Managing Peer-to-Peer Networks with Human Tactics in Social Interactions

Lu Liu[1], **Nick Antonopoulos**[2], **Stephen Mackin**[3]

Abstract — Small-world phenomena have been observed in existing peer-to-peer (P2P) networks which has proved useful in the design of P2P file-sharing systems. Most studies of constructing small world behaviours on P2P are based on the concept of clustering peer nodes into groups, communities, or clusters. However, managing additional multilayer topology increases maintenance overhead, especially in highly dynamic environments. In this paper, we present Social-like P2P systems (Social-P2Ps) for object discovery by self-managing P2P topology with human tactics in social networks. In Social-P2Ps, queries are routed intelligently even with limited cached knowledge and node connections. Unlike community-based P2P file-sharing systems, we do not intend to create and maintain peer groups or communities consciously. In contrast, each node connects to other peer nodes with the same interests spontaneously by the result of daily searches.

Keywords — peer-to-peer, social interactions, small world, performance evaluations, simulations

---

[1] Surrey Space Centre, University of Surrey, Guildford, Surrey GU2 7XH, United Kingdom. Email: l.liu@surrey.ac.uk
School of Computing, University of Leeds, Leeds, West Yorkshire LS2 9JT, United Kingdom. Email: luliu@comp.leeds.ac.uk
[2] Department of Computing, University of Surrey, Guildford, Surrey GU2 7XH, United Kingdom. Email: n.antonopoulos@surrey.ac.uk
[3] Surrey Satellite Technology Limited, Surrey Research Park, Guildford, Surrey GU2, 7XH, United Kingdom. Email: s.mackin@sstl.co.uk

1. INTRODUCTION

For resource discovery in social networks, people can directly contact some acquaintances that potentially have knowledge about the resources they are looking for. However, in current peer-to-peer (P2P) networks, peer nodes lack capabilities similar to social networks, making it difficult to route queries efficiently. Similar to social networks where people are connected by their social relationships, two autonomous peer nodes can be connected if users in those nodes are interested in each other's data. The similarity between P2P networks and social networks, where peer nodes are people and connections are relationships, leads us to believe that human tactics in social networks are useful for improving the performance of object discovery [1] by self-managing autonomous peers on unstructured P2P networks.

Existing solutions for object discovery in the P2P systems can be classified into two categories: structured and unstructured P2P systems. Structured P2P systems (e.g. Chord [2], CAN [3], and Pastry [4]) have dedicated network structure on the overlay network. Distributed Hash Tables (DHTs) have become the dominant methodology for object discovery in structured P2P networks [5]. However, some recent studies (e.g. [6], [7]) argued that most DHTs can not handle the cost of maintaining a consistent distributed index in the dynamic and unpredictable Internet environments. Some structured P2P protocols (e.g. Kademlia [8]) are beginning to seek ways to save the cost of maintaining a consistent index. In contrast, unstructured P2P systems do not control data placement and are more resilient in dynamic environments, but current search techniques in unstructured P2P systems tend to either require large storage overhead or generate massive network traffic. Additionally, node connections of some unstructured P2P systems are formed randomly that is less efficient in contrast to the human communities which are formed by the social interactions between people having common interests.

The small world phenomenon, proposed by Stanley Milgram, is the hypothesis that everyone in the world can be reached through a short chain of social acquaintances [9]. Duncan Watts proposed a mathematical model [11] to analyze the small world phenomenon with highly clustered sub-networks consisting of local nodes and random long-range shortcuts that help produce short paths to remote nodes. The theory of small world in social sciences has also been applied to the system design of P2P overlay networks. But most studies of constructing small world behaviours on P2P networks are based on the concept by clustering peer nodes into groups, communities, or clusters (e.g. [12], [13], [14], [15]). Studies like [16], [17], [18] have explored the possibility of building an information sharing system by clustering peer nodes into "groups" or "communities" based on their interests. However, the simple community formation and discovery becomes much more complex due to the lack of a central server. A large communication overhead is required to compensate for the server even when operating with information dissemination techniques (e.g. Gossiping and Rumour Spreading [19]) and compact data structures (e.g. Bloom Filters [20]).

In this paper, we present Social-like P2P systems (Social-P2P and Active Social-P2P) for object discovery by mimicking human behaviours in social networks. Different from most informed search algorithms (e.g. local indices [21]), peer nodes learn knowledge from the results of previous searches and no overhead is required for Social-P2P to obtain additional information from neighbouring nodes on the P2P overlay. Unlike community-based P2P file-sharing systems (e.g. [12], [13], [14]), we do not intend to create and maintain peer groups or peer communities consciously. In contrast, each node connects to other peer nodes with the same interests gradually by the results of daily searches. Finally, peer nodes with the same interests will be

highly connected to each other spontaneously. Social-P2Ps can be deployed on the top of existing unstructured P2P networks (e.g. Gnutella) to improve performance of object discovery.

2. RELATED WORK

2.1. Resource Discovery in P2P Networks

Although current search methods in unstructured P2P systems are heterogeneous and incompatible, most of them are dedicated to solving the observed issues of blind flooding mechanisms and generally can be classified into the following approaches according to their design principles. The first approach enables peer nodes to create query routing tables by hashing file keywords and regularly exchanging those with their neighbours (e.g. [21]). Peer nodes normally maintain additional indices of files offered by overlay neighbours or neighbours' neighbours within a specific distance. A peer node can decide which peer nodes to forward a query to by using this additional information. The second approach is based on hierarchical architecture which reorganises peer nodes into a two-layer hierarchy with super-peer nodes (e.g. [22], [23]). Super-peer nodes are capable and reliable peer nodes that take more responsibility for providing services in P2P networks. In many P2P applications, topology determines performance. The third approach improves network performance by adapting and optimizing the overlay topology (e.g. [24], [25]).

The fourth approach is closely related to the algorithms we are presenting in this paper. The fourth approach utilizes the historic record of previous searches to help peer nodes make routing decisions, such as Adaptive Probabilistic Search (APS) [26] and NeuroGrid [27]. Different from topology optimization methods (e.g. [24], [25]), the search algorithms of APS are not allowed to alter the overlay topology. In APS, each node keeps an index describing which files were requested by each neighbour. The probability of choosing a neighbour to find a particular file

depends on previous search results. This "file-oriented" approach leads to situations where popular files could be located very fast, while it is difficult to locate other less popular files. NeuroGrid utilises the historic record of previous searches to help peer nodes make routing decisions. In the NeuroGrid network, peer nodes support distributed searches through semantic routing by maintaining routing tables at each node [27]. In the local routing tables, each peer node is associated with keywords regarding the content it stores. When a peer node forwards a query, it will search for the peer nodes that are associated with the query keywords. However, NeuroGrid is only effective for previous queried keywords and is not suitable for networks where peer nodes come and go rapidly [28].

In addition, "small world" social phenomenon has been observed in current P2P networks [10]. Maintaining and searching "small world" has been discussed in recent studies. Kleinberg [13] discussed the issue of decentralized P2P search with partial information about the underlying structure. Small world architecture for P2P networks has been proposed in the previous work [14] with a semi-structured P2P algorithm in multi-group P2P systems, which has the advantages of both structured and unstructured P2P approaches. A study in [15] proposed an enhanced clustering cache replacement scheme for Freenet by forcing the routing tables to resemble neighbour relationships in a small-world acquaintance graph.

2.2. Social P2P Networks

TSN [29] is a social P2P infrastructure, which aims to give computers a rudimentary social network. TSN allows applications to work in more humanly natural way, seamlessly integrating centralised services and distributed contacts. TSN is designed to be configurable and dynamic. Applications can specify their own both structures and matching policies for the meta-data. TSN provides a general infrastructure for a social peer-to-peer network. However, the search

mechanism of TSN is very simple, which does not provide any matching policies and node selection algorithms for application development.

Tribler [30] is a social-based P2P file sharing paradigm built on the top of BitTorrent. Tribler exploits social phenomena by maintaining social networks and using these in content discovery and download. Tribler uses an epidemic protocol named Buddycast to discover buddies with similar tastes. By using Buddycast, each peer node maintains a list of the top-*N* most similar peers along with their current preference lists. Periodically, each peer node connects to either one of its buddies to exchange preference lists, or to a randomly chosen peer node, to exchange this information. However, Tribler focuses on cooperative downloading rather than resource discovery in P2P networks. The periodical exchange of preference lists introduces a potentially large amount of communication overhead as well as new security and privacy issues into the system.

3. ALGORITHM DESCRIPTION

3.1. Social-P2P

In this section, we will describe the algorithms of Social-P2P by analogizing from the human strategies in social networks.

3.1.1 Knowledge Index Formation

In social networks, people remember and update potentially useful knowledge from social interactions. As similar to social networks, each Social-P2P node builds a knowledge index that stores associations between topics and associated nodes by the results of searches.

When a peer node receives a query, it will first search the local content index to find matched files. If the query need to be further forwarded, it will use the local knowledge index to find associated peer nodes and multicast the query to these peer nodes as shown in Figure 1. If a

search is successful, the requesting node updates its knowledge index to associate the peer nodes that have responded data successfully with the requested topic and the response time in the following form: {query topic, responding node's address, last response time}. In the meantime, the requesting node also removes invalid cached knowledge according to the results of searches.

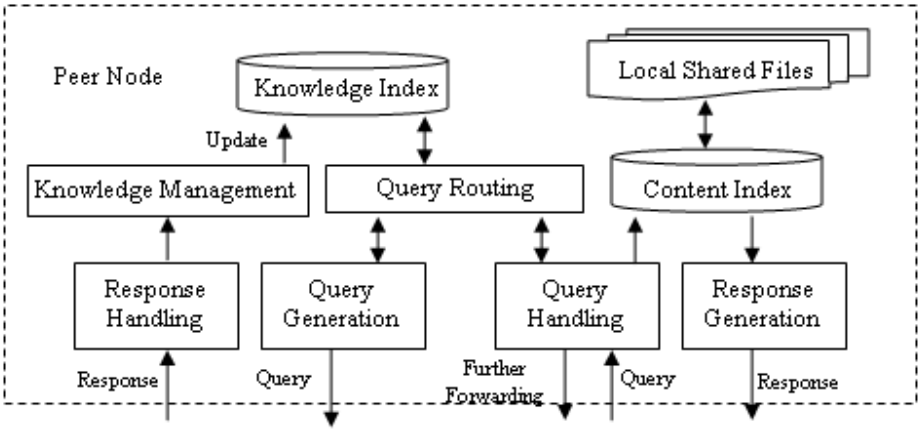


Figure 1. Key components of peer nodes.

Therefore, peer nodes can learn from the results of previous searches, which makes future searches more focused. When more searches have been done, more knowledge can be collected from search results. If this process continues, each node can cache a great deal of knowledge that is useful to quickly find the peer nodes with the required data in the future.

3.1.2 Query Handling

For resource discovery in social networks, people usually recall knowledge in memory to find the right people to contact. The persons recalled from memory may directly relate to their requests. For example, Bob wants to borrow an Oxford English Dictionary and remembers that he once borrowed it from his friend Alice. Therefore, he can directly contact Alice again for the dictionary. However, in most circumstances, people cannot find the persons who are directly related to their requests. For example, Bob may never have borrowed or he can not clearly remember whether he has ever borrowed an Oxford English Dictionary. But he believes his

friend Alice, who is a linguist, probably has the dictionary or at least has more knowledge about who has this dictionary. In this case, Oxford English Dictionary is in the area of linguistics and Bob find Alice has abundant knowledge on linguistics from previous intercommunications. Alice probably has not the dictionary, but she will use her own knowledge to help Bob find the dictionary with a high likelihood.

Analogous to social networks, Social-P2P utilises a logic-based semantic approach to route queries to a selected subset of neighbouring nodes on the P2P overlay in each hop. In addition, Social-P2P also involves a dedicated strategy to address the network overload problem of existing P2P systems (e.g. Gnutella). In order to conduct a more efficient search, the number of peer nodes to be forwarded is adjustable according to the correlation degree of the selected node to the query between a minimal number $FN_{\min}$ and a maximum number $FN_{\max}$ in each hop. Social-P2P uses a similar method to Gnutella to prevent infinite propagation: *Time to Live* (*TTL*). *TTL* represents the number of times a message can be forwarded before it is discarded.

3.1.3 Routing Algorithm

The routing algorithm of Social-P2P involves the following three phases. When a node receives a query which needs to be forwarded, the node routing algorithm firstly searches for the peer nodes directly associated with the requested topic from the local knowledge index and ranks them with the last response time in the corresponding entry. The peer node that is input or updated more recently gets a higher rank. These directly associated peer nodes have the greatest likelihood of finding the requested files. Hence, at most $FN_{\max}$ peer nodes will be forwarded.

However, the success probability of finding $FN_{\max}$ directly associated nodes in the first phase is very low, especially for new peer nodes with little knowledge. If there are not enough directly associated nodes found in the first phase, the algorithm will move to the second phase

that searches for the peer nodes sharing content associated with the interest area of the requested topic from the local knowledge index. An interest area in Social-P2P is a semantic area with a set of topics. The corresponding interest area of a specific topic and the other topics in this interest area can be found from the Open Directory Categories [31], which is the most widely distributed data base of Web content with a hierarchical topic structure. Social-P2P users use the common topic hierarchy of the Open Directory to generate a query. When a user generates a query to search files about the topic "Gnutella", the query will be constructed as "Computer: Software: Internet: Client: File Sharing: Gnutella". The closest parent directory "File Sharing" is the interest area of the topic "Gnutella". The other topics in the same area (BitTorrent, Gnutella, FastTrack, Napster, Freenet, Overnet and eDonkey) will be used in the second phase of node selection. Users can also define a category for their own query, if Open Directory cannot provide any satisfactory category for the query.

These peer nodes having content associated with the other topics in the same interest area of the requested topic will be sorted according to the degree of correlation to the interest area of the requested topic. The routing algorithm prefers to select the peer nodes with higher degrees of correlation rather than the peer nodes with lower degrees of correlation. If two or more nodes have the same correlation degree, we put the peer node that responded most recently first.

Searching for a piece of information in social networks is most likely a matter of searching the social network for an expert on the topic together with a chain of personal referrals from the searcher to the expert [32]. If a peer node has a large amount of content in a particular area like an "expert", it is very likely that it will also have other content and knowledge in this area. In our simulations, the correlation degree of a selected node in a particular area is generated by how many topics in the area the peer node is associated with: $c = {n_{matches}}/{n_{total}}$, where $n_{matches}$ is the

number of topics in this area that the peer node is associated with and n_{total} is the total number of topics in this area.

The ranking of peer node *r* respective to the query is determined by the correlation degree of the peer node to the interest area of the requested topic with the equation:

$$r = round\left(c \cdot \left(FN_{\max} - FN_{\min}\right)\right) + FN_{\min}, \qquad (1)$$

where the function *round(x)* returns the closest integer to the given value *x*. When the correlation degree of a peer node is very low ($c \approx 0$), there is a low likelihood to find the requested files from the peer node. Therefore, the probability of querying the node should be low with a small ranking ($r \approx FN_{\min}$). In contrast, when the selected node is highly correlated with the area of the requested topic by matching most topics in this area ($c \approx 1$), the ranking of the node $r \approx FN_{\max}$.

The flowchart in Figure 2 shows the second-phase of the node routing algorithm used in our simulations where *n* is the number of peer nodes that have been selected in the first and second phase. As shown in this flowchart, the peer nodes associated with the interest area of the requested topic are ranked with their correlation degrees. Because peer nodes are taken from the list in order, the rankings of these peer nodes *r* will decrease with the reducing correlation degrees *c* (according to Equation 1). But *n* is increased by one when one more node is selected. The query will be sent to the selected peer nodes only when the number of selected nodes *n* is smaller than its ranking to the query *r* ($n < r$). If $n \geq r$, node selection procedure is completed in the second phase. If all peer nodes associated with the area of the requested topic ($c > 0$) have been taken from the list in the second phase but there still are not enough nodes $n < FN_{\min}$, the selection procedure will move to the third phase to randomly pick up peer nodes from the rest of cached peer nodes irrelevant to the requested topic and its interest area ($c = 0$) with $FN_{\min}$.

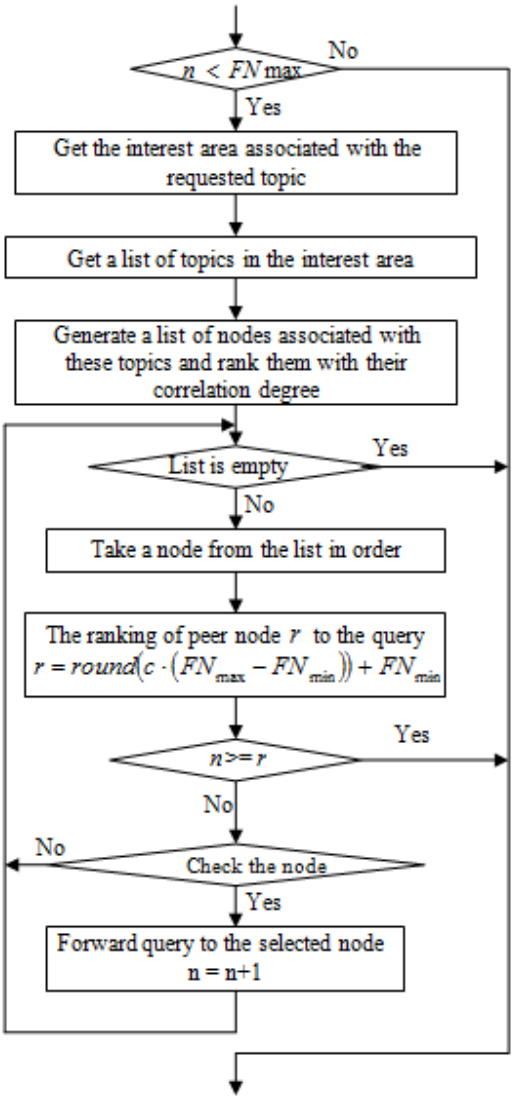


Figure 2. Flowchart of the second phase of node routing algorithm.

Figure 3(a) illustrates a simple example of query routing with the Social-P2P algorithm where $FN_{\max} = 4$ and $FN_{\min} = 1$. Node A receives a query with the topic "radar remote sensing". In the first phase, node A finds the node B from its local knowledge index which is directly associated with the topic "radar remote sensing". Due to $n < FN_{\max}$ ($n = 1$, $FN_{\max} = 4$), node A searches for the peer nodes associated with the topics "optical remote sensing" and "laser remote sensing" from the same interest area of the requested topic. In this case, node A gets node C and node D associated with these topics from the knowledge index. Since node C is associated

with two topics and node D is associated with only one topic in the area with three topics, node C ($c_C = 2/3$, $r_C = 3$) is more correlated with the area of remote sensing according to the cached knowledge than node D ($c_D = 1/3$, $r_D = 2$). Hence, node C will be sorted on the top of node D in the list. The query will be sent to node C, because the number of selected nodes is smaller than the ranking of node C, $n < r_C$ ($n = 1$, $r_C = 3$). Then $n + 1 \rightarrow n = 2$ and the selection procedure will stop because $n \geq r_D$ ($n = 2$, $r_D = 2$). The actual number of queried nodes in this case is two.

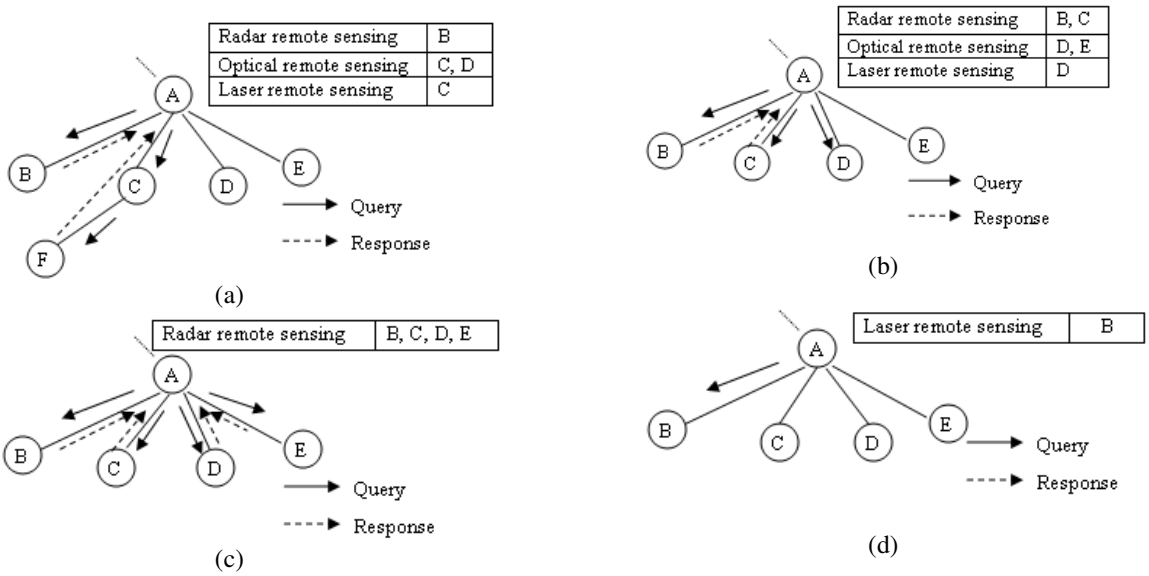


Figure 3. Examples of query routing of Social-P2P. (a) Two nodes are selected. (b) Three nodes are selected. (c) Four nodes are selected. (d) One node is selected.

Node C may not have the requested files, but it will use its own cached knowledge to propagate the query further and find the peer nodes for the query that will have a higher likelihood. In this example, node C knows that node F is associated with the requested topic according to its local knowledge index and the requested files are obtained in node F. In the cases illustrated in Figure 3(b), (c) and (d), the actual number of nodes to be queried is change to 3, 4 and 1, respectively, according to the different cached knowledge.

In social networks, a person builds his/her personal network by the result of experiences in previous interactions with other people. Generally, a personal network is a set of people that are preferably contacted by an individual person to get information or advice. Similarly, in the Social-P2P network, a node builds its social network by connecting to other peer nodes according to the results of previous searches. If a search is successful, the requesting node will link to the remote nodes that supplied the requested files.

In social networks, some events with associated people fade from a person's memory with time and a person's social network is keeping up with changing environments. Similarly, in the Social-P2P network, the size of knowledge index is finite and the node connections are adjusted with cached knowledge. The knowledge index is maintained in a queue without duplicates. The oldest knowledge will be dropped when the knowledge index reaches a maximum.

It is not necessary for a peer node to declare its interest since it already has been implied by its shared files, which is similar to social networks where a person does not need to tell everybody he/she is an expert in the areas which has been indicated with his/her social behaviours. Because connections are built according to the results of searches, a node has more probability to connect to other peer nodes with the same interests that have files of interest to him/her with a high degree of likelihood. Therefore, the peer nodes that have the same interests are highly connected to each other and form a virtual community spontaneously, which is a similar environment to Watts's model [11] in social networks.

3.2. Active Social-P2P

Recall that people remember potentially useful knowledge from social interactions. However, in social networks, people not only passively learn knowledge by remembering useful information from daily occasional events, but also actively collect potentially useful knowledge

consciously. For example, when people seek out help from friends, they are often introduced to new acquaintances who can be of assistance. But the communication with the new acquaintance is not normally limited to a strict exchange of assistance; often, the person receiving help will inquire more about the new acquaintance in order to expand his/her knowledge of that person's abilities which may be useful at a later time.

We also extend Social-P2P by involving such active social behaviours (Active Social-P2P). In Active Social-P2P, when a node responds to the query successfully, the requesting node will actively collect knowledge by further querying the new node for information about more topics of interest. In the simulations, the responding node will be further queried by the topics in the interest area of the requesting node. The obtained new information will be put into the knowledge index for future queries. With these active behaviours, Active Social-P2P can gather more pieces of knowledge from each successful query, but additional traffic will be generated for shipping such additional knowledge.

4. SIMULATION METHODOLOGY

4.1. Content Creation and Distribution

We try to build a near-realistic environment to evaluate the performance of Social-P2Ps by simulations. Most of simulations settings are according to the measurement studies of P2P networks. The topic keyword distribution to files is uneven in P2P file-sharing networks, where popular topics are widely distributed to files but unpopular topics receive little attention by people. Previous studies (e.g. [33]) observed that the distribution of keywords in files could be approximated by Zipf's law in the form of $y \sim \frac{1}{x^{\alpha}}$, where y is frequency, x is rank and α is constant. The estimated distribution in the study [33] was followed in our simulations to generate

topic keyword distribution to files. In each simulation run, we generated 1600 topics, distributed them to 10000 files, and each file was randomly assigned 3 topics.

Previous measurement studies have shown the distribution of the number of shared files in P2P networks is also unbalanced. Some nodes observed in existing P2P networks tend to download a large amount of files, but share few files or none at all [34]. In the simulations, we implemented the distribution of file sharing in the measurement study [33]. In the simulations, each peer node was assigned a primary interest area and shared a number of files to the network with a probabilistic method: these shared files were mostly relevant to the interest area of a node with a probability of 90%, but occasionally were irrelevant to this area. For files relevant to the interest area, at least one of the topics of each file should be in the interest area of the hosting node. A total of 40 interest areas were generated and each covered 40 topics.

4.2. Request Generation

In each time step of the simulations, we randomly chose an online node as the requesting node and started a search with a topic. The requested topic was generated with a probabilistic method: the topic is randomly selected from the interest area of the requesting node with a probability p ($p = 90\%$, if no other value is specified), but sometimes from a random area with a probability of $(1-p)$. Each query was tagged by a *TTL* to limit the life time of a message to 3 hops with FN_{max} = 5 and FN_{min} = 2, if no other setting is specified. Even though the request frequency was variable for different users in different periods, the study [36] observed that each peer node generates an average of two requests each day. This has been implemented in our simulations.

Ren [37] argued that user interest shift is a vital factor for P2P file-sharing networks, especially in today's dynamic information era. To address this issue, 1% of peer nodes randomly

shifted their interest each day in the simulations, if no other setting is mentioned. Their major requests and additional file sharing will follow the new interests after shifting interest.

4.3. Network Initialization

The studies in [33], [35] suggested that some P2P file-sharing networks (e.g. Gnutella) are scale-free networks where the connectivity of peer nodes follows a scale-free power-law distribution: $p(k) = \alpha \cdot k^{-\gamma}$. The probability $p(k)$ that a node in the network connects with k other nodes is proportional to $k^{-\gamma}$. The factors affecting the distribution of connectivity are various (e.g. preference to early entrants, preference to more powerful and well-connected nodes, preference to nodes sharing more useful content, etc). Therefore, it is unreasonable to generate a random power-law distribution of connectivity in the simulations irrespective of other characters of peer nodes. In order to better observe the evolution of network topology in the simulations, we started from a small-size random network (with 100 nodes). Each peer node randomly connected to 4 peer nodes bi-directionally to generate a random topology, so each peer node kept about 8 links at start-up of the simulations. In the beginning of each simulation run, since there were no interactions between peer nodes, each peer node kept an empty knowledge index which can contain a maximum of 80 topics and associated peer node addresses (if no other size is specified).

4.4. Network Evolution

Some popular P2P networks are growing very fast on the Internet according to media reports. However, some measurement studies (e.g. [38]) observed that the size of some mature P2P networks stayed constant. The phenomenon of quick growth to stability has not been considered by most P2P simulations. In our simulations, we simulated a growing network started with a small set of peer nodes (100 nodes). A number of peer nodes (100 nodes) joined the

network every 3 days (6000 time steps) in the first month until it reached 1000 nodes. Then the network became a mature network with 1000 nodes, but the peer nodes were still present and absent from the network frequently with a random distribution described in the next paragraph. We ran simulations to trace the results of about two months (60 days, 120000 time steps). Even though the simulations were undertaken with a medium-sized P2P network (1000 nodes) and over a short term (two months), the simulations of a growing network is representative of the evolution of larger P2P networks over longer periods.

4.5. Network Churn

In the dynamic and unpredictable Internet environment, network churns are usually caused firstly by peer nodes frequently going online and offline and secondly by content sharing and removing. High churns significantly influence the performance of P2P systems, which even lead to the maintenance difficulty of consistent DHTs [6] in structured P2P systems. The study [38] measured network churn by using a user ID instead of an IP address that was used in previous measurement studies (e.g. [35]), because IP address aliasing is a significant issue in the deployed P2P systems (almost 40% of peer nodes use more than one IP address over one day according to their measurements [38]). Therefore, our simulations followed the availability data in the study [38].

Content sharing of each peer node is changing with time and users' interest, which has seldom been considered by previous P2P simulations. To simulate the dynamics of file sharing, we randomly picked 1% of peer nodes to add a file to the network and 1% of peer nodes to remove a file from the network every day. Network churns in this case could affect the "correctness" of information in the knowledge index. The selected peer nodes that previously had the requested files could be offline from the network at the moment of requesting or the

requested files that were previously available on the selected nodes could have already been removed from the network.

4.6. Performance Metrics

Performance was evaluated with the following metrics:

- Recall: the ratio of the number of files found to the number of all the files that match the query in the network. For example, there are a total of 100 requested files in the network, but only 5 files are found. The recall is 5% in this case.
- Average number of found files: the average number of files found that match the query.
- Average path length of searches: the average distance from the requesting node to the targeted node which first finds a matched file. If none are found, the average path length of the search is set as 4 ($TTL+1$) in the simulations.

In Watts's model [11], a small world network is a kind of network with a high clustering coefficient of nodes and a short average path length to other peer nodes. These two properties of small world networks were recorded to observe topology evolution in the simulations:

- Average path length to nodes: the average of the shortest distances between any two peer nodes in the network.
- Average clustering coefficient: the average of the clustering coefficients of all nodes in the network. The clustering coefficient of a node is the proportion of the links between nodes within its neighbourhood divided by the possible number of links between them.

The results shown in Section 5 were average values calculated from the experimental results of each simulation day (2000 queries, 2000 time steps).

5. SIMULATION RESULTS

Many search methods are emerging in unstructured P2P system in the last decade, as reviewed in Section 2. Most of existing search methods, such as [21] and [26], are obviously different from the methods we are presenting in this paper. In contrast, NeuroGrid is a well-known method closely related to Social-P2P, which enables peer nodes to learn the results of previous searches to make future searches more focused. In this section, NeuroGrid will be simulated and analysed as a benchmark to evaluate the performance of Social-P2P.

Additionally, as the first unstructured P2P search method, the Gnutella-like random protocol has been widely used as a benchmark for many follow-up unstructured protocols (e.g. [7]), which is also simulated and compared as a benchmark to evaluate the performance of NeuroGrid.

The simulations are performed to provide a comparison among the Social-P2Ps and two relevant methods: Random and NeuroGrid:

- Random: a constrained Gnutella routing strategy. When a peer node receives a query, the received queries are randomly passed to FN_{min} connected peer nodes in each hop.
- NeuroGrid: each peer node builds a knowledge index with results of previous searches. If a search is successful, the requesting node updates its knowledge index to associate the peer nodes that have responded data successfully with the requested topic. When a peer node receives a query, the received query will be passed to peer nodes directly associated with the requested topic from the knowledge index in each hop. If not enough matches are found ($< FN_{min}$), the algorithm randomly forwards the query to peer nodes from the rest of the connected nodes.

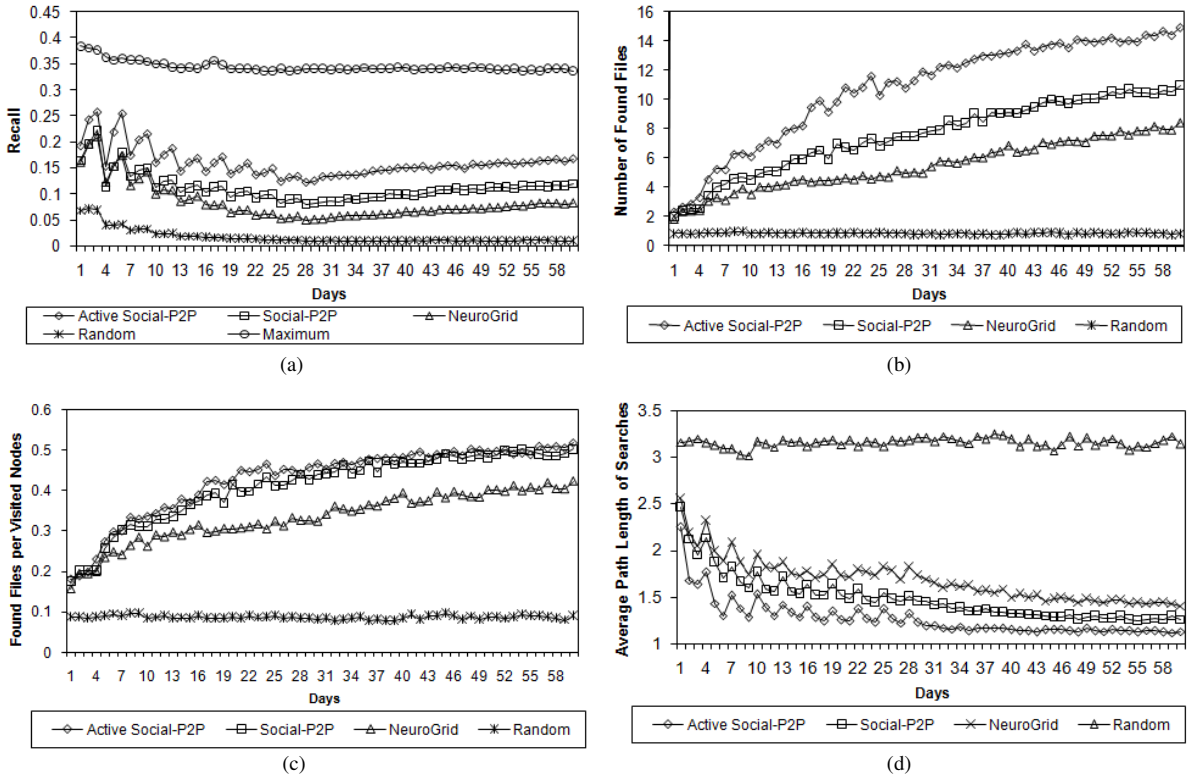


Figure 4. Performance comparison. (a) Recall. (b) Number of found files. (d) Number of found files per visited node. (e) Average path length of searches.

5.1. Performance Evaluation

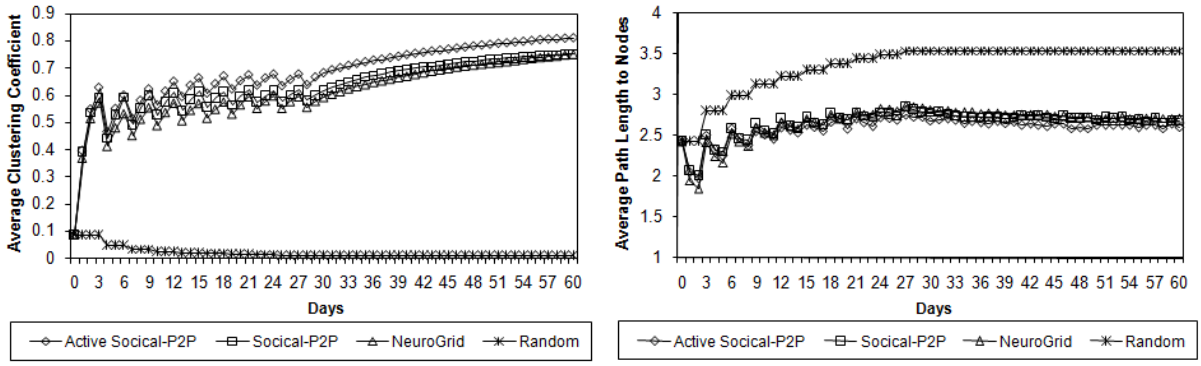
As shown in Figure 4(a) and (b), Social-P2Ps achieve better performances than the others by more quickly and efficiently retrieving more files. In Figure 4(a), the maximum possible recalls are all below 40%, because a large amount of files are available on a large number of offline nodes. As shown in Figure 4(a), the recalls of all simulated methods are also in a low-value area by setting a small *TTL* ($TTL = 3$). Since many new files are added into the network by newly joining peer nodes, the recall decreases during the network growing period, while the number of found files increases. Because new peer nodes joined the network per three days in the first month, periodic oscillations are seen in Figure 4.

At the early stage of searches, it is very difficult for peer nodes to find directly associated nodes with the requested topic by using either Social-P2Ps or NeuroGrid method due to limited

knowledge cached, but Social-P2Ps are capable of retrieving the peer nodes who share associated files with the relevant topics more often. These selected peer nodes that are highly correlated with the interest area of the requested topic have more knowledge about the query than random nodes. Therefore, Social-P2Ps can find the requested files more efficiently with the same knowledge. More successful searches, in turn, help to build the knowledge index more efficiently. Therefore, Social-P2Ps have better search capabilities and better knowledge-collecting capabilities, which enable peer nodes to search the network more efficient by finding more files per query message as shown in Figure 4(c). With these advantages, Social-P2Ps achieve better performances than the other methods. With active social behaviours, Active Social-P2P can more quickly establish a knowledge index than Social-P2P by gathering more pieces of knowledge from each successful query. Because Active Social-P2P can quickly accumulate a large amount of useful knowledge about file locations in a short term, the average path length of searches quickly decreases to just above one, as shown in Figure 4(d).

5.2. Topology Evolution

Figure 5(a) and (b) show the clustering coefficient and the average path length to nodes observed in the simulations. As shown in Figure 5(a), the clustering coefficients of Social-P2Ps are greater than other methods. We also compare the clustering coefficient of Active Social-P2P to that of a randomly connected network with the same number of nodes and connections. The clustering coefficients of the randomly connected network are given by the equation: $C \approx \langle k \rangle / N$ [39], where $\langle k \rangle$ is the average node degree of the network and N is the total number of nodes in the network. As shown in Figure 6, the clustering coefficient of Active Social-P2P systems is much greater than that of the randomly connected network with the same number of nodes and connections.



(a) (b)

Figure 5. (a) Average clustering coefficient. (b) Average path length to nodes.

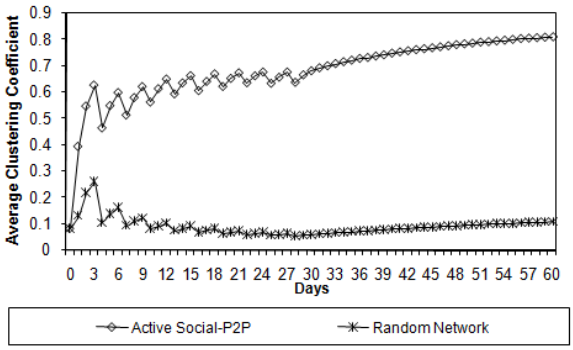
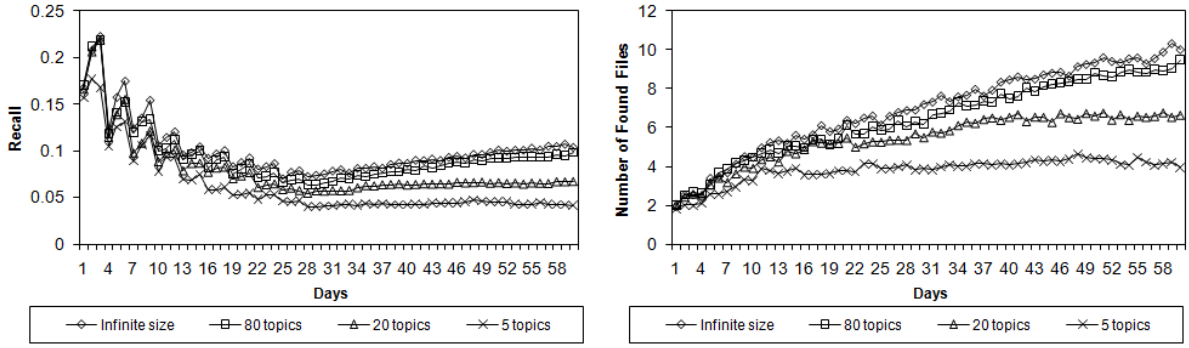


Figure 6. Clustering coefficient comparison between Active Social-P2P and a random network with the same number of nodes and connections.

Even though the network size increases quickly at the early stage of the simulations, the average path length to nodes only increases a little except the case of the Random method (as shown in Figure 5(b)) due to increasing connectivity of the network. The average path lengths to nodes of Social-P2Ps are only slightly smaller than those of NeuroGrid due to using the same connection adaptation strategy. However, by using different routing strategies, their search performances are significantly different as shown in Figure 4. The simulation results show that the small-world phenomenon also appears in the Social-P2Ps with a high clustering coefficient of nodes and short average path length to other peer nodes.

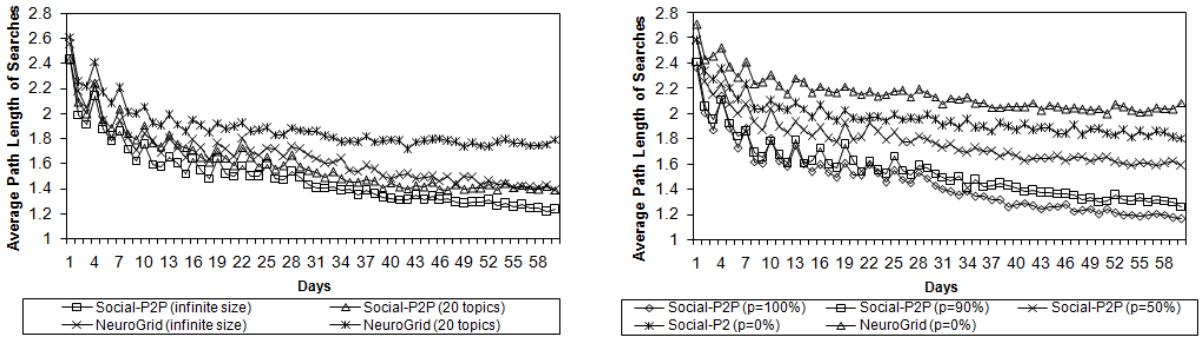
5.3. Knowledge Size



(a) (b)

Figure 7. (a) Recall with different sizes of knowledge index. (b) The number of found files with different sizes of knowledge index.

Figure 7(a) and (b) show the results of recall and the number of found files by Social-P2Ps where each node has a knowledge index containing a maximum of 5, 20, 80, infinite topics and associated peer node addresses. Clearly in Figure 7, the requesting nodes have more difficulty in finding the requested files from nodes with less knowledge. The results in Figure 7(a) also suggest that a large knowledge index containing most commonly used topics achieves close performance to a knowledge index with an infinite size. Hence we defined the knowledge index with a maximum of 80 topics for other experiments.



(a) (b)

Figure 8. Average path length of searches. (a) Different size of knowledge index. (b) Different request structures.

Figure 8(a) shows the comparison of the average path lengths of searches by Social-P2P and NeuroGrid with a knowledge index containing a maximum of 20 topics and infinite topics. As

shown in Figure 8(a), the average path length of Social-P2P with a size of 20 topics, only increases a little from about 1.3 of infinite size to around 1.4. In contrast, the average path length of NeuroGrid changes significantly from about 1.4 of infinite size to around 1.8. Since NeuroGrid chooses nodes randomly if it can not find enough directly associated nodes to queries, the limited size of knowledge index more significantly affects the search performance of NeuroGrid than that of Social-P2P which can also find the peer nodes that are able to give a good referral. Therefore, Social-P2P can find the requested files more efficiently and quickly based on the same small pieces of knowledge.

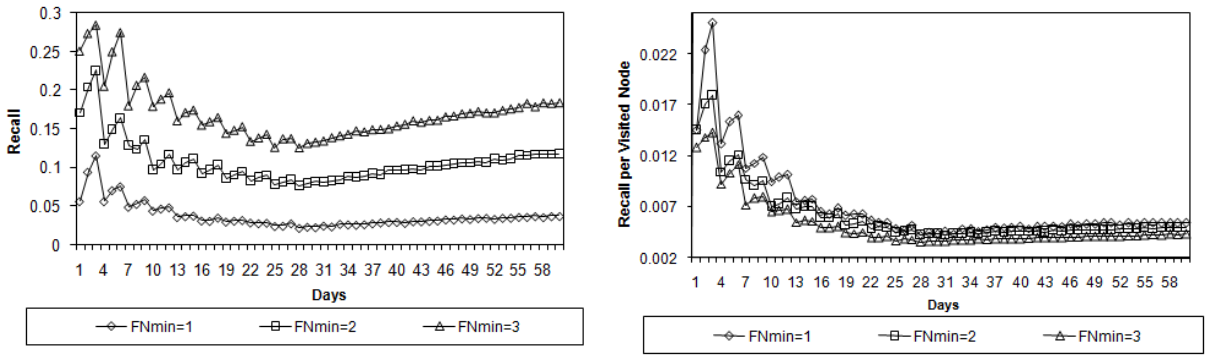
5.4. Request Structure

We simulated Social-P2P with different request structures. Recall that the requested topic was selected from the interest area of the requesting node with a probability p, but was not from the area with a probability $(1-p)$. In the case of $p = 0\%$, a purely random topic was chosen as the requested topic which is the worst case since the requesting peer node cannot benefit from the repeated queries in its interest area. On the contrary, in the case of $p = 100\%$, all requested topics were randomly selected from the interest area of the requesting node. Figure 8(b) shows the results of average path length of searches by Social-P2P on some representative samples of p of 0%, 50%, 90%, and 100%, respectively. In this simulation, the request scope was enlarged by setting a smaller p. Since the probability of matching cached knowledge decreases with p, the average path length of each search increases along with p which means the peer nodes generally need more hops to target the requested files in the network where users have very wide interests. But the performance of Social-P2P is still better than that of NeuroGrid method even in the worst case of $p = 0\%$ as shown in Figure 8(b), because Social-P2P can still find the peer nodes that

potentially have the knowledge about queries even though it can not find the directly associated peer nodes from the knowledge index.

5.5. Number of Receivers

The minimum number of receivers FN_{min} and the maximum number of receivers FN_{max} are important factors to achieve adaptive query forwarding. In this experiment, Social-P2P is simulated with different FN_{min} and FN_{max} to see the effect of each setting makes.

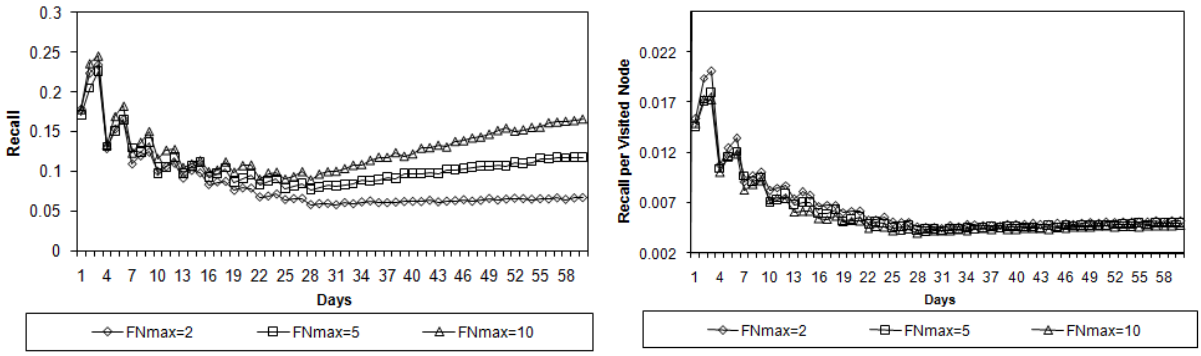


(a) (b)

Figure 9. (a) Recall with alteration of the minimum number of receivers. (b) Recall per visited node with alteration of the minimum number of receivers.

As shown in Figure 9 (a) and (b), recall increases by changing FN_{min} from 1 to 3, while recall per message decreases with increasing FN_{min}. However, if a very small $FN_{\min}$ is defined as $FN_{\min} = 1$, a very limited peer nodes could be accessed for answering a query, which seriously affects the speed of knowledge collection. Therefore, recall is very low in the case of $FN_{\min} = 1$.

The effect of different maximum numbers of receivers is tested in the experiment on some representative samples of FN_{max} of 2, 5 and 10. The algorithm does not achieve good performance by defining a small FN_{max}, since useful nodes that may not be included in the knowledge index cannot be reachable.



(a) (b)

Figure 10. (a) Recall with alteration of the maximum number of receivers. (b) Recall per visited node with alteration of the maximum number of receivers.

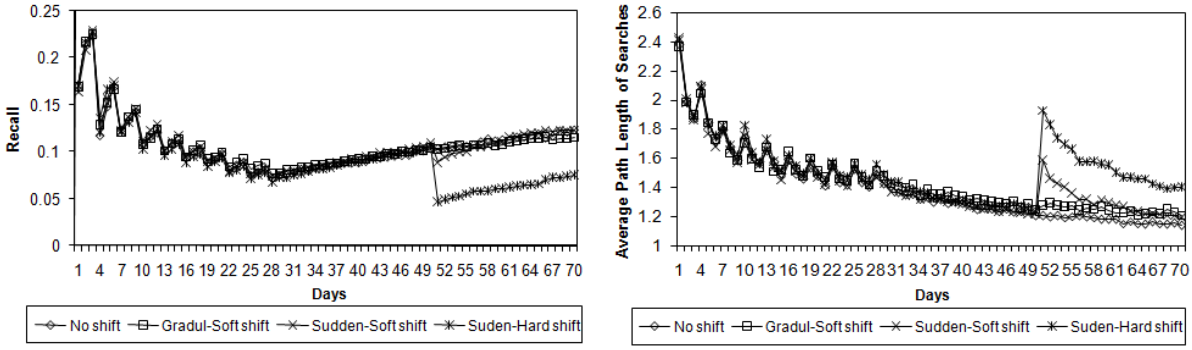
Compared to the results with changing the minimum number of receivers FN_{min} shown in Figure 9, recall and recall per visited node are more sensitive to the alternation of the minimum number of receivers FN_{min} rather than the alternation of the maximum number of receivers FN_{max} as shown in Figure 10(a) and (b). Recall per message is only slightly changed by alternation of FN_{max}, since adaptive lookups are achieved by Social-P2P. Since high performance and efficiency are achieved by a bigger FN_{max}, a bigger FN_{max} could be defined to achieve a higher recall rather than a bigger FN_{min}.

5.6. Interest Shifts

We further simulated the Social-P2P with the different kinds of interest shifts:

- Gradual shift: peer nodes shift their interest gradually, which is the same as the interest shift we did in the previous simulations: 1% of peer nodes randomly shifted their interest each day.
- Sudden shift: high number of peer nodes changes their interests in a short time interval. We defined that 60% of nodes changed interests suddenly on the 50th day.
- Soft shift: the additional file sharing will follow the new interest, but the peer nodes will not remove all previous shared files when shifting interest.

• Hard shift: the peer nodes will replace all shared files with new files when shifting interest and the additional file sharing will follow the new interest.



(a) (b)

Figure 11. (a) Recall with different interest shifts. (b) Average path length of searches with different interest shifts.

From the results in Figure 11(a) and (b), the gradual and soft interest shift achieves a similar performance to the network without interest shift. The gradual and soft interest shift can be handled by Social-P2P. The average path length of searches experiences a clear increase at the moment of sudden and soft interest shift, but the performance quickly recovers back to normal. However, a large-scale sudden and hard shift affects performance more significantly which is also hard to recover. Therefore, massive content changes caused by hard interest shifts aggravate the effect of interest shift, which suddenly produces a large amount of invalid knowledge in the network.

6. CONCLUSIONS

Due to the similarity of social networks and P2P networks, we believe and demonstrate that human tactics in social networks are useful for improving P2P object discovery by building a social-like P2P network. In this paper, we presented Social-P2P and Active Social-P2P for object discovery by self-managing autonomous peers with social tactics. With these methods, queries

are routed efficiently even with limited knowledge and node connections. More successful searches of Social-P2Ps, in turn, help to build the knowledge index more efficiently.

Social-P2Ps have been simulated with probabilistic request structure and file sharing in a near-realistic environment with a growing number of peer nodes. From the simulation results and analysis, Social-P2P and Active Social-P2P achieved better performances, more quickly targeted more requested files and more efficiently established a knowledge index about the location of files, than current methods. With the active behaviours, Active Social-P2P achieved even better performance by gathering more pieces of knowledge from each successful query. Social-P2Ps have been further simulated with different sizes of knowledge index, with different request structures, and with different kinds of interest shifts. Moreover, the small-world phenomenon has been observed in Social-P2Ps with a high clustering coefficient and a short average path length to nodes. In future work we will further optimize social-like P2P algorithms and simulate the algorithms in a larger-scale P2P system over longer periods to analyse the evolution of the social-like P2P network. Moreover, since the number of topics may be different in different files, social-like P2P algorithms will be further simulated by setting individual number of topics for each peer node.

REFERENCES

[1] L. Liu, N. Antonopoulos, S. Mackin, Social Peer-to-Peer for Resource Discovery. In: 15th Euromicro International Conference on Parallel, Distributed and Network-based Processing, pp. 459-466, IEEE Computer Society Press, Naples, Italy, February 2007.

[2] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In: ACM SIGCOMM, pp.149-160, San Diego, CA, August 2001.

[3] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, A Scalable Content-Addressable Network. In: ACM SIGCOMM, pp.161-172, San Diego, CA, August 2001.

[4] A. Rowstron and P. Druschel, Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems. In: IFIP/ACM International Conference on Distributed Systems Platforms, Heidelberg, Germany, November 2001.

[5] N. Antonopoulos and J. Salter, Efficient Resource Discovery in Grids and P2P Networks, *Journal of Internet Research*, 14:339-346, 2004.

[6] S. Rhea, D. Gells, T. Roscoe, and J. Kubiatowicz, Handling Churn in a DHT. In: the USENIX Annual Technical Conference, Boston, MA, June 2004.

[7] B. Yang and H. Garcia-Molina, Efficient Search in Peer-to-Peer Networks. In: International Conference on Distributed Computing Systems, Vienna, Austria, July 2002.

[8] P. Maymounkov and D. Mazieres, Kademlia: A Peer-to-Peer Information System Based on the XOR Metric. In: IPTPS, Cambridge MA, March 2002.

[9] S. Milgram, the Small World Problem. *Psychology Today*, 2:60-67, 1967.

[10] T. Hong, Chapter Fourteen: Performance. Peer-to-Peer: Harnessing the Power of Disruptive Technologies, pp. 03-241, O'Reilly, 2001.

[11] D. Watts and S. Strogatz, Collective Dynamics of Small-World Networks. *Nature*, 393:440-442, 1998.

[12] J. Salter, N. Antonopoulos, An Optimised 2-Tier P2P Architecture for Contextualised Keyword Searches, *Elsevier Future Generation Computer Systems*, 23:241-251, 2007.

[13] J. Kleinberg, Small-World Phenomena and the Dynamics of Information, *Advances in Neural Information Processing System (NIPS)*, 14:431-438, 2001.

[14] L. Liu, N. Antonopoulos, S. Mackin, Fault-tolerant Peer-to-Peer Search on Small-World Networks*, Future Generation Computer Systems*, 23: 921-931, 2007.

[15] H. Zhang, A. Goel, and Govindan, Using the Small-World Model to Improve Freenet Performance, *Computer Networks*, 46:555-574, 2004.

[16] F.M. Cuenca-Acuna and T.D. Nguyen, Text-based Content Search and Retrieval in ad hoc P2P Communities, In: International Workshop on Peer-to-Peer Computing, Pisa, Italy, May 2002.

[17] M.S. Khambatti, K.D. Ryu, and P. Dasgupta, Efficient Discovery of Implicitly Formed Peer-to-Peer Communities, *International Journal of Parallel and Distributed Systems and Networks*, 5:155-164, 2002.

[18] J. Vassileva, Motivating Participation in Peer-to-Peer Communities. In: Workshop on Engineering Societies in the Agent World, Madrid, Spain, March 2002.

[19] R. Karp, S. Shenker, C. Schindelhauer, and B. Vocking, Randomized Rumour Spreading. In: 41st Symposium Foundation on Computer Science, Warsaw, Poland, August 2002.

[20] B. Bloom, Space/time Tradeoffs in Hash Coding with Allowable Errors. *Communication of ACM*, 13:422-426, 1970.

[21] A. Crespo and H. Garcia-Molina, Routing Indices for Peer-to-Peer Systems. In: International Conference on Distributed Computing Systems, Vienna, Austria, July 2002.

[22] JXTA. Available: http://www.jxta.org.

[23] Bearshare. Available: http://www.bearshare.com.

[24] L. Xiao, Y. Liu, and L.M. Ni, Improving Unstructured Peer-to-Peer Systems by Adaptive Connection Establishment, *IEEE Transactions on Computers*, 54:176-184, 2005.

[25] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, Making Gnutella-Like P2P Systems Scalable. In: ACM SIGCOMM, Karlsruhe, Germany, August 2003.

[26] D. Tsoumakos and N. Roussopoulos, Adaptive Probabilistic Search for Peer-to-Peer Networks. In: International Conference on Peer-to-Peer Computing, Linkoping, Sweden, September 2003.

[27] S. Joseph, NeuroGrid: Semantically Routing Queries in Peer-to-Peer Networks. In: the International Workshop on Peer-to-Peer Computing, Pisa, Italy, May 2002.

[28] S. Joseph, P2P MetaData Search Layers. In: International Workshop on Agents and Peer-to-Peer Computing, Melbourne, Australia, July 2003.

[29] N. Borch, "Social P2P for Social People," In: International Conference on Internet Technologies & Applications, Wrexham, UK, September 2005.

[30] J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. Epema, M. Reinders, M.v. Steen, and H. Sips, "Tribler: A Social-based Peer-to-Peer System," *Concurrency and Computation: Practice and Experience*, 19: 1-11, 2007.

[31] The Open Directory Project. Available: http://dmoz.org/.

[32] H. Kautz, B. Selman and M. Shah, Combining Social Networks and Collaborative Filtering. In: *Communications of ACM*, 40:63-65, 1997.

[33] P. Makosiej, G. Sakaryan, H. Unger, Measurement Study of Shared Content and User Request Structure in Peer-to-Peer Gnutella Network. In: International Conference on Design, Analysis, and Simulation of Distributed Computing System, Arlington, Virginia, April 2004.

[34] C. Pauli, M. shepperd., An Empirical Investigation into P2P File-Sharing User Behaviour. In: Americas Conference on Information Systems, Omaha, Nebraska, August 2005.

[35] S. Saroiu, A Measurement Study of Peer-to-Peer File Sharing Systems. In: International Conference on Multimedia Networking and Computing, Santa Barbara, CA, October 2002

[36] P. Krishna, Measurement, Modelling and Analysis of a P2P File-sharing Workload. In: ACM Symposium on Operating Systems Principles, Bolton Landing, New York, October 2003.

[37] Y. Ren, et al., Explore the "Small World Phenomena" in Pure P2P Information Sharing System. In: International Symposium on Cluster Computing and the Grid, Tokyo, Japan, May 2003.

[39] R. Bhagwan, S. Savage, G.M. Voelker, Understanding Availability. In: International Workshop on Peer-to-Peer System, Berkeley, CA, February 2003.

[39] H. Zhou, Scaling Exponents and Clustering Coefficients of a Growing Random Network, *Physical Review*, 66: 016125, 2002.