Dynamic Reorganization of P2P Networks Based on Content Similarity

Takuya Yamaguchi
Graduate School of Science and
Engineering, Saitama University
Saitama, Japan
takuya@ss.ics.saitama-u.ac.jp

Noriko Matsumoto
Graduate School of Science and
Engineering, Saitama University
Saitama, Japan
noriko@ss.ics.saitama-u.ac.jp

Norihiko Yoshida
Graduate School of Science and
Engineering, Saitama University
Saitama, Japan
yoshida@ss.ics.saitama-u.ac.jp

***Abstract*—A unstructured P2P network does searching by packet forwarding which has some problems: hit ratios are low, and the network is filled with packets. A structured P2P network based on the distributed hash table (DHT) solves these problems. However, it is restricted to keyword search. This paper proposes a P2P network which reorganizes itself dynamically, aiming at search efficiency of the structured P2P and the search flexibility of the unstructured P2P at the same time. We define similarity of contents based on the folksonomy in social networks, and make the network update its links dynamically based on the content similarities. By simulation-based experiments, we confirmed improvements of query hits in this P2P network.**

Keywords-P2P; content-based reorganization; folksonomy

I. Introduction

P2P networks have some categories according to content search methods. An unstructured (pure) P2P such as Gnutella, which uses flooding-based search, has advantages in regards to network flexibility and robustness. However, flooding causes network congestion. Some techniques have been proposed to suppress the congestion such as Expanding Ring [1] and Random Walks [1]. However, they have no concern with the properties of contents.

Usually, the time-to-live (TTL) parameter is used to control flooding. It specifies the maximum number of forwarding hops of search queries. The smaller the TTL is, the less the congestion is. However, the smaller TTL leads to the lower hit ratio (or success ratio) as well. A peer node which emits a search query (searcher) is assumed to have a similar interest to a peer which has the target content. This means that peers with similar interests are better located nearer in order to suppress network congestion and to assure hit ratio at the same time.

The Distributed Hash Table (DHT) is another category of P2P, which suffers no network congestion. However, a DHT-based P2P network must have a strictly structured topology, and consequently is prone to failure, costful in dynamic restructuring, and also search in the DHT is limited to exact matching in principle.

This paper proposes a P2P network with a restructuring function similar to a consensus formation theory [2]. The function simulates a group formation in social networks, and is to make groups of nodes with similar contents dynamically.

We begin with our observation on P2P content search.

- It is likely that the searcher has already some contents similar to the one being searched.
- It is likely that the searcher is always interested in the search keyword.
- It is likely that the searcher will be interested in related keywords in the future.

The interest of a peer must be inferred from the set of its contents. Our P2P network restructures itself based on the peers' similarity.

Hereafter, we introduce related works regarding network reconstruction in Section 2, and propose a reconstruction method based on similarity in Section 3. The simulation and consideration in a P2P network using our technique are shown in Section 4. Section 5 includes some concluding remarks.

II. Related Works

From the early days of P2P networks, there were some attempts to content-based retrieval and peer clustering. Lu and Callan (2003) [3] and Wang and Yang (2006) [4] proposed such mechanisms on top of a super-peer-based hybrid P2P network, in which a super peer acts as an index server for contents. On the other hand, Tang, et al. (2003) [5], Kacimi and Yetongnon (2008) [6], and Tirado, et al. (2010) [7] proposed a semantic overlay network over a DHT-based structured P2P network. We have taken an alternative approach. A P2P network itself is an overlay on top of a physical network. Therefore, instead of constructing a content-based overlay on top of a P2P overlay, we reorganize a P2P overlay to be a content-based overlay as well. Vazirgiannis, et al. (2006) [8] proposed an approach similar to ours. However, their work stayed at a preliminary stage.

Sripanidkulchai, et al. (2003) [9] proposed a content allocation scheme based on interest proximity (or similarity), and Voulgaris, et al. (2004) [10] extended it towards a semantic overlay. Our originality lies in aggregation of content similarities to get node similarities, reducing the network traffic.

Below are some topics related to P2P network reorganization.

A. Reorganization for Reliability Improvement

Simple Trust Exchange Protocol (STEP) [11] is a protocol for P2P reorganization to improve network reliability. STEP aims at taking care of a normal peer by eliminating free riders, which do not provide contents, but only consume, and malicious peers which distribute inaccurate contents.

A receiver evaluates service provided by a sender, and issues a "token" with rating of the service. Each node exchanges

ISBN: 978-1-61208-292-9

the tokens by messages called "knowledge" with its neighbors periodically, and sums up single evaluations to make a more precise evaluation. Then, each node decides how tightly it keeps its link with the evaluated node, and any node with a "bad" evaluation is eliminated from the network in this manner.

B. *Network Reorganization by Consensus Building*

A social network consists of various groups. People in a group usually share the same interest and/or opinion. Holme and Newman [2] tried to model this group formation under agreement and opinion adjustment.

The initial network has N nodes and M random links. Each node has one out of G opinions. This network repeats the below every unit time.

1) Choose one node i at random.
2) If the node i does not have a link, do nothing. Otherwise, choose one link at random, which is to connect to the node j.
    - Upon probability of ϕ, reconnect the link to a node with the same opinion as i.
    - Upon probability of $1-\phi$, change i's opinion to the same as j's.

They simulated the model and confirmed that clusters emerged in the network according to the opinions.

III. Method

Our proposed method is divided into the similarity calculation method and the network reconstruction method. Below we present them respectively.

A. *Similarity Calculation Method*

Network reconstruction is done based on the peer similarity. Each peer calculates the peer similarity value based on the content similarity value of the peer's contents. The content similarity value is calculated based on the content information exchanged between peers. If a peer must exchange and calculate the similarity value for all the contents it contains, it would cause severe network traffic and overhead. Therefore, we introduce "Virtual Typical Content" (VTC) for each peer, whose similarity value is an aggregation of the values of all the contents of the peer. A peer's VTC represents the tendency of the content which the peer has. We may say, looking at the VTC, we can get the "taste" of the peer.

The purpose of VTCs is to reduce network traffic and overhead drastically. It causes significantly lower traffic to exchange only VTCs between peers than to exchange all the contents on peers. Each peer has the predefined number of VTCs (not necessarily one) regardless of the number of the contents it really has. This method is particularly effective in a network composed of poor performance peers and narrow band communication.

The similarity value of the VTC is calculated from the similarity of contents. It is difficult and resource consuming to get the similarity of contents by analyzing the contents. Therefore, we use folksonomy [12] instead.

1) Folksonomy: Folksonomy is a sort of information classification. Users attach tags to contents. A tag is typically a keyword which the users think represents the meaning or nature of the contents. Then the contents are classified based on a collection of tags (Fig. 1).

Recently, this method is getting widely used on the Internet, for example, as social bookmarks. Although having some problems, i.e. tags cannot handle synonyms, and tags may be unsuitable intentionally, folksonomy is promising because of its significantly lower cost compared to automatic keyword distillery using "TF-IDF" for example.

In our method, content suppliers give tags to each content, and the system calculates similarity values from tags.

2) Making Virtual Typical Content: Virtual typical contents are created as follows:

Let C, T be sets, and (M, m) be a multiplex set. The number of VTCs is N, and the max number of tags assigned to VTC is M.

1) Let a peer have contents $C = \{c_1, c_2, \ldots, c_n\}$, and let each content c_x have tags T_{c_x}.
2) Calculate $M_1 = \bigcup_{c_k \in C} T_{c_k}$.
3) Calculate the most common tag t_{max} that is $m_1(t_{max}) = max(\{m_1(x) \mid x \in M_1\})$ (Fig. 2).
4) Find a content having the most common tag $C_{max} = \{c_x \mid t_{max} \in T_{c_x}\}$ (Fig. 3).
5) Calculate $M_2 = \bigcup_{c_k \in C_{max}} T_{c_k}$.
6) Calculate $T_{VTC} = \{t \mid t \in M_2, m_2(t) \geq \alpha\}$. But, α is decided suitably about $n(T_{VTC}) = M$ (Fig. 4).
7) Making VTC which assigned T_{VTC} (Fig. 4).
8) Calculate $C = C - C_{max}$.
9) If $C = \emptyset$ or the number of VTC is N, the loop terminates. Otherwise, the loop goes back to 2) and continues.

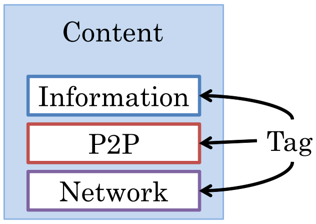


Figure 1. Assignment of tags.

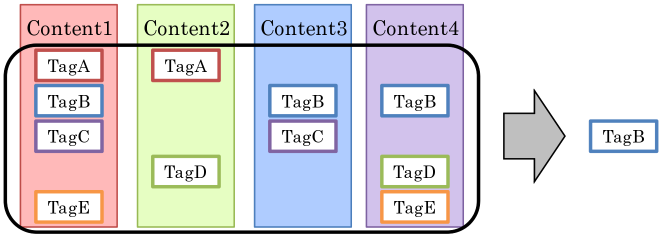


Figure 2. Selection of the most common tag.

ISBN: 978-1-61208-292-9

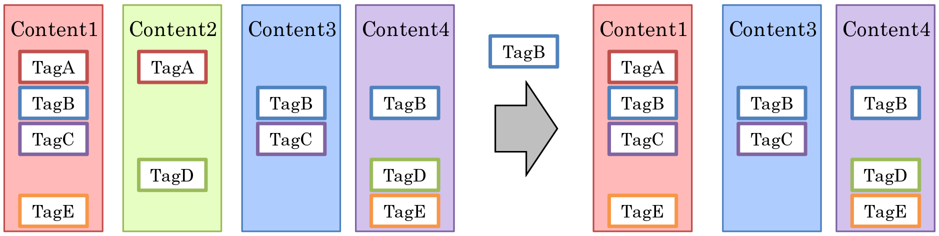


Figure 3. Selection of contents with the most common tag.

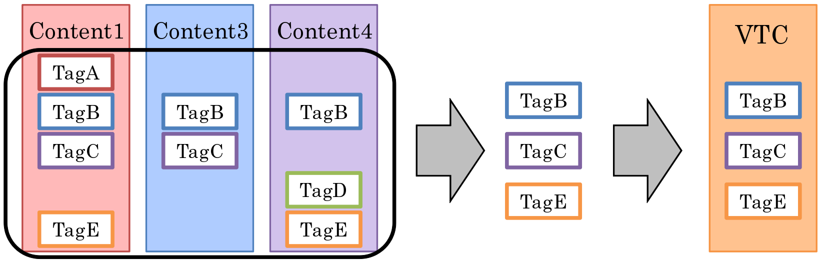


Figure 4. Making of a virtual typical content.

3) Content Similarities: Content suppliers attach tags to each content. The content similarity is calculated from an agreement ratio of these tags.

Let content A be assigned tags $T_A = \{T_{A1}, \ldots, T_{AN}\}$, and content B be assigned tags $T_B = \{T_{B1}, \ldots, T_{BM}\}$. The similarity value R_C between content A and B is defined as follows:

1) If $T_A = \emptyset$ or $T_B = \emptyset$, then $R_C = 0$
2) Otherwise, (i.e. $T_A \neq \emptyset$ and $T_B \neq \emptyset$),

$$R_C = \frac{n(T_A \cap T_B)}{min(n(T_A), n(T_B))} \tag{1}$$

where $n(X)$ means the number of elements in the set X.

Therefore, the content similarity satisfies the below properties:

1) If $T_A \cap T_B = \emptyset$ then $R_C = 0$.
2) If $T_A \subseteq T_B$ or $T_A \supseteq T_B$ then $R_C = 1$.
3) The domain of R_C is $0 \leq R_C \leq 1$.

4) Peer Similarities: We calculate the peer similarity value R_P from the content similarity value as follows. We specify the number of VTCs and the number of tags attached to each VTC, given a set of contents on a peer, and create VTCs. Then, the peer calculates content similarity values for all the VTCs, and make the maximum value of the outcome R_C as the peer similarity value R_P.

B. Reconstruction Method

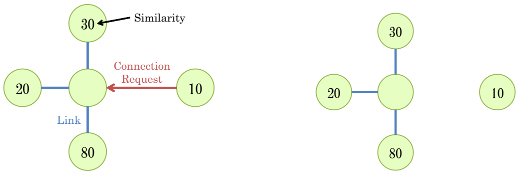
Network reconstruction is done by reconnecting network links, using a technique similar to the neighbor peer replacement technique in STEP.

Two peers connected by a link are called neighbor peers. For each peer, let there be the predefined maximum number of neighbor peers. Each peer can have this number of links at the most.

If a peer $P1$ receives a new connection request from a peer $P2$ which is not a neighbor peer, $P1$ approves or denies the request as follows:

1) If $P1$ does not have the maximum number of neighbor peers, the request from $P2$ is approved and a link between $P1$ and $P2$ is created.
2) If $P1$ already has the maximum number of neighbor peers, similarities to all neighbor peers as well as $P2$ are calculated.
    a) If the similarity to $P2$ is lower than any of the similarities to all the neighbor peers, the request is denied (Fig. 5).
    b) Otherwise, a link to a peer whose similarity is the lowest among the neighbor peers is discarded, and the request to create a link to $P2$ is approved (Fig. 6).

Each peer does the above, and some cluster of peers with the high similarities emerges in the network autonomously.



Connection request

Result

Figure 5. Connection denial.

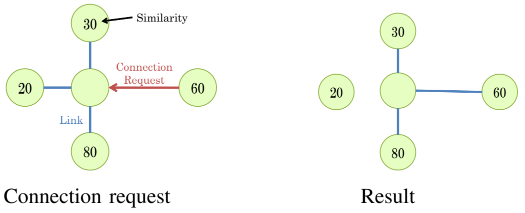


Figure 6. Connection approval.

IV. Simulation

We built a simulator which constructs virtual P2P networks on a single computer, and performed some experiments and evaluation.

A. Simulation Model

As described in Section 1, each peer is supposed to have some tendency, or deviation, in its interests. The simulator reflects this as follows.

Each peer is assigned an unique integer of 1 or more, PID, as its identification number. A tag is assigned also an integer of 1 or more, although a tag in the real world would be some keyword. Peers in the network are grouped in the manner that a peer having such a PID that $(k-1) \times M + 1 \leq \text{PID} \leq k \times M$ belongs to the group G_k. Peers in a group G_k has an interest in such a tag t that $(k-1) \times M + 1 \leq t \leq k \times M$. Let p be a search deviation ratio. With the probability p, a peer searches a tag within the interests of G_k. Otherwise (with the probability $1-p$), a peer searches a random tag. Likewise, Let $p\prime$ be a content deviation ratio. With the probability $p\prime$, each content on a peer within G_k has a tag within the interests of G_k. Otherwise, a content has a random tag.

Some major parameters in the simulation are summarized in Table I. We performed simulations for networks in which the number of peers are 100, 200, and 300, and for a case with network reconstruction and a case without reconstruction. We repeated simulations five times.

We define a unit time of the simulation as a period necessary to forward a message from a peer to its neighbor peer. Each peer does all the necessary computation and this one hop communication within the unit time. We call the unit time "second" in this simulation, and one simulation lasted for ten hours.

TABLE I. Simulation Parameters

Parameter	Value
Max number of neighbor peers	4
Time-to-Live (TTL)	4
Number of VTC	10
Max number of tags assigned to VTC	6
Number of peers in a group	10 (20 in case of 300 peers)
Search deviation ratio	80%
Content deviation ratio	80%
Minimum connection time	3 second
Disconnection Probability	0.2%
Disconnection interval	60 second

B. Simulation Results

1) *The Number of Search Hits:* The number of average hits (QueryHit) to one query is shown in Fig. 7, Fig. 8, and Fig. 9 for the cases of 100, 200, and 300 peers respectively. The number of hits in search is shown to be improved in the network with reconstruction compared to the network without reconstruction under the same small value of TTL.

2) *Overhead of Similarity Calculation:* Table II shows the average number of VTCs transfer per peer per one hour during similarity calculation. It must cause message overhead to the network by the proposed method. Real overhead to an actual

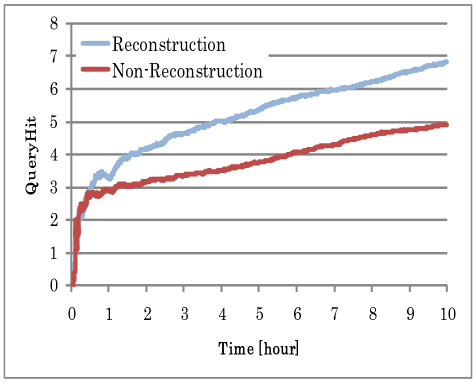


Figure 7. Averages of QueryHits (100 Peers).

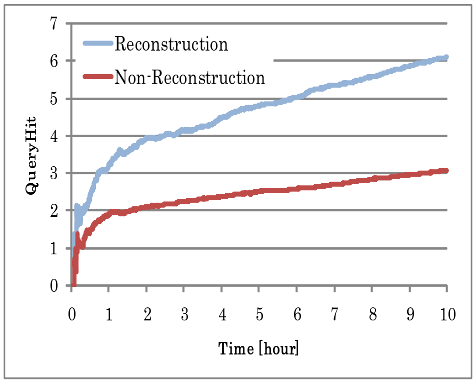


Figure 8. Averages of QueryHits (200 Peers).

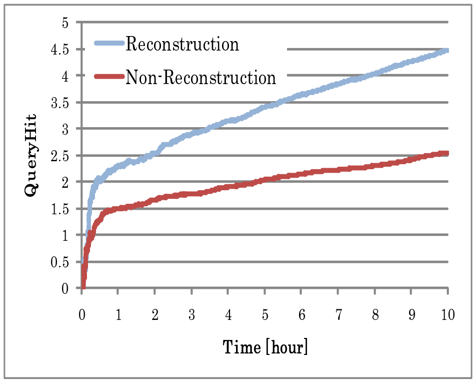


Figure 9. Averages of QueryHits (300 Peers).

TABLE II. THE NUMBERS OF VTCS

	100	200	300
First	157.22	204.24	184.58
Second	188.04	177.68	175.96
Third	196.21	191.04	201.41
Fourth	154.32	195.68	168.34
Fifth	167.45	200.40	184.09
Average	172.65	193.81	182.88

Table III. COMPARISON OF VTCS AND QUERIES

	VTC	Query
First	157.22	3211.68
Second	188.04	3356.04
Third	196.21	3240.48
Fourth	154.32	3152.99
Fifth	167.45	3275.56
Average	172.65	3247.35

network would be a product of this average number and the size of a VTC message. However, this size must be small, because a VTC message only contains tag information, and comparable to the size of a search query message, and much smaller than the size of a content.

Table III shows comparisons of the number of VTCs and the number of queries per peer per hour in the 100 peer network. The number of VTCs is about $1/20$ of the number of queries. This 5% overhead of VTC messages added to query messages in the network traffic is supposed to be acceptable compared to the traffic for content delivery.

We suppose this overhead of VTC messages could be reduced. These results shown here are obtained out of the worst cases in the sense that the numbers of VTCs are the largest in these networks. More than one VTC messages from the same peer could be aggregated into a single message. Also, caching of VTC messages could reduce the network traffic.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a reconstruction method of P2P networks based on content similarity. The proposed method uses tags to each content in a peer, makes virtual typical contents (VTCs) representing interests of the peer from the tags assigned to the contents, calculates similarity values from VTCs, and updates links between peers according to similarity values. This reorganization improves success ratios of queries even if the time-to-live (TTL) value is unchanged. In other words, we could make the time-to-live value smaller to achieve the same success ratios, which leads to lower network traffic.

We are still at the starting point toward practical implementation and deployment of this design. Future work includes some improvement for selecting a peer to whom a connection request is sent using the similarity values. In the current design, a connection request is sent to an arbitrary peer. This improvement must bring more efficient clustering. Another work would be aggregation of VTC messages to query messages to reduce the overhead of VTC messages to the network traffic as well as to convey the VTC messages farther than its neighbors.

REFERENCES

[1] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, “Search and Replication in Unstructured Peer-to-Peer Networks”, Proc. ACM 16th Int. Conf. on Supercomputing, 2002, pp. 84–95.

[2] P. Holme and M. E. J. Newman, “Nonequilibrium Phase Transition in the Coevolution of Networks and Opinions”, Physical Review E, Vol. 74, 056108, November, 2006, 5 pages.

[3] J. Lu and J. Callan, “Content-Based Retrieval in Hybrid Peer-to-Peer Networks”, Proc. ACM 12th Int. Conf. on Information and Knowledge Management, 2003, pp. 199–206.

[4] J. Wang and S. Yang, “Content-based Clustered P2P Search Model Depending on Set Distance”, Proc. 2006 IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology, 2006, pp. 471–476.

[5] C. Tang, Z. Xu, and S. Dwarkadas, “Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks”, Proc. ACM SIGCOMM, 2003, pp. 175–186.

[6] M. Kacimi and K. Yetongnon, “Content-Based Information Routing and Retrieval in Cluster-based P2P Overlay networks”, Proc. 2008 IEEE Int. Conf. on Signal Image Technology and Internet Based Systems, 2008, pp. 70–77.

[7] J. M. Tirado, D. Higuero, F. Isaila, J. Carretero, and A. Iamnitchi, “Affinity P2P: A Self-Organizing Content-Based Locality-Aware Collaborative Peer-to-Peer Network”, Computer Networks No. 54, 2010, pp. 2056–2070.

[8] M. Vazirgiannis, K. Norvag, and C. Doulkeridis, “Peer-to-Peer Clustering for Semantic Overlay Network Generation”, Proc. 6th Int. Workshop on Pattern Recognition in Information Systems, 2006, 10 pages.

[9] K. Sripanidkulchai, B. Maggs, and H. Zhang, “Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems”, Proc. IEEE INFOCOM, 2003, pp. 2166–2176.

[10] S. Voulgaris, A.-M. Kermarrec, L. Massoulie, and M. v. Steen, “Exploiting Semantic Proximity in Peer-to-Peer Content Searching”, Proc. 10th IEEE Int. Workshop on Future Trends of Distributed Computing Systems, 2004, pp. 238–243.

[11] I. Martinovic, C. Leng, F. A. Zdarsky, A. Mauthe, R. Steinmetz, and J. B. Schmitt, “Self-Protection in P2P Networks: Choosing the Right Neighbourhood”, Proc. 1st Int. Workshop on Self-Organizing System, 2006, pp. 23–33.

[12] A. Mathes, “Folksonomies - Cooperative Classification and Communication Through Shared Metadata”, LIS590CMC, University of Illinois Urbana-Champaign, December, 2004, 13 pages.