PAPER *Special Section on Networking Technologies for Overlay Networks*

Design and Evaluation of a Cooperative Mechanism for Pure P2P File-Sharing Networks

Junjiro KONISHI†, *Nonmember*, **Naoki WAKAMIYA**†a), *Member*, *and* **Masayuki MURATA**†, *Fellow*

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

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

1. Introduction

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

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

Our research group considers the symbiosis among competing overlay networks [8]–[10]. In the model of symbiotic overlay networks, overlay networks in a system evolve, interact with each other, and dynamically change internal structures, but they still behave in a selfish manner. Overlay networks meet and communicate with each other in a probabilistic way. Overlay networks that benefit from each other reinforce their relationship, eventually having many inter-overlay links, and merging one overlay network. Otherwise, they separate from each other. All evolutions, interactions, and internal changes are performed in a self-organizing way. Each node independently decides its behavior based on locally available information. Symbiosis among overlay networks emerges as a consequence of independent and autonomous behaviors of nodes and networks.

For this purpose, we need mechanisms for overlay networks to communicate with each other. In our previous works [9], [10], we proposed mechanisms for efficient and effective cooperation among P2P networks of file-sharing applications. In a P2P network, hosts called peers directly communicate with each other and exchange information without the mediation of servers. According to user's intention, peers behave on its own decision as an individual does in a group or society. One typical example of P2P applications is a file-sharing system. Napster and WinMX

Manuscript received December 16, 2005.
Manuscript revised March 24, 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@ist.osaka-u.ac.jp

DOI: 10.1093/ietcom/e89–b.9.2319

are categorized as hybrid P2P networks where there are so-called meta-servers to maintain meta-information, e.g., list of file holders to help peers to find files. In the case of the cooperation among hybrid P2P networks, it is an architectural problem that meta-servers must deal with a large amount of messages since peers always try to obtain meta-information from meta-servers. We proposed a mechanism for cooperation among hybrid P2P networks and investigated the influence of system conditions such as the number of peers and the number of meta-information in [9]. On the other hand, Gnutella and Winny are pure P2P networks without a server for searching files. Thus, a peer has to find the desired file by itself by emitting a query message into the network. Other peers in the network response to the query with a response message and relay the query to their neighbor peers (Fig. 1).

The cooperation among pure P2P networks is accomplished by exchanges of search and response messages among them through logical connections established among so-called cooperative peers [10]. 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 [3], [4], 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 through cooperative P2P networks. Therefore, the robustness and the resilience of P2P network are improved by cooperation as verified in [8].

However, to accomplish the efficient and effective cooperation without introducing much load on logical and physical networks, some careful considerations must be made. For example, if a cooperative peer is located at the edge of a P2P network, it has to set a large TTL (Time to Live) value for search messages to spread over the network. As a result, the number of rejected duplicated search messages over P2P networks increases. They waste network bandwidth and causes network congestions. Therefore, in [10], we proposed an algorithm to choose appropriate cooperative peers. We also gave some considerations on incentives that a selfish peer began cooperation. Through simulation experiments, we found that the average load on peers to accomplish some degree of reachability, which is defined as the ratio of the number of peers that flooded search messages reach to the total number of peers in a network, became lower than the case without the cooperation. However, it was also shown that the load on the highest-degree cooperative peers became considerably high. In this paper, for more efficient and effective cooperation, we propose a mechanism for cooperation of pure P2P networks with a caching mechanism at cooperative peers.

The rest of this paper is organized as follows. In Sect. 2, we propose a mechanism for cooperation among pure P2P networks of file-sharing applications. In Sect. 3, we evaluate our mechanism through several simulation experiments from the viewpoint of the number of found file holders, the search latency, and the load on peers. Finally, we conclude the paper and describe future works in Sect. 4.

2. Cooperation Mechanism for Pure P2P File-Sharing Networks

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

Our proposed mechanism consists of the following steps. First, a peer in a P2P network is promoted to a candidate peer by running a cooperative program. It joins to a candidate network constituting by candidate peers to ex-

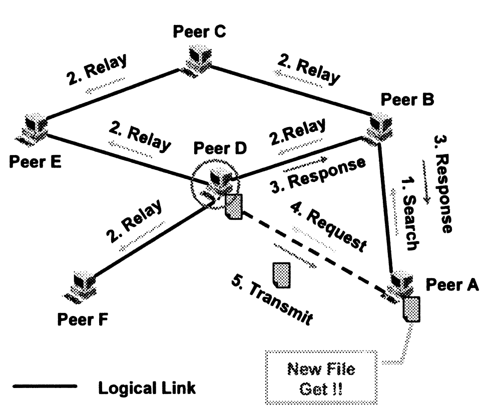


Fig. 1 Flooding over a pure P2P file-sharing network.

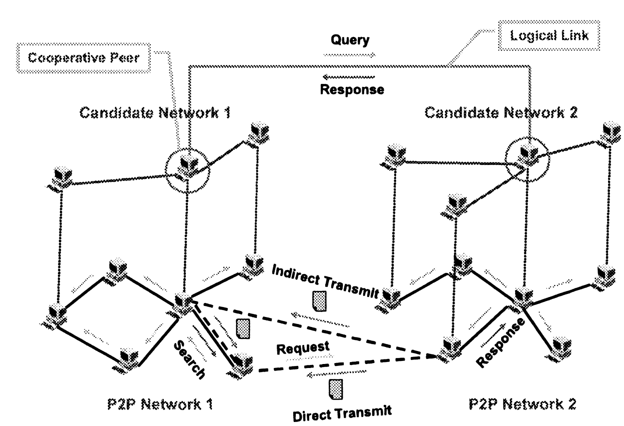


Fig. 2 Cooperation of pure P2P file-sharing networks.

change 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 finds a cooperative peer in another overlay network and establishes a logical link to it. If the link is accepted by the counterpart cooperative peer, a cooperative peer finally begins to exchange search and response messages with the cooperative peer at the other end of the logical link. We describe in the following the selection of cooperative peers, the preparation before the cooperation of P2P networks, and the behavior of peers in cooperative P2P networks in detail.

2.1 Joining a Candidate Network

When a peer is not satisfied with an application-level QoS received from a P2P network of file-sharing application, it considers to enhance and improve its application-level QoS by its own decision. For example, when a peer cannot find a desired file at all, when a peer cannot find enough number of files, or when a peer cannot tolerate the delay in retrieving a file from a provider peer, a peer, i.e., a user should have some frustrations. The peer will consider that it can receive the higher QoS by connecting to another P2P network which provides it with the higher probability of successful search, the larger number of provider peers, and the smaller delay in file retrieval. In such a case, intending to enhance and improve its application-level QoS, the peer introduces the cooperation program independently of others. It implies that the peer does not care whether the other peers in the same P2P network will benefit from the cooperation or not. Then, it becomes a candidate peer, i.e., a candidate for cooperative peers. As illustrated in Fig. 2, candidate peers in a P2P network construct a candidate network to communicate with each other to select cooperative peers.

A new candidate peer first finds another candidate peer in the same P2P network by flooding a special message over the P2P network or using the i3 network [5]. In the latter case, a new candidate peer registers itself to an i3 service repository by sending a *trigger* message containing a service identifier and its address. On the other hand, candidate peers in a candidate network send *packet* messages containing a service identifier and its address to the i3 network periodically. A new candidate peer receives one of their *packet* messages and establishes a logical link to the candidate peer. After that, the new candidate peer deletes its *trigger* message from the i3 service repository. For details of the mechanism, please refer to the paper [10].

2.2 Selecting Cooperative Peers

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

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

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

2.3 Finding Other P2P Networks

A newly chosen cooperative peer first finds a candidate peer in other P2P networks by using the i3 network, which mediates communications among overlay networks. A cooperative peer sends a *trigger* message containing a service identifier and its address to the i3 network. The last bits of the service identifier, which are used as an identifier of a

P2P network and a candidate peer, are generated at random to find arbitrary P2P network registered in the i3 network. When a cooperative peer receives a *packet* message which matches the *trigger* message by *inexact matching* where the *packet* message has a service identifier matching the longest pattern of bits with the *trigger* message, it sends a cooperation request to the candidate peer, i.e., the sender of the *packet* message, in another P2P network. Next, the selection of a cooperative peer is initiated by the candidate peer in a newly found P2P network. Then, the cooperation request is forwarded from the candidate peer to a new cooperative peer. Finally, a logical link is established between those cooperative peers.

2.4 Decision of Starting Cooperation

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

When application protocols are different, cooperative peers must convert one protocol into the other. Therefore, it is desirable that protocols are the same or compatible to reduce the load on cooperative peers. When P2P networks are different in their size, peers in a larger P2P network cannot expect the benefit from the cooperation very much. However, the newly introduced load from a smaller cooperative P2P network is considered not much. On the other hand, peers in a smaller P2P network can share and find more files 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 Sect. 1, it is worth cooperating with a different P2P network from a system-oriented viewpoint.

A cooperative peer obtains that information and defines priorities to each of them. When the weighted sum is beyond a threshold for both cooperative peers, the cooperation is started. 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 topics.

2.5 Relaying Messages and Retrieving Files

In the following, we call a P2P network where a search message is originated a guest network, and another P2P network a host network. In Fig. 2, P2P network 1 is a guest network served by a host network, i.e., P2P network 2. A search message sent from a peer is disseminated over a guest network by a flooding scheme. When a search message reaches a cooperative peer, it looks up its local cache. Only if a desired file is not found 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 a file is found in a host network, a response message is generated by a provider 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 provider peer with its own address in the response message. A response message reaches the source peer of the search message along a reverse path of the search message over a guest network. The searching peer establishes a connection to a provider peer and obtains a file. In the case that a protocol for file retrieval is different, the peer regards a cooperative peer as a provider peer. Then, the cooperative peer retrieves the file from the original provider 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 in Cooperative Peer

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

There are benefits in caching meta-information of files available in a host network at a cooperative peer of a guest network. First, the load on a host network is decreased, since it does not need to receive and respond search messages that it has already answered. Second, the load on a logical link 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 easily be occupied by meta-information of popular files. However, popular files are easily found in a guest P2P itself. Therefore, to avoid 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, a time stamp of the entry (file-TS) and a list of P file holders. Each file holder also has a time stamp (holder-TS) as shown Fig. 3. Therefore the size of the whole cache amounts to $Q \times P$ meta-information. For easier discussions and experiments, we only consider a set of a file-ID and holder-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 at time T, the cooperative peer obtains a file-ID and a holder-ID 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 file-TS is replaced with the new entry. A file-TS of the new entry of a file-ID and a holder-TS of a holder-ID of the entry are set at T. If the meta-information of the same file-ID is in the cache, the file-TS is renewed with T. Then, a list of file holders is investigated to see whether there already is the same holder-ID or not. If there is, a holder-TS of the holder-ID is set at T. Otherwise, the holder-ID is added to the list with a new holder-TS, or the holder-ID with the oldest holder-TS is replaced with the new holder-ID in a full list.

On receiving a search message from peers in the same P2P network, a cooperation peer first examines its local cache. If there is a match in the cache, it generates a response message constituting a list of file holders and sends it back to the requesting peer via the reversed path that the request 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.

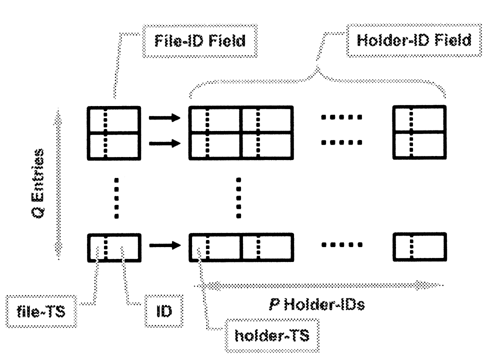


Fig. 3 Construction of a cache mechanism.

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. Simulation Evaluation

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

3.1 Simulation Environments

We generate two scale-free networks of 10,000 peers based on BA model [13]. We assume that logical links among peers have infinite capacity and zero latency. We consider static and stable networks where there is no change in their topologies due to joins and leaves of peers. There are 5,000 kinds of files in both P2P networks. Their popularity is determined by Zipf distribution of $\alpha = 1.0$. The number of files also follows Zipf distribution of $\alpha = 1.0$, where the number of the least popular file is 1. For example, in a P2P network of 10,000 peers, there are 5,000 kinds of 43,376 files and the number of the most popular file is 5,000. Figure 4 illustrates the cumulative distribution of the number of files against the popularity. 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 TTL of 7, the default value of Gnutella. To keep the distribution of files to follow Zipf, a peer does not retrieve a file in our evaluation. In all cases, 20,000 search

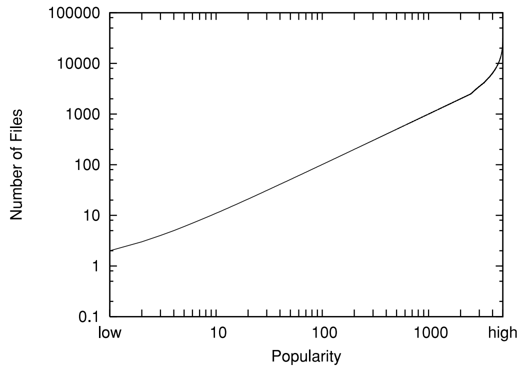


Fig. 4 CDF of the number of files against the popularity.

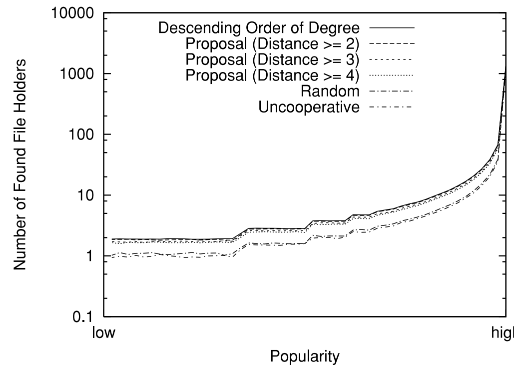


Fig. 5 Relationship between the number of found file holders and the file popularity.

messages are generated in P2P networks.

The number of cooperative peers is set at 10. Therefore, there are ten logical links among P2P networks. Cooperative peers are chosen among all peers, that is, all peers are candidate peers in our simulation experiments. 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 $P = 10$ holder-IDs. For comparison purposes, we conducted simulation experiments for different mechanisms. "Descending Order of Degree" in the following figures corresponds to the degree-dependent selection where cooperative peers are chosen in a descending order of degree. In "random," cooperative peers are chosen at random. "Uncooperative" indicates the result of the case where there is no cooperation. "Proposal (distance $\geq j$)" 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 j hops. In this case, a TTL value of a confirmation message is set at $j - 1$. Metrics of our evaluation are the number of found file holders, the search latency, and the load on peers. The number of found file holders is defined as the average number of file holders found in P2P networks per search message. The search latency corresponds to the number of hops between a searching peer and the nearest file holder in P2P networks. The load on peers is the average number of times that a peer sends and receives search and response messages.

3.2 Evaluation of Number of Found File Holders

Figure 5 illustrates the relationship between the number of found file holders and the file popularity. It is shown that by connecting two networks by the degree-dependent selection or our proposal, a peer can find twice the number of uncooperative networks. In addition, it can be seen that the number of found file holders of "Random" is almost the same as that of "Uncooperative." It means that the cooperation of P2P networks by randomly-chosen cooperative peers does not improve the application-level QoS very much. Since the majority is low-degree peers in a power-law network, a random selection algorithm often chooses low-degree peers as cooperative peers which cannot effectively disseminate search messages over a host P2P network. The reason of step-shaped lines in Fig. 5 is that the number of files, which follows Zipf distribution, takes integer values based on the popularity.

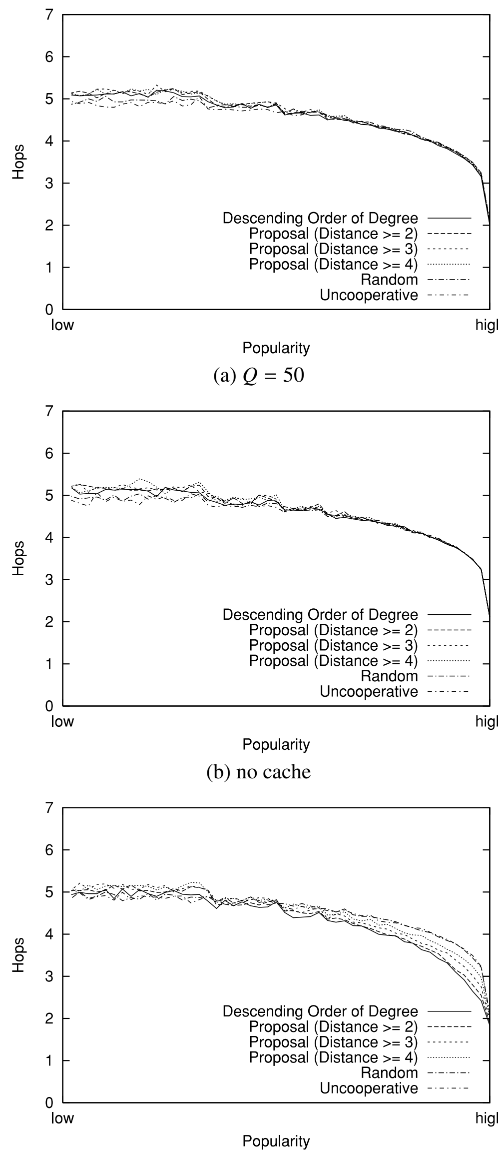
3.3 Evaluation of Search Latency

Figure 6(a) illustrates the relationship between the number of hops to the nearest file holder and the file popularity. It is shown that the number of hops is almost the same among mechanisms. Especially, we should notice that caching at a cooperative peer does not contribute to faster search in comparison with results of no-cache cases shown in Fig. 6(a). This is because that the average rate of cache hit at a cooperative peer is about 40% in simulation experiments. When we increase the capacity from 50 entries to 500 entries, the search latency to receive the first response message is reduced for popular files as shown in Fig. 6(c). In this case, the average hit rate is about 70%.

However, the cache capacity of 500 entries is 10% of 5,000 kinds of files available in the P2P networks. In reality, about 10,000,000 to 100,000,000 are shared in a P2P network [14] and maintaining a cache of 1,000,000 to 10,000,000 entries costs too much against the improvement of one-hop delay. Therefore, we conclude that caching meta-information at cooperative peers does not provide peers with a higher application-level QoS in term of the search latency. Next, we evaluate the effect of caching from a system-level point of view.

3.4 Evaluation of Load on Peers

Figure 7 illustrates the distribution of the number of times that a peer receives duplicated search messages. The duplicated search messages are redundant and lead to the waste of physical network resources and the processing power of peers. In comparison with "Descending Order of Degree," our proposal can reduce the number of duplicated messages especially at high-degree 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. In "Descending Order of Degree," since cooperative peers



(a) Q = 50

(b) no cache

(c) Q = 500

Fig. 6 Relationship between the number of hops and the file popularity.

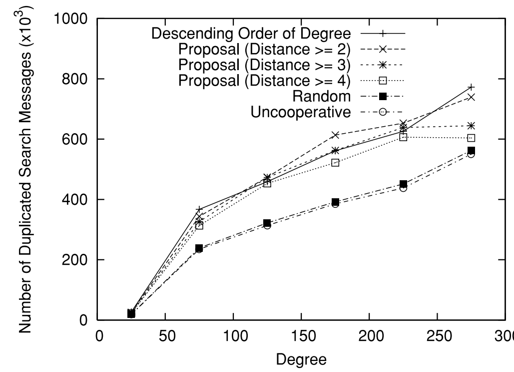


Fig. 7 Distribution of the number of duplicated search messages.

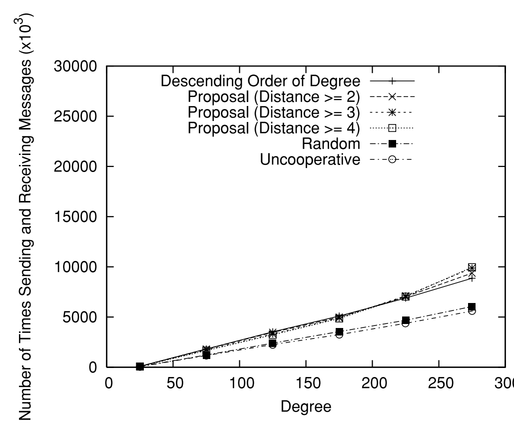


Fig. 8 Distribution of the load on peers with caching.

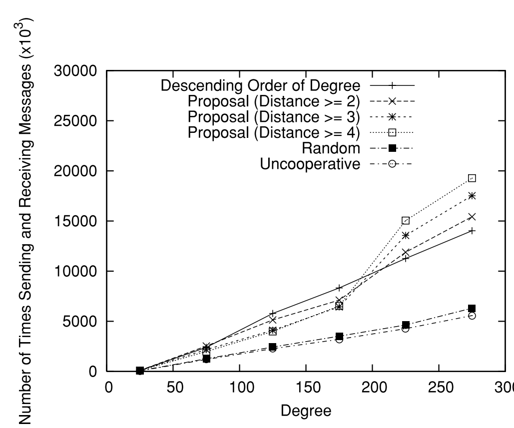


Fig. 9 Distribution of the load on peers without caching.

are selected purely based on their degree, they quickly flood the core of a P2P network with copied and duplicated search messages. On the other hand, in our proposal, cooperative peers are apart from each other. 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. 8 illustrates the distribution of the number of times that a peer sends and receives search and response messages including duplicated messages. As Fig. 8 shows, the load on high-degree peers, which are chosen as cooperative peers, is high and increases as the number of hops between cooperative peers increases in our proposed methods. However, the load on lower-degree peers becomes lower than that of "Descending Order of Degree." Although not shown in a figure, since peers with lower degree are the majority in P2P networks, the total and average load on a P2P network is much lower with our proposal than "Descending Order of Degree" by about 8.9%. In comparison with results of the case without cache in Fig. 9, it is obvious that our caching mechanism considerably reduces the load on peers as we intended. Especially, the load on the high-degree peers becomes about the half. The most of messages that a cooperative peer handles is response messages. By introducing a cache, a cooperative peer can avoid receiving a large number of response messages from a host network.

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 simulation experiments, it was shown that application-level QoS in term of the number of found file holders 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 of cooperative peer was shown to be effective in reducing, the load on cooperative peers, but it did not contribute to faster search.

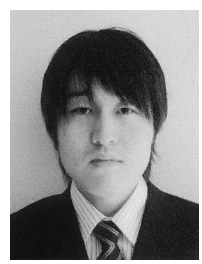
As future research works, we first propose an algorithm to decide to start and finish the cooperation among P2P networks. We also investigate behaviors of cooperation among dynamic P2P networks, which change their topology as consequences of cooperation. Furthermore, we should evaluate influences of cooperation to a physical network.

Acknowledgements

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.

References

[1] L. Qiu, Y.R. Yang, Y. Zhang, and S. Shenker, “On selfish routing in Internet-like environments,” Proc. ACM SIGCOMM Conference 2003, pp.151–162, Aug. 2003.
[2] M. Seshadri and R.H. Katz, “Dynamics of simultaneous overlay network routing,” Technical Report of Electrical Engineering and Computer Sciences (EECS), University of California Berkeley (UCB), UCB/CSD-03-1291, Nov. 2003.
[3] M. Kwon and S. Fahmy, “Toward cooperative inter-overlay networking,” 11th IEEE International Conference on Network Protocols (ICNP), poster, Nov. 2003.
[4] M. Kwon and S. Fahmy, “Synergy: An overlay Internetworking architecture and its implementation,” Proc. 14th IEEE International Conference on Computer Communications and Networks (ICCCN), pp.401–406, Oct. 2005.
[5] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, “Internet indirection infrastructure,” Proc. ACM SIGCOMM Conference 2002, pp.73–88, Aug. 2002.
[6] A. Nakao, L. Peterson, and A. Bavier, “A routing underlay for overlay networks,” Proc. ACM SIGCOMM Conference 2003, pp.11–18, Aug. 2003.
[7] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, “Resilient overlay networks,” Proc. 18th ACM Symposium on Operating Systems Principles (SOSP), pp.131–145, Oct. 2001.
[8] N. Wakamiya and M. Murata, “Toward overlay network symbiosis,” Proc. Fifth IEEE International Conference on Peer-to-Peer Computing (P2P2005), pp.154–155, Aug. 2005.
[9] H. Fu, N. Wakamiya, and M. Murata, “Proposal and evaluation of a cooperative mechanism for hybrid P2P file sharing networks,” Proc. 4th IASTED International Conference on Communications, Internet, and Information Technology, pp.7–13, Oct. 2005.
[10] J. Konishi, N. Wakamiya, and M. Murata, “Proposal and evaluation of a cooperative mechanism for pure P2P file sharing networks,” 2nd International Workshop on Biologically Inspired Approaches to Advanced Information Technology (Bio-ADIT), pp.33–47, Jan. 2006.
[11] M.E.J. Newman, “The structure and function of complex networks,” SIAM Review, vol.45, no.2, pp.167–256, 2003.
[12] L.A. Adamic, R.M. Lukose, A.R. Puniyani, and B.A. Huberman, “Search in power-law networks,” Phys. Rev. E, vol.64, 046135, Sept. 2001.
[13] A.L. Barabasi and R. Albert, “Emergence of scaling in random networks,” Science, vol.286, pp.509–512, Oct. 1999.
[14] C.H. Wu, “Peer-to-peer systems: Macro-computing with micro-computers,” 3rd International Conference on Open Source, July 2003.



Junjiro Konishi received the M.E. degree in the Information and Computer Sciences from Osaka University, Osaka, Japan, in 2006. He studies information networking and his research work is in the area of overlay networks.

Naoki Wakamiya received the M.E. and Ph.D. degrees from Osaka University in 1994 and 1996, respectively. He was a Research Associate of Graduate School of Engineering Science, Osaka University from April 1996 to March 1997, a Research Associate of Educational Center for Information Processing from April 1997 to March 1999, an Assistant Professor of Graduate School of Engineering Science from April 1999 to March 2002. Since April 2002, he is an Associate Professor of Graduate School of Information Science and Technology, Osaka University. His research interests include overlay networks, mobile ad-hoc networks, and wireless sensor networks. He is a member of IPSJ, ACM, and IEEE.

Masayuki Murata See this issue, p.2280.