Similarity-Based Content Retrieval in Self-Organizing Peer-to-Peer Networks

Takuya Yamaguchi[1]**, Andrii Zhygmanovskyi**[1]**,**
Noriko Matsumoto[1] **and Norihiko Yoshida**[1*]

[1]*Graduate School of Science and Engineering, University in Saitama, Japan.*

Original Research Article

Abstract

This paper presents dynamic reorganization of peer-to-peer networks to make query routing and content retrieval efficient. The reorganization is conducted which use content similarity information. Unlike other related studies using semantic proximity, the method proposed in this paper relies on folksonomy, which gained wide use in figuring out content similarity in various social networks. The proposed method is designed primarily for flooding-based unstructured P2P networks, however it also can be applied to structured P2P networks such as Tapestry. Simulation-based experiments confirm the effectiveness of the proposed method.

Corresponding author: E-mail: yoshida@mail.saitama-u.ac.jp

1 Introduction

Peer-to-peer (P2P) networks have advantages in load distribution and failure resilience over conventional client/server networks. However, an unstructured (pure) P2P, which uses flooding-based search, causes network congestion. Some techniques have been proposed to deal with congestion such as Expanding Ring and Random Walks [1].

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.

Structured P2P, which uses a 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, therefore it is prone to failure and is costly in dynamic restructuring. Also search in the DHT is limited to exact matching in principle.

Our observation is that 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 as near as possible in order to suppress network congestion and to assure hit ratio at the same time. However, most P2P networks outlined above have no concern with the properties of contents.

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. Its essence was already presented in [3].

The technique potentially can be applied not only to unstructured P2P networks but also to structured P2P networks. This paper also discusses this possibility as well.

Hereafter, we introduce related works regarding network reorganization in Section 2, then propose a similarity estimation method in Section 3, and a reorganization method based on similarity in Section 4. Section 5 discusses applicability of this technique to structured P2P networks. The simulation and consideration in a P2P network using our technique are shown in Section 6. Section 7 includes some concluding remarks.

2 Related Works

From the early days of P2P networks, there were some works concerning content-based retrieval and peer clustering. Lu and Callan [4] and Wang and Yang [5] proposed such mechanisms on top of a super-peer-based hybrid P2P network where the super peer acts as an index server for contents, therefore becoming a single point of failure. On the other hand, Tang, et al. [6], Kacimi and Yetongnon [7], and Tirado, et al. [8] 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. [9] proposed an approach similar to ours. However, their work remained at a preliminary stage.

Sripanidkulchai, et al. [10] proposed a content allocation scheme based on interest proximity (or similarity), and Voulgaris, et al. [11] extended it towards a semantic overlay. Our originality lies in

the use of folksonomy and aggregation of content similarities to get node similarities, reducing the network traffic.

Raftopoulou and Petrakis [12] proposed a similarity-based clustering overlay on top of a P2P network. The cluster is rigid in the sense that its border is strict. Our scheme is more fuzzy, allowing a cluster to be more elastic. There were some other attempts as well. For example, Asvanund, et al. [13] proposed an idea to use an IR (information retrieval) technique at super-peer in their working paper; however, no reviewed paper based on it has been found. Chen, et al. [14] proposed an idea of semantics-based indexing scheme in their position paper although no result has been presented.

Below are some topics related to P2P network reorganization.

2.1 Reorganization for Reliability Improvement

Simple Trust Exchange Protocol (STEP) [15] 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.

Each node evaluates another node based on its service quality, and propagates its set of the evaluations, or ratings, to its neighbors periodically. Accordingly, each node obtains a set of ratings of a particular node evaluated by several nodes, and accumulates them to make a decision how tight its link should be to this particular node. In this manner, any node with a "bad" evaluation is eliminated from the network eventually [16].

2.2 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 using 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 following procedure every time unit.

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 connected to the node j.

   - With probability ϕ, reconnect the link to a node with the same opinion as i.
   - With probability $1 - \phi$, 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.

3 Similarity Estimation

Network reorganization is done based on the "peer similarity". Each peer calculates its peer similarity value based on the "content similarity" value of its 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 VTC is to reduce network traffic and overhead. 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 [17] instead.

3.1 Related Technology

3.1.1 Folksonomy

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

Recently, this method is getting widely used on the Internet, for example, for social bookmarks. Although there are 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 assign tags to each content, and the system calculates similarity values from tags.

3.2 Our Method

3.2.1 Establishing Virtual Typical Content

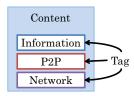Virtual typical contents are created as follows:
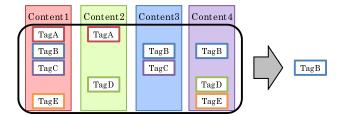

Figure 1: Assignment of tags.

Figure 2: Selection of the most common tag.

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 $M_1 = \bigcup\limits_{c_k \in C} T_{c_k}$.

3. Calculate the most common tag t_{max} that is $m_1(t_{max}) = max(\{m_1(x) \mid x \in M_1\})$ (Figure 2).

4. Find a content having the most common tag $C_{max} = \{c_x \mid t_{max} \in T_{c_x}\}$ (Figure 3).

5. Calculate $M_2 = \bigcup\limits_{c_k \in C_{max}} T_{c_k}$.

6. Calculate $T_{VTC} = \{t \mid t \in M_2, m_2(t) \geq \alpha\}$, where α is decided using the equality $n(T_{VTC}) = M$ (Figure 4).

7. Let VTC be the current value of T_{VTC} (Figure 4).

8. Calculate $C = C - C_{max}$.

9. If $C = \emptyset$ or the number of VTC is N, the loop terminates. Otherwise, the loop continues from 2).

3.2.2 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:

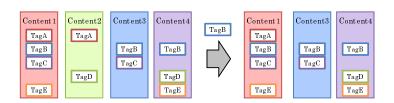1. If $T_A = \emptyset$ or $T_B = \emptyset$, then $R_C = 0$

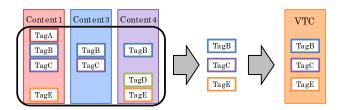Figure 3: Selection of contents with the most common tag.

Figure 4: Establishing a virtual typical content.

2. If $T_A \neq \emptyset$ and $T_B \neq \emptyset$,

$$R_C = \frac{n(T_A \cap T_B)}{min(n(T_A), n(T_B))} \tag{3.1}$$

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

Therefore, the content similarity satisfies the following 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 $0 \leq R_C \leq 1$.

3.2.3 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 set the maximum value of the outcome R_C as the peer similarity value R_P.

4 Reorganization of Unstructured P2P Networks

Network reorganization is done by reconnecting network links, using a technique similar to the neighbor peer replacement technique in STEP [15].

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 the number of neighbor peers for $P1$ does not exceed the maximum, 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 the neighbor peers, the request is denied (Figure 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 (Figure 6).

Provided that each peer does the above procedure, clusters of peers with the high similarities will emerge in the network autonomously.

5 Application to Structured P2P Networks

Similarity estimation based on Folksonomy and network reorganization described so far can be applied not only to unstructured P2P networks but also to structured P2P networks to improve efficiency. This paper uses Tapestry as a structured P2P network because of its flexibility.

Conventional structured P2P networks, including Tapestry, use hash functions which realize exact match between keywords only. Our method uses a Bloom filter, in particular a Spectral Bloom filter, made from tags of contents, which may be actual contents, or VTCs presented above. Content similarities are calculated using the spectral Bloom filters.

5.1 Related Technologies

5.1.1 Bloom Filter

A Bloom filter [18] is a data structure composed of a bit string, or a bit vector, designed to specify whether an element is present or not in a set. Each bit in the vector is initially set to zero. Given an element, it is fed to K hash functions to obtain K vector indexes, and bits at the indexes in the vector are set to one. The primitive Bloom filter has a drawback that false positive matches are possible.

There are some improvements and extensions to the primitive Bloom filter. A counting Bloom filter [19] is composed of a vector of fixed-size counters, not of bits, enabling element deletion. A spectral Bloom filter [20] is composed of a vector of variable-size counters.

5.1.2 Tapestry

Structured P2P networks include circular, grid, tree, and mesh structures. Tapestry [21] is a typical one of mesh-based structured P2P networks, and is an implementation of a graph called Plaxton mesh [22]. It has advantages in flexibility and robustness compared to other structured P2P networks.

Each peer in a Tapestry network is assigned a unique ID, and has connections to other peers based on the *nearness* of IDs. The connection information is stored in a *neighbor map* on each peer. A neighbor map has several levels, where level 1 has links to peers whose ID have nothing in common, level 2 has the first digit in common, and so on.
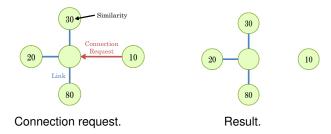

Connection request.　　　　　　Result.

Figure 5: Connection denial.
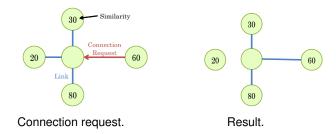

Connection request.　　　　　　Result.

Figure 6: Connection approval.

5.2 Our Proposal

Our network for similarity-based reorganization is a variant of Tapestry in the sense that each peer propagates tag information represented in the form of spectral Bloom filter, and the network reorganizes itself using the following information.

1. A peer makes a spectral Bloom filter (SBF) from hash values of all the tags assigned to its contents or VTCs. Then, it propagates the SBFs to neighboring peers (Figure 7). Each peer receiving the SBFs stores them into a table along with the link they come from. The table becomes a routing table based on Bloom filters.

2. When a query is issued, a SBF made from the query keyword is compared to each SBF in the routing table so as to find a suitable route. The comparison is done using Bloom filter similarity defined as:
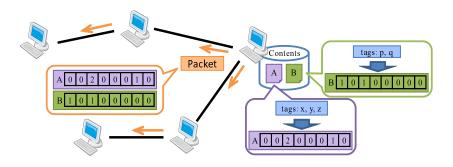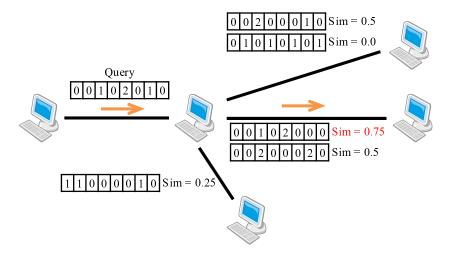

Figure 7: SBF distribution.

Figure 8: SBF-based routing.

$$Sim = \frac{\sum_{i=1}^{n} min(c_i,\ c_i')}{\sum_{i=1}^{n} c_i}$$

where c_i is an i'th counter in an SBF in the routing table, and c_i' is an i'th counter in an SBF of the query. Then the peer emits the query to the link with the maximum Sim in the routing table (Figure 8).

This forwarding is repeated until the query reaches a peer which has the target content.

3. If propagated SPFs are calculated not from actual contents but from VTCs, they can be used not only for routing but also for reorganization of the network. Namely, a peer receiving an SPF compares the SPF with its own SPF calculated from VTCs, and reconnect its links accordingly. This comparison is done using the same method mentioned above. In this way, the SPF-based routing table replaces the neighbor map in the original Tapestry.

6   Simulation-based Evaluation

6.1   Simulation Model

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

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 an integer of 1 or more as well, although a tag in the real world would be some keyword. Peers in the network are grouped in the following manner: a peer with a PID such that $(k-1) \times M + 1 \le \text{PID} \le k \times M$ belongs to the group G_k. Peers in a group G_k has an interest in a tag t such that $(k-1) \times M + 1 \le t \le k \times M$. 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.

Main parameters in the simulation are summarized in Table 1. We performed simulations for networks in which the number of peers are 100, 200, and 300, both for a case with network reorganization and without it. We repeated simulations five times.

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

Figure 9 and Figure 10 shows an schematic example of network reorganization, where Figure 9 is the network before reorganization, and Figure 10 is after. Thick lines indicate links connecting peers in the same group, and the number of them increases after reorganization.

6.2   The Number of Search Hits

The number of average hits (QueryHit) to one query is shown in Figure 11 (a), (b), and (c) 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 reorganization compared to the network without reorganization under the same small value of TTL.
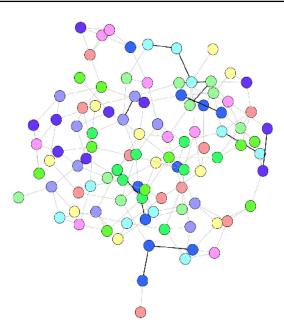
Figure 9: Network before reorganization.

6.3 Overhead of Similarity Calculation

Table 2 shows the average number of VTCs transfer per peer in one hour during similarity calculation. While the proposed method causes message overhead to the network which is a product of this average number and the size of a VTC message, this overhead is small comparable to the size of a search query message, because a VTC message only contains tag information, and is much smaller than the size of a content.

Table 1: 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

Table 2: 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 3 shows comparisons of the number of VTCs and the number of queries per peer in one hour for 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.

6.4 Experiment on Structured P2P Networks

We modified the simulator, and performed a small experiment on structured P2P networks. Main parameters for this are summarized in Table 4.

Our method for structured P2P networks is influenced by the range how far the Bloom filter (BF) is propagated in the network. This range of propagation is measured by the number of hops, or TTL, of the SBF message. Therefore, this experiment focuses on its influence. The experiment leaves out
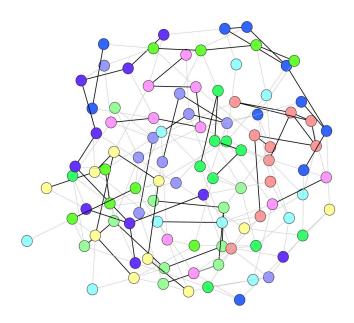

Figure 10: Network after reorganization.

Table 3: 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

all the original query routing of Tapestry, relying on the newly added routing alone, and measures the success ratio, or "hit rate", of queries against the TTL of BF. As presented in Figure 12, the result
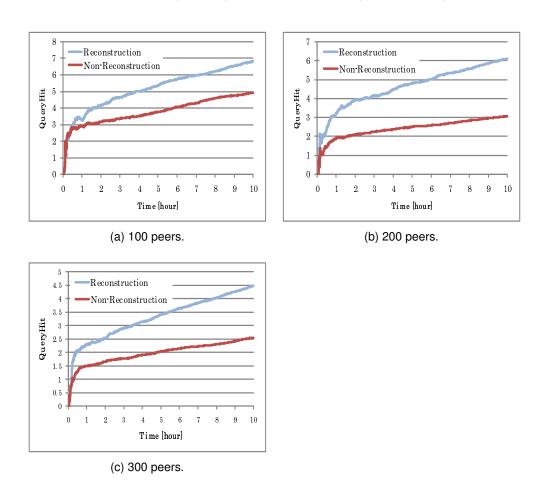

(a) 100 peers.

(b) 200 peers.

(c) 300 peers.

Figure 11: Averages of QueryHits.

Table 4: Simulation parameters for structured P2P.

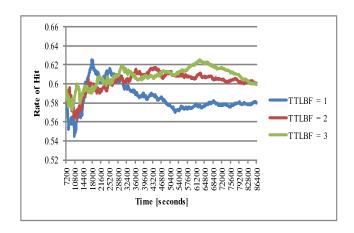| Parameter	Value
ID range	4-figure hexadecimals
Number of peers	100
Number of tags	1000
Number of counters in SBF	2^8

Figure 12: Hit ratios against TTL of BFs.

shows that the hit rate is not much affected by the TTL of BF as we had supposed. However, the hit rate itself is less than the expected 100%. This is possibly because the routing is insufficient by the limited range of BF propagation, and query messages are lost. We must investigate this phenomenon further.

7 Conclusion and Future Work

In this paper, we proposed a reorganization 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 TTL value is unchanged. In other words, we could make the TTL value smaller to achieve the same success ratios, which leads to lower network traffic.

We are still at the starting point regarding 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 in network traffic as well as to propagate the VTC messages farther than its neighbors.

Competing Interests

The authors declare that no competing interests exist.

References

[1] Lv Q, et al. Search and replication in unstructured peer-to-peer networks. In: Proc. ACM 16th Int. Conf. on Supercomputing. 2002;84-95.

[2] Holme P, Newman MEJ. Non-equilibrium phase transition in the coevolution of networks and opinions. Physical Review E, 74, 056108. 2006;5.

[3] Yamaguchi T, et al. Dynamic reorganization of P2P networks based on content similarity. In: Proc. 5th Int. Conf. on Emerging Network Intelligence. 2013;77-81.

[4] Lu J, Callan J. Content-based retrieval in hybrid peer-to-peer networks. In: Proc. ACM 12th Int. Conf. on Information and Knowledge Management. 2003;199-206.

[5] Wang J, Yang S. Content-based clustered P2P search model depending on set distance. In: Proc. 2006 IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology. 2006;471-476.

[6] Tang C, et al. Peer-to-Peer information retrieval using self-organizing semantic overlay networks. In: Proc. ACM SIGCOMM. 2003;175-186.

[7] Kacimi M, Yetongnon K. Content-based information routing and retrieval in cluster-based P2P overlay networks. In: Proc. 2008 IEEE Int. Conf. on Signal Image Technology and Internet Based Systems. 2008;70-77.

[8] Tirado JM, et al. Affinity P2P: a self-organizing content-based locality-aware collaborative peer-to-peer network. Computer Networks. 2010;54:2056-2070.

[9] Vazirgiannis M, et al. Peer-to-Peer clustering for semantic overlay network generation. In: Proc. 6th Int. Workshop on Pattern Recognition in Information Systems. 2006;10.

[10] Sripanidkulchai K, et al. Efficient content location using interest-based locality in peer-to-peer systems. In: Proc. IEEE INFOCOM. 2003;2166-2176.

[11] Voulgaris S, et al. Exploiting semantic proximity in peer-to-peer content searching. In: Proc. 10th IEEE Int. Workshop on Future Trends of Distributed Computing Systems. 2004;238-243.

[12] Raftopoulou P, Petrakis EGM. iCluster: a self-organizing overlay network for p2p information retrieval. Advances in Information Retrieval, LNCS. 2008;4956:65-76.

[13] Asvanund A, et al. Interest-Based Self-Organizing Peer-to-Peer Networks: A Club Economics Approach. 2004;39. Available: http://ssrn.com/abstract=585345

[14] Chen Y, et al. A scalable semantic indexing framework for peer-to-peer information retrieval. In: Proc. SIGIR 2005 Workshop on Heterogeneous and Distributed Information Retrieval. 2005;8.

[15] Martinovic I, et al. Self-protection in P2P networks: choosing the right neighbourhood. In: Proc. 1st Int. Workshop on Self-Organizing System. 2006;23-33.

[16] Yoshida Y, et al. Efficient decentralized evaluation of node trustworthiness in peer-to-peer networks. In: Proc. Int. Conf. on Computer Engineering and Technology. 2009;177-179.

[17] Mathes A. Folksonomies -cooperative classification and communication through shared metadata. Technical Report, University of Illinois Urbana-Champaign, LIS590CMC. 2004;13.

[18] Bloom BH. Space/Time trade-offs in hash coding with allowable errors. Comm. ACM. 1970;13(7):422-426.

[19] Fan L, et al. Summary cache: a scalable wide-area web cache sharing protocol. IEEE/ACM Trans. on Networking. 2000;8(3):281-293.

[20] Cohen S, Matias Y. Spectral bloom filters. In: Proc. ACM SIGMOD Conference. 2003;241-252.

[21] Zhao BY, et al. Tapestry: a resilient global-scale overlay for service deployment. IEEE J. On Selected Areas in Communications. 2004;22(1):41-53.

[22] Plaxton CG, et al. Accessing nearby copies of replicated objects in a distributed environment. In: Proc. 9th Annual ACM Symp. on Parallel Algorithms and Architectures. 1997;311-320.