas a high-degree cooperative peer receives many messages for both of popular and unpopular files. Consequently, a low-degree cooperative peer has meta-information of popular files only. Since desired files are chosen following a Zipf distribution, a low-degree cooperative peer offers a higher hit ratio than a high-degree cooperative peer. In addition, we can observe that the ratio of cache-hits is higher for the cooperation among P2P networks of different sizes. This is because that the number of files is proportional to the number of peers. Then, a cooperative peer can cache a relatively large number of meta-information in a small-sized P2P network.

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

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

4. CONCLUSION

In this paper, in a context of the overlay network symbiosis, we proposed a mechanism for pure P2P networks of file-sharing applications to cooperate with each other. Through several simulation experiments, it was shown that application-level QoS in term of the number of discovered providing peers was improved by selecting high-degree peers as cooperative peers. Furthermore, it was shown that by keeping cooperative peers apart from each other, the redundant load on the P2P network was reduced. A caching mechanism for cooperative peers was shown to be effective in reducing the load on cooperative peers.

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

REFERENCES

1. L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker, "On Selfish Routing in Internet-Like Environments," in *Proceedings of ACM SIGCOMM Conference 2003*, pp. 151–162, Aug. 2003.
2. M. Seshadri and R. H. Katz, "Dynamics of Simultaneous Overlay Network Routing," Tech. Rep. UCB/CSD-03-1291, Electrical Engineering and Computer Sciences (EECS), University of California Berkeley (UCB), Nov. 2003.
3. N. Wakamiya and M. Murata, "Toward Overlay Network Symbiosis," in *Proceedings of the 5th IEEE International Conference on Peer-to-Peer Computing (P2P2005)*, pp. 154–155, Aug. 2005.
4. 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-ADIT)*, pp. 33–47, Jan. 2006.
5. M. Kwon and S. Fahmy, "Toward Cooperative Inter-overlay Networking," in *Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP)*, poster paper, Nov. 2003.
6. M. Kwon and S. Fahmy, "Synergy: An Overlay Internetworking Architecture and its Implementation," in *Proceedings of the 14th IEEE International Conference on Computer Communications and Networks (ICCCN)*, pp. 401–406, Oct. 2005.
7. I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet Indirection Infrastructure," in *Proceedings of ACM SIGCOMM Conference 2002*, pp. 73–88, Aug. 2002.
8. M. E. J. Newman, "The Structure and Function of Complex Networks," *SIAM Review* **45, 2**, pp. 167–256, 2003.
9. L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, "Search in Power-law Networks," *Physical Review E* **64, 046135**, Sept. 2001.
10. A. L. Barabasi and R. Albert, "Emergence of Scaling in Random Networks," *Science* **286**, pp. 509–512, Oct. 1999.