Towards a Society of Peers: Expert and Interest Groups in Peer-to-peer Systems

Achmad Nizar Hidayanto[1] and Stéphane Bressan[2]

[1] University of Indonesia
[2] National University of Singapore
nizar@cs.ui.ac.id , steph@nus.edu.sg

Abstract. The social behavior of peers in peer-to-peer network can be inferred from the observable factors of the system and its components as it is created, lives and evolves. Following a social metaphor, it should be possible to use the observation of these behaviors to organize the network of peers for purposes as various as improving the retrieval performance, efficiently managing storage, improving robustness and increasing security, for instance. In order to concretely illustrate this idea and to precisely quantify its benefits in a concrete scenario, we consider the important example of the improvement of retrieval performance. We propose an unstructured peer-to-peer architecture in which the system, adaptively and in a decentralized manner, learns the expertise and interest of peers, and dynamically re-organizes itself by creating efficient communities (groups) of peers.

1. Introduction

The point of view that we are proposing in this paper is one of a social network view of peer-to-peer systems. The social behavior of peers in peer-to-peer network can be defined by the observable factors of the system and its components as it is created, lives and evolves. Following a social metaphor, it should be possible to use the observation of these behaviors to organize the network of peers for purposes as various as improving the retrieval performance, efficiently managing storage, improving robustness and increasing security, for instance. The building blocks of this social organization are the connection of two peers and the grouping of peers. Notice that, a priori, we do not know whether the peer and the peer-to-peer architecture reflect the logic and structure of the natural social network of their users. In other words we cannot rely on a relationship between peers' knowledge and interest and the ones of their users, but only on the emergent society of peers.

In order to concretely illustrate the idea of leveraging social behaviors of peers and to precisely quantify its benefits in a concrete scenario, we consider the important example of the improvement of retrieval performance. Through its observable behavior, a peer displays traits that can be assimilated with those of an individual. A peer appears to be specialist in some domains; the domains are induced by the topics of the documents, which are stored and served by the peer. A peer appears to be interested in some (possibly different) domains; the domains are induced by the topics of the queries, which are issued by the user or users of the peer. We propose an unstructured peer-to-peer architecture in which the system, adaptively and in a decentralized manner, learns the expertise of peers, and dynamically re-organizes itself by creating efficient communities (groups) of peers.

In the next section we present a rapid overview of the background and related works. In section 3, we present our proposed architecture. We show how a routing-index-based unstructured peer-to-peer network can be constructed to evolve and adapt to observable expertise and interests of peers. In section 4 we comparatively and empirically evaluate the performance of the proposed solution. Finally we summarize our contribution and outline future work in section 5.

2. Backgrounds and Related Works

Searching in unstructured peer-to-peer system relies on either broadcasting queries to random nodes or selecting some neighbors that have high probability to answer the queries with the help of routing index [9]. Our previous works in [10, 11] showed the use of reinforcement algorithm to update information in routing index. Such approaches are proven in reducing the routing time of resource searching. Thus the scalability of the system is maintained in [12] by managing the reasonable size of the routing index. However, we believe that we still can improve the performance of the peer-to-peer system by creating overlay on top of existing unstructured topology using concept of grouping. The ideas of grouping and reconnecting peers in peer-to-peer networks have been investigated in various forms and for various purposes.

Merugu et al. [1] proposed an overlay network with two types of link: short-links and long-links. Each node in the system selects at random some close nodes to be its short-link neighbors and some distant nodes to be its long-link neighbors. They concluded that having many close neighbors and few random neighbors, will increase the chances of locating files and the nodes where these files are found are, on average, close to the query source. However this is under uniform distributions hypothesis for queries and documents.

Ramaswamy et al. [2] proposed a clustering algorithm based on the connectivity of peers, called Connectivity-based Distributed Node Clustering (CDC). It is a distributed approximation of graph clustering algorithms such as Star clustering [8]. In order to improve the quality of clustering, they proposed Two Hops Return Probability (THP) as a mechanism to select initiator node. The simulation result showed that CDC with THP approaches the accuracy of a centralized algorithm.

Vazirgiannis et al. [3] proposed another algorithm. Instead of measuring the connectivity between peers to cluster peer as in [2], they used content similarity. The algorithm of clustering is divided into four steps: local clustering, initiator selection, zone creation, and intra zone clustering. The clustering process starts from the local peer by clustering the documents using standard clustering algorithm such as K-Means until it is determined globally by collecting feature vector from the peers.

Other works that are considered important are in [4, 5, 6, 7]. They proposed the concept of grouping, which is close to the concept of clustering. A peer is made closer to other peers that have high probability to answer the query. Each peer stores a profile that consists of set of queries and the peers that have answer to those queries. [6] is the simplest one as it only keeps one shortcut for one topic. When there is no shortcut in the profile, it will broadcast the query through random nodes. In [4], each query is associated with one link that most recently answers the query. Incoming queries in a peer will be forwarded to the peer, which are stored in the profile, with highest aggregate similarity to the query. The profile size is reduced by applying Least Recently Used (LRU) algorithm. However, their approach allows storing similar queries in the profile thus requiring more profile size to keep various queries. In [5], a peer will keep n peers that have strong relationship. However, they do not have mechanism to keep the profile size in manageable size.

The architectures that we investigate in this paper combine content similarity, for the expert groups, query similarity, for the interest groups, and clustering. We propose learning mechanisms for peer organization that help peers finding answers efficiently and effectively. The mechanism can adapt to the network changes by continuously examining the content/interest of the peers.

3. Grouping in Peer-to-peer Networks

3.1 Information Retrieval in Unstructured Peer-to-peer Network with Routing Indices

We consider a peer-to-peer architecture for the keyword-based retrieval of text documents. Peers store collection of documents. Queries are in form of list of terms or keywords and the retrieval of documents uses a similarity measure like the cosine similarity. The network of peers is unstructured. Each peer maintains a routing index for the routing of queries. In our case, the peers in our routing index act as link to create overlay on top of existing random network. A routing index [9] is a data structure that records each topic in a selected set of terms or keywords, a list of potentially relevant neighbors. The maximum number of topics and neighbors are parameters of the system that can be set explicitly or managed indirectly with an appropriate replacement strategy

such as Least Recently Used (LRU), Least Frequently Used (LFU), or based on the popularity of the topics.

In addition to the neighbors in the routing index, each peer maintains a list of random neighbors. These are peers (randomly selected) of those with which the peer was acquainted as it joined the network or received various messages. Let us say that a peer knows a total of N neighbors, S neighbors are in the routing index and R = N-S neighbors are randomly chosen (as peers join the network and interact with other peers). R and S are parameters of the system. The peers and their neighbors form a small word graph in the sense of [1].

A query has the form $q(\tau, \lambda, \eta, \theta)$ where τ is a (possibly weighted) list of keywords. It forms a vector that we call a topic. λ is the time-to-live (TTL), i.e. global expiry time or maximum number of hops allowed for the query (this is to avoid ghost queries to indefinitely circulate in the network). η is the maximum number of documents requested. θ is the minimum similarity of the documents to the query and only those documents with similarity above θ are considered as the answers. A peer receiving the query first compares its own documents with the query. It computes the similarity of the weighted vector of keywords or terms representing the document in the vector space model, with query τ. If the similarity is above the threshold θ, the document is returned to the requestor as an answer of the query and η is decremented accordingly. If it cannot provide η results and the TTL is not yet reached (i.e. λ is not equal to 0 if λ is number of hops), the query will be forwarded to some neighbors (and λ is decremented if λ is number of hops).

The query is forwarded to a maximum of n neighbors ($n \leq N$). However, we have choices to forward queries to random neighbors only, to neighbors in the routing index only, or by combining them. Thus, the query can be forwarded to s neighbors which are selected from the routing index and r=n-s neighbors which are selected from the list of random neighbors. s and n are parameters of the system. If s=0 the peer does not use the routing index and it will behave as n-broadcast à la Gnutella (see [16,17]). If $s > 0$, the peer will compute the similarity between the query and the t topics in the routing index. The query is forwarded to those up-to s neighbors in the lists of neighbors of topics whose similarity with the query is maximum and higher than a threshold σ, which is a parameter of the system.

We have shown in [10,11,12] that adaptive routing indices using reinforcement learning can efficiently and effectively route queries in unstructured peer-to-peer networks. We explore in this paper three semantically motivated approaches to the creation and maintenance of routing indices. The different algorithms that we propose below differ in the way the routing index is created and maintained. We present approaches based on the expertise of peers (i.e. the topics representative of the documents which are stored by a peer), the interest of peers (i.e. the topics representative of the queries which are issued at the peer or received and forwarded), and combination between both of them.

3.2 Expert Networks

In this architecture, peers are grouped according to topics representative of their content or expertise, i.e. of the documents that a peer stores, thus creating expert groups/networks.

As in the architecture described in [3], peers in our expert group maintain a set of feature vectors that are centroids of the clusters of the documents, which are stored by the peers. These vectors can be obtained and maintained by online versions of vector or graph clustering algorithms (see [8] for an example of an efficient and effective online clustering algorithm). These vectors represent the abstract topics, in which the peer is an expert, i.e. about which it can answer queries. The vectors or topics are the entries of the routing index. Notice that the routing index is initially empty and evolves as documents are added or removed from the peer.

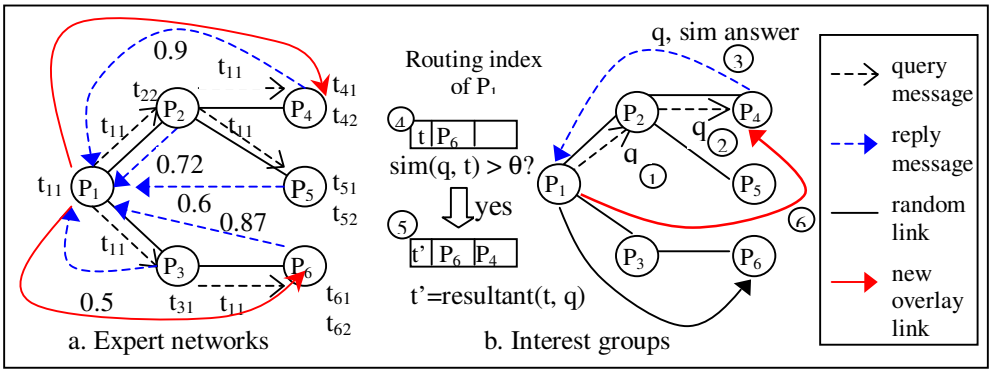

Fig. 1. Illustration of links establishment

When a peer joins the network, it chooses random peers to be linked. Hereafter, the peer needs to advertise its expertise to get knowledge of other peers that share similar expertise by broadcasting all its feature vectors. The depth of broadcasting is controlled by TTL parameter. Figure 1.a illustrates the process. Suppose we set the TTL parameter for advertising to 2 and P_1 that has topic t_{11} joining the network. P_1 will broadcast topic t_{11} to P_2 and P_3, and TTL is decremented. P_2 and P_3 will evaluate t_{11} against their own topics and send the similarity of the most similar topic to P_1, for instance 0.72 and 0.5 respectively. As the TTL is not yet reached, P_2 and P_3 will broadcast t_{11} to P_4, P_5 and P_6 and decrement the TTL accordingly. P_4, P_5 and P_6 will evaluate against their own topics and send the similarity of the most similar topic to P_1, for instance 0.9, 0.6, and 0.87 respectively. As the TTL has reached 0, the broadcasting is stopped. P_1 has received proposal of links with similarity 0.72, 0.5, 0.9, 0.6, 0.87. If we set the value of parameter S to 2, P_1 will establish links to P_4 and P_6 as they are the top two of the most similar peers it can reach within 2 hops.

However, in order to adapt to the evolution of the network, broadcasting continues. In this way, the system learns the network status and adapts to the new changes. We

proposed four strategies for a peer to advertise its expertise as it communicates with other peers. We refer to the strategies as requestor, return path, backward propagation, and dual propagation.

In the requestor strategy, a peer answering a query advertises its expertise to the peer that issues the query. In the return path strategy, a peer answering a query advertises its expertise to all peers on the return path to the peer that issues the query. In the backward propagation strategy, a peer receiving a query (but not necessarily answering it) advertises its expertise to the peer that sends the query. In the dual strategy, in addition to a backward propagation, the peer also advertises its expertise to the peers it contacts (a forward propagation). Upon sending query to the neighbor, the peer also advertises its expertise corresponding to the query and asks the neighbor to update its routing index (if applicable). The neighbor will reply and advertise back its expertise to the peer.

When a peer receives advertising message about another peer's expertise, it compares it with its own by computing the similarity between the vector of expertise and the vectors of topics in the routing index. If the similarity is above a threshold ρ, which is a parameter or the system, the peer sending the expertise is added as a neighbor corresponding to the topic in the routing index if size constraints for the routing index allow. It becomes an expert neighbor. Each topic in the routing index is associated to an explicit maximum of γ neighbors. If the peer already has γ neighbors for the topic concerned, it replaces one of expert neighbors with smallest similarity to the topic with the new candidate neighbor, provided that the similarity of the new candidate of expert neighbor to the topic is higher; otherwise, it ignores the new candidate.

If the similarity is above the threshold, conversely the receiving peer advertises its topics to the sending peer, which, in turn, considers it for inclusion in the list of expert neighbors in the routing index. Since the routing index is dynamically modified, the topology of the network of peers and their groups that it defines evolving. Peers are dynamically grouped according to their expertise.

3.3 Interest Groups

We also design an architecture in which peers are grouped according to topics representative of their interest, i.e. of the queries, which are issued or forwarded by the peer, thus creating interest groups. In this architecture, that we call interest groups or networks, the evolution of the network is different than in the previous scenario, as the interest of peers, i.e. the queries that it issues or forwards, can only be observed over time.

The routing index contains direct links to other peers, which have answer for particular topic. It is initially empty and evolves as the queries elapsed/received/forwarded to other peers. Figure 1.b shows the mechanism. Upon receiving a query, a peer will search in its own database first. If it can satisfy the query, it will send the result to the requestor otherwise it will broadcast the query to the peers that have the answer (if they exist in the routing index) or broadcast to random nodes.

When a peer receives answer from other peers, it compares it with its own by computing the similarity of the vector of query with the vector of topics in the routing index. If the similarity is above a threshold ρ, which is a parameter of the system, the answering peer is added as a neighbor corresponding to the topic in the routing index if size constraints for the routing index allow. As the vector of query and the corresponding vector of topic in the routing index are considered similar, we can update the vector of topic in the routing index by taking the resultant between both of them. Such approach will reduce number of vectors that should be stored in the routing index thus maintaining the scalability of the system.

Each topic in the index is associated to an explicit maximum of γ neighbors. If the peer already has γ neighbors for the topic concerned, it replaces one of neighbors with smallest similarity to the topic with the new candidate neighbor, provided that the similarity of the candidate neighbor to the topic is higher; otherwise, it ignores the new candidate neighbor. The same mechanism can be applied for links exchange upon sending/receiving queries. We also can apply the same mechanism for updating routing index as stated previously in sub section 3.2. The system will evolve and dynamically change its topology and follow the behavior of user interest. Thus peers are dynamically grouped according to the user interest.

3.4 Hybrid Groups

Finally we have designed a hybrid architecture combining neighbors obtained from both the expert group and the interest group. More entries are put in the routing index that consists of two parts; topics representing the peer's expertise and their corresponding links and topics representing the peer's interest and their corresponding links. This latter design has anthropomorphic value: individual seeking for knowledge will navigate networks of acquaintances characterized by their expertise and interest.

4. Experiment Results

4.1 Experimental Set-up

We simulated the three proposed approaches using the Peersim simulator [13]. Peersim is a discrete event simulator for peer-to-peer networks, which is used to evaluate the performance of several approaches, including those of [15,16]. The experiments were conducted using 4000 nodes on WireK topology with $k = 4$ [18]. In order to simulate documents and queries with their topics, we created 500 root topics coming from 1000 keywords. Each root topic consists of 5 keywords and the weight of each keyword is assigned by generating random values from 1 to 100. For each peer we generated its

topics by randomly modifying the root topics: we create a new topic from a root topic by subtracting/adding a random value between 0 and 30 from each weight component of the vector. The weight below 0 will be set to 0 and the weight above 100 will be set to 100. We generated 1 to 3 topics per peer.

Documents in a peer were generated by randomly modifying the peer's topics. We generated 1 to 10 documents per topic per peer. Queries were generated by randomly modifying root topics. For each topic, the number of queries generated was drawn from a Zipfian with a factor of 0.5. We issued 25 queries at random peers at each unit of time. Peers issue queries related to documents in their collection (interest meets expertise) as well as unrelated queries. We also varied the ratio X/Y of related to unrelated queries for each peer. A related query is a query similar to a topic of the peer issuing it. We requested 10 documents for each query.

In our experiments, two topics are considered different if their similarity below 0.8 (θ, σ and ρ). We say that a query is answered if a result at least is returned within its TTL λ.

4.2 Expert Network

We first evaluate the performance of expert networks. Figure 2 to 4 show the performance for varying r, number of random neighbors, and s, number of expert neighbors in the routing index, to which a query is forwarded. Figure 2 shows that increasing number of expert neighbors will decrease the number of messages, as search is more focused. But in contrast, fewer queries are answered with more expert neighbors as shown in Figure 3. The explanation of this phenomenon is peers may issue queries beyond the expertise of the groups. As peers within the groups have similar expertise, fewer peers can answer such queries. However the average number of hops for answered queries is better with more expert neighbors as shown in Figure 4.

We now vary the proportion X/Y of related queries to unrelated queries. As shown in Figure 5 and 6, issuing more related queries will improve the performance in terms of both traffic and number of hops as many peers within the groups can answer the queries quickly.

We now study the impact of γ the maximum number of expert neighbors stored for each topic in the routing index on the performance. Figure 7 shows that increasing the number of links from 2 to 4 improves efficiency. However, the improvement is less and less significant as the number of expert neighbors increases. When we increase number of links for particular topic, part of them may less relevant to the topic.

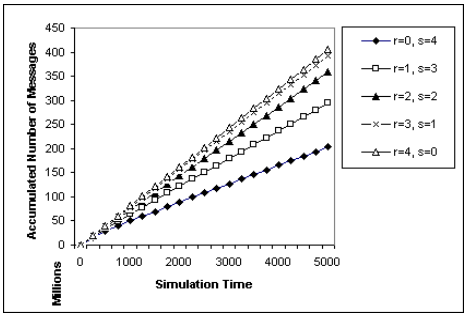

Fig. 2. Number of messages generated for varying number of random links

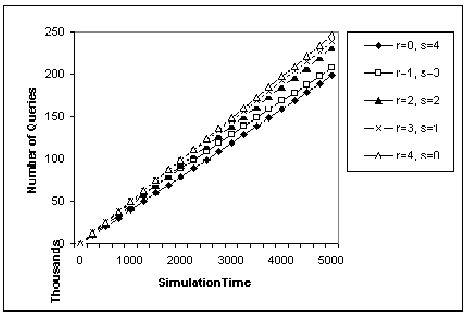

Fig. 3. Number of answered queries for varying number of random links

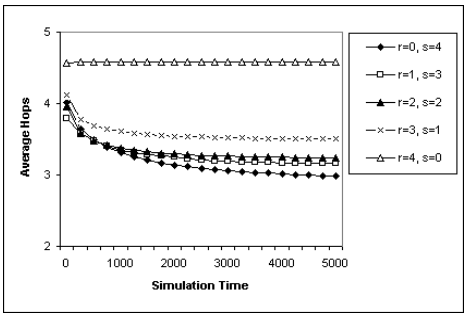

Fig. 4. Average number of hops of first answer for varying number of random links

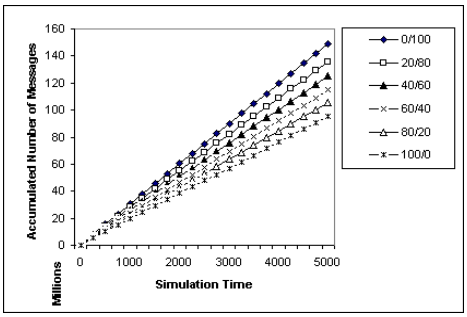

Fig. 5. Number of messages generated for varying X/Y

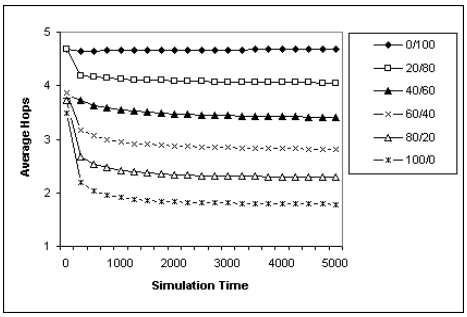

Fig. 6. Average number of hops for varying X/Y

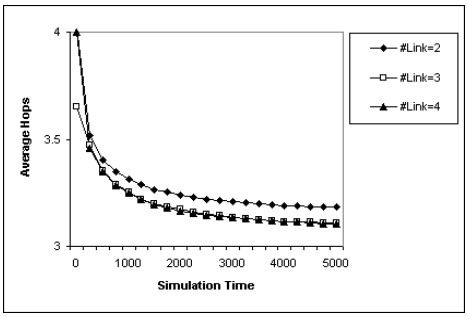

Fig. 7. Average number of hops of first answer on various number of links

We also measure the influence of the TTL of initial advertising messages (when a peer is joining) on the performance of the system. We see that the increase cost of a longer

TTL in terms of traffic is neglectable compared to the gain in retrieval efficiency, as illustrated on Figure 8.

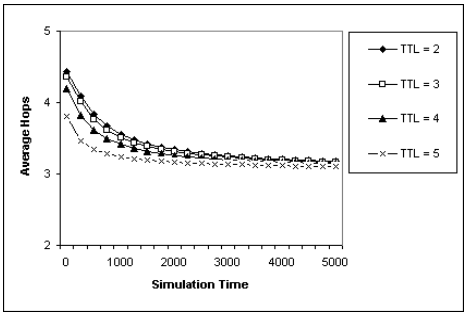

Fig. 8. Average number of hops for the first answer for varying TTL for advertising

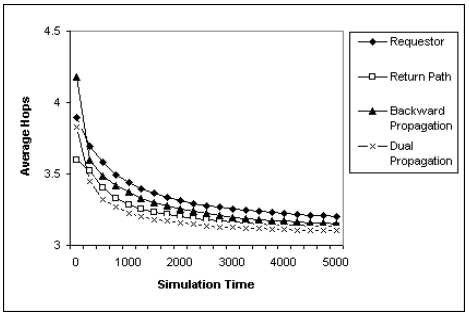

Fig. 9. Average number of hops for the first answer for various update strategies

We proposed 4 strategies to dynamically update the routing index that we refer to as requestor, return path, backward propagation and dual propagation. Figure 9 illustrates the performance of the various strategies. Dual propagation requires the smallest number of hops. The more we exchange the expertise, the more the routing index will reflect the correct state of the network thus offering better performance.

4.3 Other Experiments

We have conducted several experiments to evaluate the performance of the interest groups and hybrid groups architectures. For the sake of concision and simplicity we do not report these results extensively in this paper. We do not report these results here but confirm that they show the expected trends. In particular a good combination of expert and interest groups can yield retrieval in few hops for first and complete answers without too much traffic for queries and advertisement.

5. Conclusion

We propose a social network metaphor for the design of unstructured peer-to-peer architectures. We present and evaluate an instance of this general idea for the improvement of retrieval performance. In this instance we observe the expertise of peers that is defined by the topics of the documents that they store. We then cluster peers in the network accordingly. We outline the details of the solution and its variants. We empirically show the performance improvement by simulating such an architecture under various realistic conditions and by comparatively measuring its performance.

Our future work includes the declination of the metaphor. We would like to start by elaborating a more complex instance for the case of expertise and interest in which clusters are hierarchal thus simulating human organizations such as Academia.

References

1. Merugu, S., Srinivasan, S., Zegura, E.: Adding Structure to Unstructured Peer-to-Peer Networks: the Use of Small-World Graphs. J. Parallel and Distributed Computing, 65(2), 142--153 (2005)
2. Ramaswamy, L., Gedik, B., Liu, L.: Connectivity Based Node Clustering in Decentralized Peer-to-Peer Networks. J. IEEE Transaction on Parallel and Distributed Systems, 16(9), (2005)
3. Vazirgiannis, M., Nørv°ag2, K., Doulkeridis, C.: Peer-to-Peer Clustering for Semantic Overlay Network Generation. In: 6th International Workshop on Pattern Recognition in Information Systems, (2006)
4. Kalogeraki, V., Gunopulos, D., ZeinalipourYazti, D.: A Local Search Mechanism for P2P Network. In: 11th International Conference on Information and Knowledge Management (2002)
5. Upadrashta, Y., Vassileva, J., Grassmann, W.: Social Networks in Peer-to-Peer Systems. In: 38th Hawaii International Conference on System Science (2005)
6. Sripanidkulchai, K., Maggs, B., Zhang, H.: Efficient Content Location Using Interest-Based Locality in Peer-to-Peer System. In: 22nd IEEE Infocom (2003)
7. Ng, C. H., Sia, K. C.: Peer Clustering and Firework Query Model. In: 11th International World Wide Web Conference (2002)
8. Aslam, J., Pelekhov, K., Rus, D.: The Star Clustering Algorithm. Journal of Graph Algorithms and Applications, 8(1), 95--129 (2004)
9. Crespo, A., Garcia-Molina, H.: Routing Indices For Peer-to-Peer Systems. In: International Conference on Distributed Computing Systems (2002)
10. Liau, C.Y., Hidayanto, A.N., Bressan, S.: Adaptive Peer-to-peer Routing with Proximity. In: 14th International Conference on Database and Expert Systems Applications (2003)
11. Bressan, S., Hidayanto, A.N., Liau, C.Y., Hasibuan, Z.A.: Adaptive Double Routing Indices: Combining Effectiveness and Efficiency in P2P Systems. In: 15th International Conference on Database and Expert Systems Applications (2004)
12. Bressan, S., Hidayanto, A.N., Hasibuan, Z.A.: Exploiting Local Popularity to Prune Routing Indices in Peer-to-Peer Systems. In: International DEXA Workshop (2005)
13. PeerSim P2P Simulator, http://peersim.sourceforge.net/
14. Jelasity, M., Montresor, A.: Epidemic-Style Proactive Aggregation in Large Overlay Networks. In: 24th International Conference on Distributed Computing Systems, pp. 102--109 (2004)
15. Montresor, A.: A Robust Protocol for Building Superpeer Overlay Topologies. In: 4th International Conference on Peer-to-Peer Computing (2004)
16. Jovanovic, M., Annexstein, F., Berman, K.: Scalability Issues in Large Peer-to-Peer Networks--A Case Study of Gnutella. Technical Report, University of Cincinnati (2001)
17. Yang, B., Garcia-Molina, H.: Efficient Search in Peer-to-Peer Networks. In: 22nd International Conference on Distributed Computing Systems (2002)
18. Oram, A.: Peer-to-Peer: Harnessing the Power of Disruptive Technologies. O'Reilly & Associates (2001)