An Improved Scheme for Interest Mining Based on a Reconfiguration of the Peer-to-Peer Overlay

Naomi AOKI   Satoshi FUJITA
Graduate School of Engineering, Hiroshima University
Kagamiyama 1-4-1, Higashi-Hiroshima, 739-8527, Japan
E-mail: {sptx6431,fujita}@se.hiroshima-u.ac.jp

***Abstract*—Tan *et al.* proposed a scheme to improve the quality of a file search in unstructured Peer-to-Peer systems by focusing on the similarity of interest of the participating peers. Although it certainly improves the cost/performance ratio of a simple flooding-based scheme used in conventional systems, the Tan's method has a serious drawback such that a query cannot reach a target peer if a requesting peer is not connected with the target peer through a path consisting of peers to have similar interest to the given query. In order to overcome such drawback of the Tan's method, we propose a scheme to reconfigure the underlying network in such a way that a requesting peer has a neighbor interested in the given query, before transmitting a query to its neighbors. The performance of the proposed scheme is evaluated by simulation. The result of simulation indicates that it certainly overcomes the drawback of the Tan's method.**

I. Introduction

Recently, unstructured Peer-to-Peer (P2P) systems have attracted considerable attentions as a way of providing scalable network services over large-scale computer networks such as the Internet. A P2P system consists of a large number of computers called **peers** (or nodes) which are connected with a logical network called P2P overlay to realize a direct connection between them. Unlike conventional Client/Server systems, each peer participating to a P2P system can simultaneously play the roles of a server and a client, i.e., each service in a P2P system is provided from a peer to another peer in a *peer-to-peer* manner. In addition, in contrast to structured P2Ps [1], [6], [7] which have a systematic way to forward a given query to a peer holding an index to a target file, unstructured P2Ps have no such systematic way for the file search; i.e., most of conventional file search schemes for unstructured P2Ps are based on the notion of *uncontrolled query propagation* such as flooding and random walk.

A flooding-based file search could find a number of files matching a given query in a relatively short time, while it generally needs a large number of message transmissions. The number of message transmissions could be reduced by appropriately setting TTL (Time-To-Live) to each query, while it restricts the location of peers from which the requester can receive the search result to a small portion of the network centered at the requesting peer. Tan *et al.* [8] proposed a scheme to improve the cost/performance ratio of a file search in unstructured P2Ps by focusing on the *similarity of interest* of the participating peers. More concretely, it modifies the transmission of a query to all neighbors in the original flooding-based scheme in such a way that the query is forwarded merely to several neighbors to have a similar file to the given query, where similarity of files is measured by the normal cosine similarity as will be described later.

Although it certainly improves the cost/performance ratio of a simple flooding, the Tan's method, which is known as an Interest Mining in the literature, has a serious drawback such that a query cannot reach a target peer if a requesting peer is not connected with the target peer by a path consisting of peers to have similar file to the given query. It would significantly reduce the hit ratio of a file search under a situation in which each peer randomly selects its neighbors as in Gnutella [2] and Winny. In order to overcome such drawback, we propose a scheme to *reconfigure* the underlying P2P overlay in such a way that a requesting peer is adjacent with a peer to have a file similar to the given query. The performance of the scheme is evaluated by simulation. The result of simulation indicates that it certainly overcomes the drawback of the Tan's method, and exhibits a better performance than a simple extension of the Tan's method which refers to peers within distance k to determine a set of recipients of a given query.

The remainder of this paper is organized as follows. Section II describes preliminaries. Section III describes an overview of the Tan's method. Our proposed scheme is given in Section IV, and the result of evaluation is shown in Section V. Finally, Section VI concludes the paper with future problems.

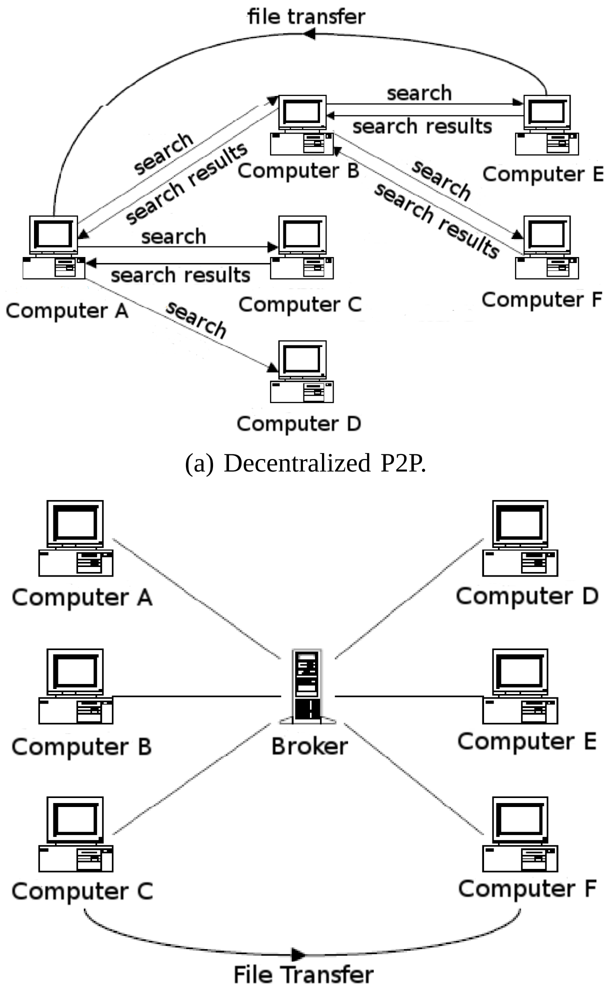
II. Preliminaries

A. P2P Architecture

Existing P2P architectures can be classified into three categories, i.e., centralized type, decentralized type, and hierarchical type.

Napster [5] is a centralized P2P file-sharing system consisting of a number of peers and an index server, where the server keeps the information on the location of shared files in order to respond to queries issued by the participating peers. Although such a centralized approach realizes an efficient query processing and a reduction of the response time to the change of the index information, it is well known that it has serious drawbacks, such as an overload and a single point of failure of the index server.

Gnutella [2] is a fully decentralized P2P system, in which each peer can identify the location of shared files by flooding



(a) Decentralized P2P.

(b) Hierarchical P2P with super-peers.

Fig. 1. Peer-to-peer networks.

queries to the other peers. Figure 1 (a) illustrates a decentralized P2P system. Although such a flooding-based approach removes the bottleneck in the centralized approach, it causes several critical issues in realizing efficient file search, such as the delay of the response to a given query, high traffic of the underlying network, and an inaccuracy of the search result.

Hierarchical P2Ps, which are based on the notion of super-peers [9], [3], have attracted considerable attentions in recent years, as a promising way to overcome drawbacks of the above two approaches. A typical hierarchical P2P consists of two layers; i.e., the top layer consisting of super-peers and the bottom layer consisting of ordinary peers. The bottleneck in the centralized scheme could be certainly overcome by replacing the (single) index server by a collection of super-peers, and the low efficiency of the fully decentralized scheme could be overcome by introducing an upper layer of super-peers. See Figure 1 (b) for illustration.

B. Vector Space Model

In this paper, we adopt the cosine similarity as the measure of the similarity between two documents. More concretely, we consider a vector space model (VSM) in which each document is represented by a vector, and the similarity between two documents is evaluated by calculating the similarity between two corresponding vectors. Each coordinate in the VSM corresponds to a word in the documents; i.e., a vector has a non-zero entry at the i^{th} coordinate iff the corresponding document contains the i^{th} word (if it contains the i^{th} word, the value of the i^{th} coordinate is determined by applying a function known as tf-idf described below). More concretely, we represents a vector corresponding to document d, in the following manner:

$$V(d) = (t_1, w_1(d); \ldots; t_j, w_j(d); \ldots; t_m, w_m(d)),$$

where m is the number of words in document d, and $w_i(d)$ is the weight of word t_i in document d whose value is determined using function tf-idf.

C. tf-idf

Let D be a set of documents, and t be a set of all words contained in D. At first, function tf is defined as the number of occurrences of t in D. Next, function idf is defined such that a word specific to the document takes a large value, and words commonly contained in many documents take a small value. More concretely, function idf is defined as follows:

$$\mathsf{idf}(t) \quad = \quad \log(N/n_t) \tag{1}$$

where N is the total number of documents, and n_t is the number of documents containing t. By definition, a popular word which is contained in many documents takes a small value close to zero, and conversely, a rare word which is contained in few documents takes a large value.

Function tf-idf, which is intended to represent the importance of t in document d, is formally defined as follows:

$$\phi(t, d) \quad \stackrel{\text{def}}{=} \quad \mathsf{tf}(t, d) \times \mathsf{idf}(t).$$

Using such notions, the weight of the i^{th} word in document d is determined as:

$$w_i(d) \quad \stackrel{\text{def}}{=} \quad \phi(t_i, d),$$

and the similarity between two documents a and b is defined as follows:

$$sim(a, b) \quad \stackrel{\text{def}}{=} \quad \frac{\sum_{i=1}^{m} \{w_i(a) \times w_i(b)\}}{\sqrt{(\sum_{i=1}^{m} w_i(a)^2) \times (\sum_{i=1}^{m} w_i(a)^2)}}.$$

The vector corresponding to a query is defined in a similar way; i.e., it is defined as

$$V(q) = (t_1, w_1(q); \ldots; t_j, w_j(q); \ldots; t_m, w_m(q)),$$

where $w_j(q)$ denotes the weight of word t_j in the query which takes value one if it is contained in q and takes value zero if it is not contained in q.

D. K-means Method

K-means method is a common technique to realize a clustering of set D. The method proceeds as follows:

1) At first, it randomly selects k documents from D, and regards those documents as the centers of clusters.
2) Each of the remaining documents is assigned to a cluster whose center is closest to him in the coordinate space.
3) For each cluster, it recalculates the center of the cluster (typically a document closest to the centroid of a clueter is selected as the center of the cluster), and excludes the other (non-center) documents from the clusters.
4) Repeat the above two steps until no further update is observed.

The time complexity of the above procedure is $O(Nkt)$, where N is the number of documents in D, k is the number of clusters, and t is the number of repetitions which depends on the spatial distribution of documents in the coordinate space. Note that in general, values k and t are very small compared with value N.

III. Related Work

This section describes the basic idea of Interested Mining (IM, for short) proposed by Tan *et al.* in [8].

A. Clustering of Documents

The IM method introduces the notion of *the sense of direction* to the flooding of query messages in fully distributed P2Ps. An outline of the scheme is described as follows: Let D be a set of documents held by a peer. At first, each peer calculates a vector representation of each document in set D (see Section II-B), and conducts a clustering of them (see Section II-D). At this time, given set of documents D is partitioned into several clusters. Let $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ be a set of such clusters. It then extracts a set of words characterizing each cluster, and by using such words, it determines a **characteristic vector** of C_i as

$$V(C_i) \quad = \quad (t_1, w_1(C_i); \ldots; t_j, w_j(C_i); \ldots; t_m, w_m(C_i)).$$

for each C_i. More concretely, the weight $w_j(C_i)$ of word t_j in cluster C_i is determined as follows:

$$w_j(C_i) \quad = \quad \frac{(\log f_j + 1.0) \times \log\left(\frac{N_i}{n_{i,j}}\right)}{\sqrt{\sum_{j=1}^{m} \left\{(\log f_j + 1.0) \times \log\left(\frac{N_i}{n_{i,j}}\right)\right\}^2}},$$

where f_j is the number of occurrences of word t_j in cluster C_i, N_i is the number of documents in C_i, and $n_{i,j}$ is the number of documents in C_i containing word t_j. After calculating weight for all words contained in C_i, we select m words with m largest weights, and use them as the elements of vector $V(C_i)$.

B. Construction of Overlay Network

Now, the set of documents held by each peer is partitioned into k clusters, each of which is associated with a characteristic vector in the coordinate space. Let $\mathcal{V}_u$ be a set of k characteristic vectors associated with peer u. The IM method tries to exchange such set of vectors among nearby peers in the underlying P2P overlay. Such overlay is constructed in a random manner in the original paper [8] as in Gnutella. More concretely, a newly arrived peer joins the network by randomly selecting a fixed number of peers as the set of neighbors, and by establishing a logical link (i.e., connection) to each of them.

Suppose that a new peer v selects a set of peers U as its neighbors. After connecting to all peers in U, peer v sends a message containing $\mathcal{V}_v$ to all peers in U. After receiving it, each peer $u \in U$ registers a pair of vectors in its local memory if the similarity between received vectors and its own vectors exceeds a predetermined threshold. For example, suppose that peer u is associated with a set of characteristic vectors $\mathcal{V}_u = \{V_1, V_2, V_3\}$, and it receives a set of vectors $\mathcal{V}_v = \{V_4, V_5, V_6\}$ from peer v (recall that each vector characterizes a cluster of documents). For each pair of vectors contained in set $\mathcal{V}_u \times \mathcal{V}_v$, peer u evaluates the similarity of those vectors. Suppose that $sim(V_1, V_5)$ exceeds the threshold. Then, peer u stores the following data to its local memory:

$$\langle V_1, V_5, \text{IP address of } v\rangle.$$

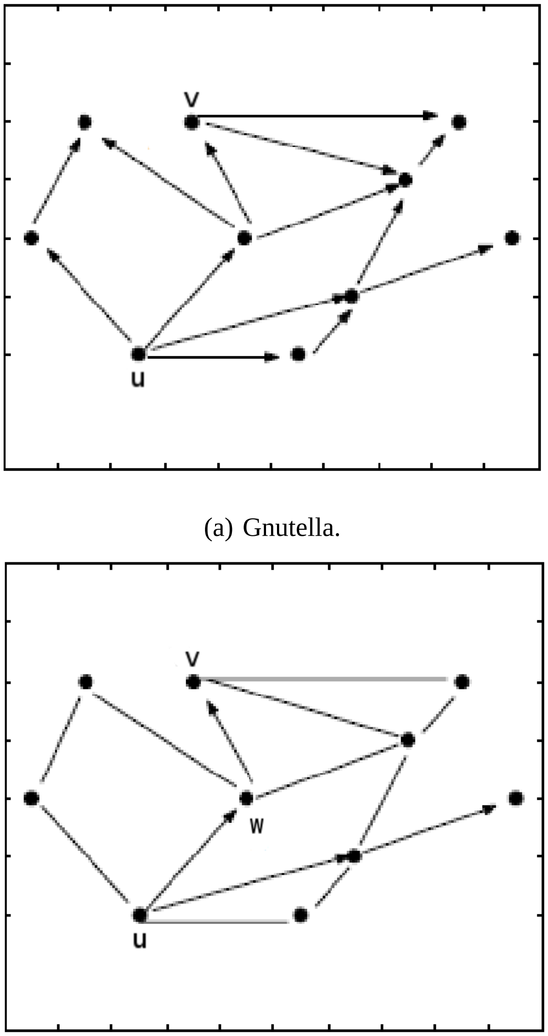
After that, peer u notifies $\mathcal{V}_u$ to its counter part v, and v stores tupple $\langle V_5, V_1, \text{IP address of } u\rangle$ to its local memory to realize a bidirectional logical link between u and v.

C. Selective Query Forwarding

Suppose that peer u wishes to find a file in a P2P network. At first, it generates a query concerned with the request by selecting keywords characterizing the request. It then refers to $\mathcal{V}_u$ to find a cluster whose characteristic vector is closest to the given query (more concretely, it evaluates the similarity of the query to each of characteristic vectors, and identifies a vector whose similarity is the largest). Let $V^* \in \mathcal{V}_u$ be the selected vector. It then refers to a table stored in its local memory to identify a (sub)set of neighbors to have a cluster similar to V^*, and to transmit the query only to those identified peers. After receiving a query from a neighbor, a peer v conducts a similar operation before forwarding the query to the next peers; i.e., it identifies a cluster closest to the received query, and forwards it only to those peers to have a cluster similar to the identified cluster.

D. Example

We explain the behavior of the IM method using a concrete example. Figure 2 illustrates the query propagation in Gnutella and the IM method. Figure 2 (a) illustrates the behavior of Gnutella, in which a requesting peer u transmits a query to all of its four neighbors, and a peer who received a copy of the query similarly forwards it to all of its neighbors. Such a query propagation (i.e., flooding of a query) is repeated until the number of hops of each copy of the query exceeds a predetermined threshold, i.e., it is controlled by setting a TTL (Time-to-Live) to the transmitted query. If peer v has a file requested by u, it replies this fact to peer u after receiving a query from its neighbor, and in this example, it is possible to reach v by setting TTL to at least two. However, such



(a) Gnutella.

(b) Interest Mining method.

Fig. 2. Query propagation in conventional methods.

an uncontrolled query propagation causes a large number of redundant query transmissions before finding a target peer v, e.g., several peers receive a copy of the query although they do not have a file relevant with the query and they are not on a path from u to v (in fact, when we set the TTL to three, the number of query transmissions becomes 14).

On the other hand, by adopting the IM method, we could significantly reduce the number of query transmissions as shown in Figure 2 (b), where we assume that peer w has a cluster similar to a cluster held by u which is closest to the given query. In fact, by restricting the recipient of the query to such peer w, the number of query transmissions conducted before finding a target peer v is reduced to four.

IV. Proposed Method

A. Outline

Although it significantly reduces the number of redundant message transmissions, the IM method has a serious drawback such that a requesting peer must be connected with a target peer through a path consisting of peers who have similar clusters with the given request. If there is no such path connecting to the target peer, a query issued by a requesting peer never reaches the target peer under the IM method. Since a newly arrived peer selects arbitrary peers as its neighbors, it is not guaranteed that such random set always contains a peer which has a cluster similar to *any* query issued by the peer. This implies that, in the worst case, a newly joined peer does not find a neighbor to which a given request should be transmitted.

The basic idea of our proposed scheme to overcome such drawback of the IM method is quite simple. We try to "reconfigure" the overlay network in such a way that a requesting peer is adjacent to a peer which has a similar cluster with the given query before transmitting a query in the IM method. In order to efficiently acquire the information on peers who have similar cluster with the given query, we introduce the notion of *community of peers* and *broadcast of advertisement* described below.

B. Community

In the proposed scheme, we explicitly construct communities of peers each of which is characterized by a characteristic vector, as in the clustering of documents in the original IM method. Each community has a peer called **master peer** which takes a responsibility for managing the members of the community. Each peer in the community knows the master peer, and the master peer knows all members of the community; i.e., it defines another logical network to have a star-like shape centered at each master peer.

In addition to the management of the members, the master of a community plays several roles in the proposed scheme; i.e., maintenance of a characteristic vector of the community, determination of Liaison peer to the community, and broadcast of an advertisement. Liaison peer is an access point of the community (i.e., to join the community, each peer first sends a message to the Liaison peer). Characteristic vector of a community is calculated from queries issued by the peers participating to the network which lead to the join of the requesting peer to the community. More concretely, 1) when a peer wishing to transmit a query to its neighbors joins a community, it sends the query to the master of the community with its IP address, in order to notify that "by which query it arrives at the community." 2) The master periodically accumulates the frequencies of words contained in the received queries, selects m highest frequency words among them, and regards those m words as the elements of the characteristic vector of the community (in other words, characteristic vector of a community is determined by a voting of peers participating to the community).

C. Broadcast of Advertisement

The master of a community periodically broadcasts an advertisement to all peers in the network in order to make public the existence of the community. An advertisement consists of the following fields:

- IP address of the master peer of the community and its backup peer (backup peer substitutes the master peer if the master crashes or leaves the system)
- Characteristic vector of the community
- Unique ID of the community
- IP address of the Liaison peer of the community

Concrete way of the broadcast is as follows:

- The master periodically broadcasts an advertisement.
- If and only if the number of peers which newly joined the community after the last broadcast is smaller than a predetermined threshold, it skips the next broadcast.
- Each peer keeps received advertisements for a moment, and expires them according to the FIFO (First-In First Out) rule.

D. Procedure

Procedure for the reconfiguration of the overlay is described as follows:

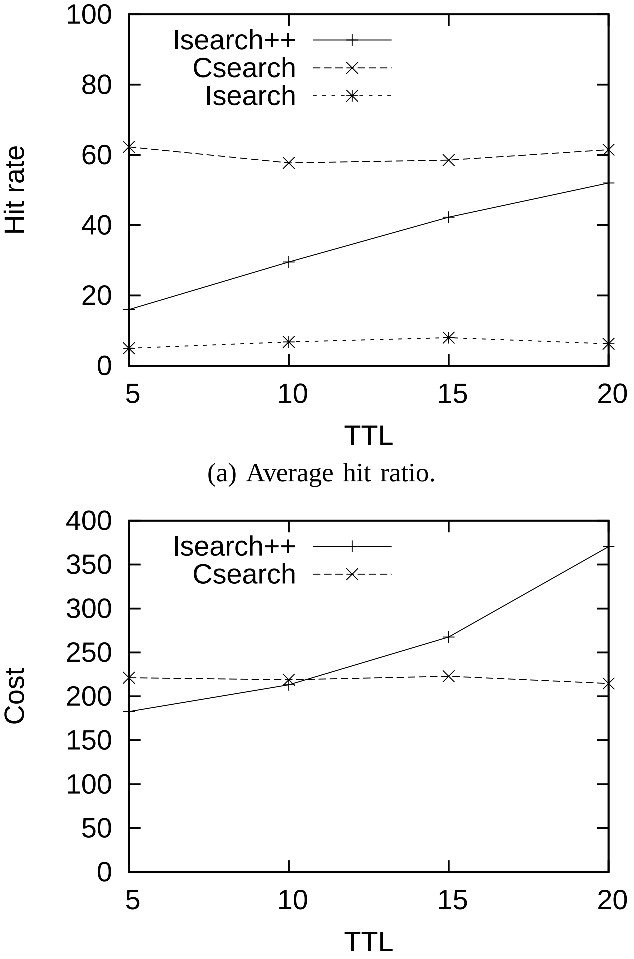
1) Before transmitting a query, each peer checks whether or not it has received an advertisement from communities to have a characteristic vector close to the query.
2) If it has received such advertisement and it is not expired yet, it connects to the Liaison peer of the community using IP address contained in the advertisement. Even if it has no such advertisement, if a neighbor of the peer has such advertisement, it connects to the community using the information held by the neighboring peer.
3) After that, the peer starts a selective query transmission as in the original IM method.

V. Experiment

A. Setup

We evaluate the performance of the proposed scheme by simulation. In the simulation, we evaluate the average hit ratio and the average number of message transmissions per request. In order to clarify the effect of the proposed scheme, we consider the following simple model of the similarity of documents:

1) We partition the set of all documents stored in the network into several clusters by designating the name of the corresponding cluster for each document.
2) The search of a file starts by designating the class of the target file; e.g., if it wishes to find a file in the third class, it transmits a query designating the class, and such query will be forwarded to the target peer through intermediate peers to have a document in the third class (note that in our setting, such peer has a cluster of documents characterized by the third class).
3) Each community of peers is characterized by the class of documents held by the peers, where each peer can belong to several communities.



(a) Average hit ratio.

(b) Average number of message transmissions.

Fig. 3. Result of experiments.

Parameters used in the experiments are determined as follows: The number of peers is fixed to 5000. The degree of each peer follows a power law, i.e., we assume that it is a *scale-free* network, and the frequency of classes contained in the queries follows the Zipf's law.

In the following experiment, we consider the following simple extension of the IM method as a competitor of our proposed method: The set of neighbors to which a received query should be forwarded is determined by referring to the information at distance k instead of referring to its immediate neighbors. In the following, we will illustrate the result for $k = 3$, since the hit ratio increases by increasing k, but it saturates around $k = 3$.

B. Result

The result of experiments is illustrated in Figure 3, where (a) and (b) illustrate the average hit ratio and the average number of message transmissions, respectively. In those figures, Isearch represents the IM method, Isearch++ represents the simple extension of the IM method, and Csearch represents the proposed scheme. As shown in Figure 3 (a), the hit ratio of Isearch is apparently worse than that of the other two schemes; e.g., it is 19% of Isearch++ and 11% of Csearch. Thus in the following, we will merely compare Isearch++ and Csearch. The figure indicates that when the cost of Isearch++ is not higher than the cost of Csearch, the hit ratio of Csearch beats the hit ratio of Isearch++. In other words, it holds either: 1) the cost of Csearch is lower than Isearch++, or 2) the hit ratio of Csearch is higher than Isearch++; i.e., Csearch is better than Isearch++ in terms of the cost/performance ratio.

The reason of the above phenomena could be explained as follows. Additional cost of Csearch is due to the broadcast of advertisements, and that of Isearch++ is due to the collection of information within three $(= k)$ hops. If it does not conduct a reconfiguration of the network after finding a target peer through the advertisements, the cost of Csearch significantly increases compared with Isearch++, while it does not increase the hit ratio of the scheme. However, a "reconfiguration" of the network drastically increases the performance of the scheme, as shown in the figure. In fact, reconfiguration of a network has several advantages for the IM method, such as: 1) it changes the structure of the network in such a way that peers holding similar documents come to a close position, 2) the distance to a target peer could be reduced by inserting a shortcut to the Liaison peer, and 3) it disconnects unnecessary links connecting peers holding completely different documents.

VI. Concluding Remarks

This paper proposed a scheme to improve the quality of a file search in the Interest Mining proposed by Tan *et al.* The basic idea of the proposed scheme is to reconfigure the underlying P2P overlay in such a way that the connectivity of peers with respect to their "interest" is kept to be sufficiently high. In order to realize such reconfiguration of the network in an efficient manner, we introduce the notion of communities and periodical broadcast of advertisement of the communities. The result of experiments indicates that the proposed scheme certainly overcomes the drawback of the Tan's method.

The following problems are left as future problems:

- In the proposed scheme, a requesting peer connects to the unique Liaison peer of a community. However, an appropriate position in the community to be connected varies depending on the issued query. Thus, we need to modify the scheme in such a way that each community is associated with several Liaison peers, and each peer selects one of such Liaison peers.
- In the proposed scheme, advertisement is broadcast to all peers in the network. We could significantly reduce the cost for such broadcast by allowing selective forwarding of advertisement as in the selective forwarding of queries in the Interest Mining method.
- Old advertisement should be expired due to the limitation of the local storage. In the proposed scheme, such an expiration is conducted according to the FIFO (First-In First-Out) rule independent of the interest of the peer. However, it would be meaningful to consider the similarity of advertisements to the queries in determining advertisements to be expired.

References

[1] J. Aspnes and G. Shah. Skip Graphs. In *Proc. 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 384–393, 2003.

[2] Available at http://ja.wikipedia.org/wiki/Gnutella.

[3] J. Li and S. T. Vuong. An Efficient Clustered Architecture for P2P Networks. In *Proc. 18th International Conference on Advanced Information Networking and Applications (AINA)*, pages 278–283, 2004.

[4] X. Luo, Z. Qin, J. Geng J. Luo. IAC: Interest-Aware-Caching for Unstructured P2P. *Proc. IEEE SKG*, 2006.

[5] Available at http://www.napster.com/.

[6] A. I. T. Rowstron and P. Druschel. Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems. In *Proc. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pp.329-350, November 2001.

[7] I. Sotica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In *Proc. ACM SIGCOMM*, pp.149-160, August 2001.

[8] Y. Tan, Z. Chen, Y. Lin, T. Dong. Research and Implementation on Routing Scheme Based on Interest Mining in unstructured P2P Systems. *Proc. IEEE WAIMW*, 2006.

[9] B. Yang, and H. Garcia-Molina. Designing A Super-peer Network. In *Proc. International Conference on Data Engineering (ICDE),* March 2003.