

# Social Peer-to-Peer for Resource Discovery

Lu Liu  
Surrey Space Centre  
University of Surrey  
Guildford, United Kingdom  
l.liu@surrey.ac.uk

Nick Antonopoulos  
Department of Computing  
University of Surrey  
Guildford, United Kingdom  
n.antonopoulos@surrey.ac.uk

Stephen Mackin  
Surrey Satellite Technology Limited  
Surrey Research Park,  
Guildford, UK  
s.mackin@sssl.co.uk

## Abstract

*For resource discovery in social networks, people can directly contact some acquaintances that have knowledge about the resources they are looking for. However, in current peer-to-peer networks, peer nodes lack capabilities similar to social networks, making it difficult to route queries efficiently. In this paper, we present a social-like system (Social-P2P) for resource discovery by mimicking human behaviours in social networks. Different from most informed search algorithms, peer nodes learn knowledge from the results of previous searches and no additional overhead is required to obtain extra information from neighbouring nodes. Unlike community-based P2P information sharing systems, we do not intend to create and maintain peer groups or communities consciously. Peer nodes with the same interests will be highly connected to each other spontaneously. Social-P2P has been simulated in a dynamic environment. From the simulation results and analysis, Social-P2P achieved better performance than current methods.*

## 1. Introduction

In human society, people are connected by their social relationships. Small world phenomenon [1], observed by Stanley Milgram, is the hypothesis that everyone in this world can be reached through a short chain of social acquaintances, known as “six degrees of separation.” Duncan Watts proposed a mathematical model [2] 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. He demonstrated that the path-length between any two nodes of his model is surprisingly small. For resource discovery in social networks, people can directly contact some acquaintances that have knowledge about the resources

they are looking for. However, in current peer-to-peer networks, peer nodes lack capabilities similar to social networks, making it difficult to route queries efficiently.

Existing solutions for resource discovery in the P2P systems can be generally classified into two categories: structured and unstructured P2P systems. Structured P2P systems (e.g. Chord [3], CAN [4], and Pastry [5]) have dedicated network structure on the overlay network. Distributed hash tables (DHTs) have become the dominant methodology for resource discovery in structured P2P networks [6]. However, some recent studies (e.g. [8], [9]) argued that the cost of maintaining a consistent distributed index is very high in the dynamic and unpredictable Internet environments. Some structured P2P protocols (e.g. Kademlia [7]) is seeking ways to save the cost of maintaining a consistent index. In contrast, unstructured P2P systems are more resilient in dynamic environments, but current unstructured P2P search techniques tend to either require high search overhead or generate massive network traffic.

Studies like [10], [11], and [12] 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 high-performance information dissemination techniques (e.g. Gossiping and Rumour Spreading [13]) and compact data structures (e.g. Bloom Filters [14]).

In this paper, we present a new social-like P2P algorithm (Social-P2P) for resource discovery by mimicking human interactions in social networks where peer nodes are people and connections are relationships. Different from most unstructured P2P systems (e.g. local indices [15]), no overhead is

required to obtain additional information from neighbouring nodes. Unlike community-based P2P information sharing systems, 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 result of daily searches. Finally, peer nodes with the same interests will be highly connected to each other spontaneously. The Social-P2P algorithm can be deployed on top of any unstructured P2P network (e.g. Gnutella) to improve the performance of resource discovery.

## 2. Related work

Though current search methods in unstructured P2P systems are heterogenous 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. [15]). Peer nodes normally maintain additional indices of files offered by connected 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. [16], [17]). Super-peer nodes are capable and reliable peer nodes that take more responsibility for providing services in P2P networks.

The third and fourth approaches are closely related to the algorithms we are presenting in this paper. In many P2P applications, topology determines performance. The third approach improves network performance by adapting and optimizing overlay topology (e.g. [18], [19]). In Social-P2P, the connections of peer nodes are adaptive with cached knowledge and only a number of associated connections are kept in each node. The fourth approach utilizes the historic record of previous searches to help peer nodes make routing decisions, such as Adaptive Probabilistic Search (APS) [20], NeuroGrid [21] and REMINDIN [22]. Different from self-organizing networks, 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. In the NeuroGrid network, peer nodes support distributed search through semantic

routing by maintaining routing tables at each node [21]. REMINDIN utilizes a similar principle as NeuroGrid which was implemented on the Semantic Web Application Platform (SWAP) [23]. However, both methods are only effective for previously queried keywords and are not suitable for networks where peer nodes come and go rapidly.

In addition, the "small world" social phenomenon has also been observed in current P2P networks. Maintaining and searching "small world" has been discussed in recent studies. Jon Kleinberg discussed the problem of decentralized search in P2P networks with partial information about the underlying structure in [24]. A study in [25] proposes an enhanced clustering cache replacement scheme by forcing the routing tables to resemble neighbour relationships in a small-world acquaintance graph. Small world architecture for P2P networks has been proposed in our previous work [26] with a semi-structured search algorithm in multi-group P2P systems, which has advantages of both structured and unstructured P2P approaches. The small world phenomenon has also been observed in our Social-P2P by mimicking social intercommunications as discussed in section 4.

## 3. Algorithm description

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

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 other peer nodes according to the results of searches. If a search is successful, the requesting node updates its knowledge index to associate the peer nodes that have responded data successfully and connects to these nodes. In the meantime, the requesting node also removes invalid cached knowledge according to the results of searches. 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 useful knowledge that is useful to quickly find the peer nodes with the required data in the future.

For resource discovery in social networks, people usually recall information 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, but people can find some acquaintances that potentially have knowledge about the resources they are looking for. 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 she has more knowledge about who has the dictionary. In this case, the Oxford English Dictionary is in the area of linguistics and Bob found Alice has abundant knowledge on the interest area of linguistics from previous intercommunications. Alice probably does not have 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 similar logic-based semantic approach to route queries to a subset of neighbouring nodes. 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.

This routing algorithm involves the following three phases. When a node receives a query which needs to be forwarded, the node routing algorithm firstly searches the local knowledge index for the peer nodes directly associated with the requested topic and ranks them with their updating time. The peer node that is input or updated more recently gets a higher rank. However, the success probability of the first phase is 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 peer nodes sharing content associated with the interest area of the requested topic from the local knowledge index. An interest area of Social-P2P is a semantic area with a set of topics. The corresponding interest area can be found from the Open Directory Categories [27], which is the most widely distributed database of Web content classified by humans. These peer nodes will be ranked according to the degree of correlation to the interest area of query. The routing algorithm prefers to select peer nodes with higher degrees of correlation rather than peer nodes with lower correlation. If two or more nodes have the same correlation degree, we put the peer node that responded most recently first. If a peer node has a large amount of content in a particular area, it is very likely that it will also have other data in this area. In our simulations, the correlation degree of a peer node in a particular area is generated by how many relevant topics in the area the peer node is

associated with:  $c = \frac{n_{matches}}{n_{total}}$ , 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.

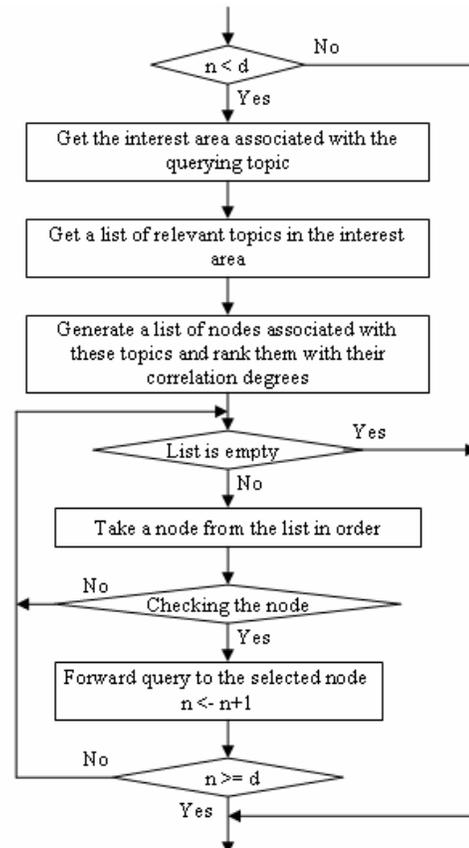


Figure 1. Flowchart of the second phase of node routing algorithm.

The flowchart shown in Figure 1 describes the second-phase of the routing algorithm used in our simulation, where  $n$  is the number of peer nodes that have been forwarded in the first and second phase and  $d$  is the defined number of peer nodes to forward a query to in each hop.  $n$  is increased by one when one more node is selected. The query will be sent to the peer nodes only if the number of forwarded nodes  $n$  is smaller than the defined number of peer nodes to be contacted in each hop  $d$  ( $n < d$ ). If  $n \geq d$ , the node selection procedure is completed in the second phase. If all peer nodes associated with the area of the requested topic have been taken from the list in the second phase, but there are still not enough nodes  $n < d$ , the selection procedure will move to the third phase to randomly pick up peer nodes from the rest of cached peer nodes.

Figure 2 shows an example of query routing with Social-P2P algorithm. Suppose node S receives a query with the topic “radar remote sensing.” Node S will retrieve the peer nodes associated with the topic “radar remote sensing.” However, no matches are found in the knowledge index in this case. Node S will further retrieve peer nodes associated with the relevant topics “optical remote sensing” and “laser remote sensing” from the same interest area of the requested topic. In this case, node S gets node A and node B associated with these topics from the knowledge index. Because node A is associated with both topics “optical remote sensing” and “laser remote sensing” and node B is only associated with one topic, node A is more correlated with the interest area of “remote sensing” according to the cached knowledge. The query will be forwarded more preferentially to node A than node B. Node A may not have the requested files, but it will use its own cached knowledge to propagate the query further and find peer nodes for the query that will have a higher likelihood of having the requested files. In this example, node A knows that node D is associated with the querying topic and the requested files are obtained in node D.

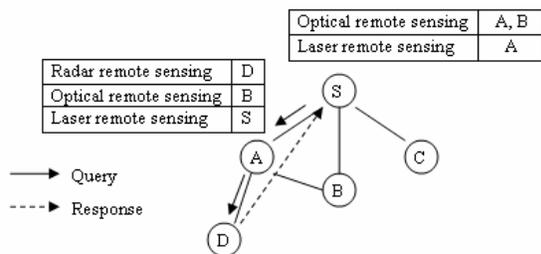


Figure 2. Social-P2P routing.

In social networks, a person builds his/her social network by the result of experiences in previous interactions with other people. Similar to social networks, 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 connect 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 adjustable with changed environments. Similarly, the size of knowledge index of Social-P2P is finite and the node connections are adjusted with cached knowledge. Social-P2P involves a topology adaptation method to resemble the dynamics of social connections. The knowledge index is maintained in a queue using a Least Recently Used (LRU) policy without duplicates. The oldest knowledge will be dropped when the knowledge index reaches a maximum. When some

peer nodes are removed with some old or invalid knowledge and no longer exist in the knowledge index, the corresponding connections will be released and only a number of associated connections are kept in each node.

In social networks, a social community is a group of people with common interests, goals or responsibilities. In Social-P2P, each peer node shares a number of files associated with its interests. It is not necessary for a peer node to declare its interest since that has already been implied by its shared files, which is similar to social networks where a person does not need to tell everybody that 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 Duncan’s model [2] in social networks. Because the virtual communities are formed spontaneously, no additional overhead is required to obtain extra information to maintain these communities compulsorily.

## 4. Simulation methodology

### 4.1. Network generation

We evaluated the performance of Social-P2P by simulation in a dynamic environment with 1000 peer nodes. Each peer node randomly connected to four peer nodes bi-directionally to generate a random topology. Each peer node kept about eight links at start-up of the simulations. Since there have been no interactions between peer nodes at the beginning of each simulation run, each peer node keeps a empty knowledge index which can contain a maximum of 40 entries about topics and associated addresses of peer nodes. We ran simulations to trace the results of about one month (30 days, 60000 time steps).

### 4.2. Content generation and distribution

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. The previous studies observed that the distribution of keywords in files could be approximated by Zipf’s law in the form of

$$y \sim \frac{1}{x^\alpha}, \text{ where } y \text{ is frequency, } x \text{ is rank and } \alpha \text{ is}$$

constant. The estimated distribution in the measurement study [28] has been followed in our simulations to generate topic keyword distribution to files. In each simulation run, we generated 1280 topics, distributed them to 10000 files, and each file was assigned two topics. Previous measurement studies have shown the distribution of the number of shared files to peer nodes 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 [29]. In the simulations, we implemented the distribution of file sharing in the measurement study [28], where about 40% of peer nodes share 10 files or less, including 27% of “free-riders” who share nothing to the network and 32% of peer nodes share more than 100 files.

The measurement study [30] for the music sharing network on Stanford shows that most peer nodes only shares one or a few styles of music that are highly correlated with users’ preferences. In our 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 primary interest area of node with a probability of 90%, but occasionally were irrelevant to this area. For files relevant to the primary interest area, at least one of the topics of each file should be in the interest area of the hosting node. A total of 32 interest areas were generated and each covered 40 topics.

### 4.3. Query generation

In each time step, we randomly chose a node as the requesting node and start a search with a topic. The querying topic was generated with a probabilistic method that the topic was randomly selected from its primary interest area with a probability of 90%, but sometimes was from a random area with a probability of 10%. Each query was tagged by *TTL* to limit the life time of message to 3 hops in the simulations. The number of peer nodes to be contacted in each hop  $d = 3$ . Even though the request frequency was variable for different users in different periods, the study [34] observed that each peer node generates an average of two requests each day. This has been implemented in our simulations.

### 4.4. Network churns

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. The study [31] measured network churns by using a user ID

instead of an IP address that was used by some previous measurement studies (e.g. [32]). 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 [31]). Therefore, our simulations followed the availability distribution of peer nodes in the study [31], where about 50% of peer nodes are present less than 30%.

The research in [33] 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. Their major requests and additional file sharing followed the new interests after shifting interest. To simulate the dynamics of file sharing, we randomly picked 1% of peer nodes to share extra 5% files to the network and 1% of peer nodes to remove 5% of shared files from the network every day (2000 queries, 2000 time steps).

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 peer nodes could have already been removed from the network.

### 4.5. Performance metrics

Performance is evaluated with the following metrics:

- Average path length of searches: the average of distances from the requesting node to the target node which firstly finds a matched file.
- Recall: the ratio of the number of found files to the number of all matched files in the network.
- 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. Clustering coefficient of a node is the proportion of the links between nodes within its neighbourhood divided by the number of links that could possibly exist between them [2].

The performance metrics were recorded and statistically analysed. Each average result was generated from the experimental results of each day (2000 queries, 2000 time steps).

## 5. Simulation results

### 5.1. Performance evaluation

We compare the performance of Social-P2P with the following methods:

- RAN: a constrained Gnutella-like routing strategy without a knowledge index. Received queries are randomly passed to a number of connected peer nodes.
- NEURO: a semantic routing strategy extended from NeuroGrid with adaptive connections that the node connections are adaptive with cached knowledge. Received queries are passed to peer nodes directly associated with the requested topic from the knowledge index. If not enough matches are found, the algorithm randomly forwards query to peer nodes from the rest of neighbours.

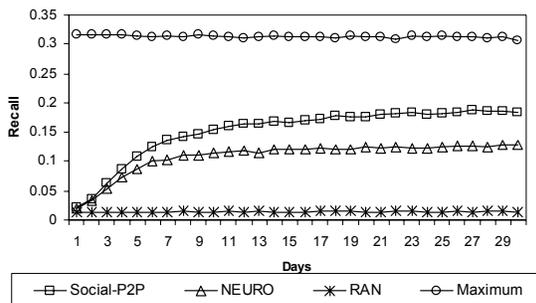


Figure 3. Recall of searches

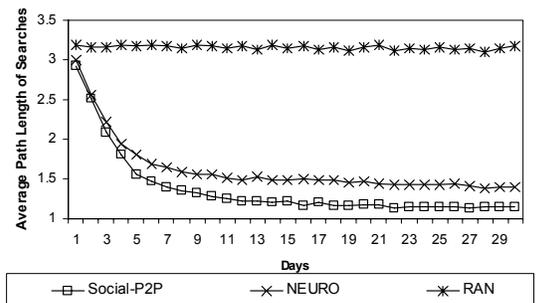


Figure 4. Average path length of searches

From the results in Figure 3 and Figure 4, Social-P2P achieved the best performance, which more quickly targeted more requested files and more efficiently established a knowledge index about the location of files than NEURO method. At the early stage of searches, it is very difficult for peer nodes to find directly associated peer nodes with the requested topic by using either Social-P2P or NEURO method, but Social-P2P is capable of retrieving the peer nodes who share associated files with the requested topic more often. These peer nodes that are highly correlated with the semantic area of the requested topic have

more knowledge about the query than random nodes. Therefore, Social-P2P can find the requested files more quickly and efficiently based on the same knowledge. More successful searches, in turn, help to build the knowledge index more efficiently. Therefore, Social-P2P has a better search capability and a better knowledge-collecting capability. With these advantages, Social-P2P achieved better performance than other methods. In Figure 3, the maximum possible recalls are all below 35%, because a large amount of files are available on a large number of offline nodes. As shown in Figure 3, the recalls of all simulated methods are also in a low-value area by setting a small  $TTL$  ( $TTL = 3$ ).

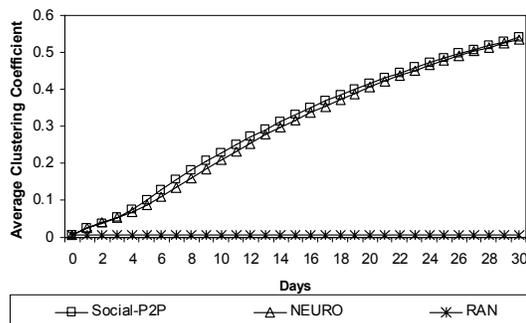


Figure 5. Average cluster coefficient.

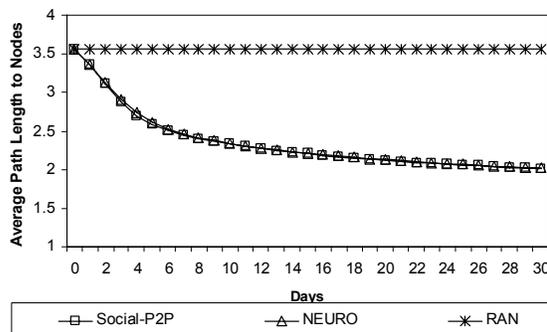


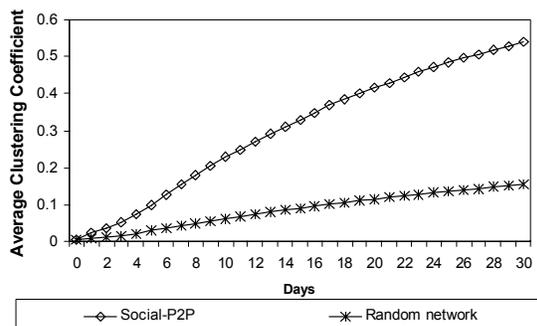
Figure 6. Average path length to nodes.

### 5.2. Topology evolution

In Duncan's model [2], 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. Figure 5 and Figure 6 show the comparison of the average cluster coefficient and average path length of Social-P2P, NEURO, and RAN, respectively. As shown in Figure 6, the average path length of Social-P2P is only slightly smaller than that of NEURO, since we imposed the same connection adaptation strategy to them. However, by using different routing strategies, their search performances are clearly different as

shown in Figure 3 and Figure 4. The average path lengths of Social-P2P and NEURO decrease with time and the peer nodes in the network are in the average of “two degrees of separation” in the end.

We also compare the clustering coefficient of Social-P2P to that of a random network with the same number of nodes and connections. The clustering coefficients of the random network are given by the equation [35]:  $C \approx \langle k \rangle / N$ , 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 7, the clustering coefficient of Social-P2P is much greater than that of the random network with the same number of nodes and connections. The simulation results show that the small-world phenomenon also appears in the Social-P2P with a high clustering coefficient and a short average path length to other peer nodes.



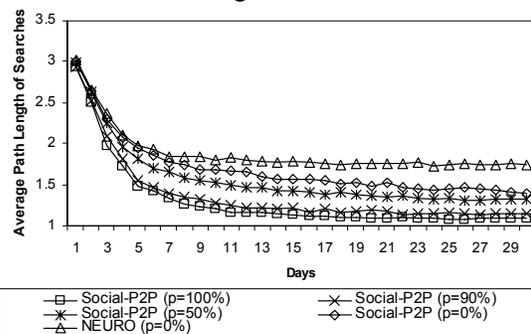
**Figure 7. Clustering coefficient comparison between Social-P2P and a random network with the same number of nodes and connections.**

### 5.3. Request structure

We simulated Social-P2P with different request structures. Recall that the requested topic was selected from the primary interest area of the requesting node with a probability  $p$ , but was from a random 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 repeated queries in its interest area. On the contrary, in the case of  $p=100\%$ , all requested topics were randomly selected from the primary interest area of the requesting node.

Figure 8 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 NEURO even in the worst case of  $p=0\%$  as shown in Figure 8, 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.



**Figure 8. Average path length of searches with different request structures.**

## 6. Conclusions

Due to the similarity of social networks and peer-to-peer networks, we believe and demonstrate that human strategies in social networks are useful for improving resource discovery by building a social P2P network. In this paper, we present a Social-P2P algorithm for resource discovery by mimicking human behaviours in social networks without additional overhead. In Social-P2P, a knowledge index is built according to the results of previous searches. Queries are routed intelligently even only with limited knowledge and connections. The small-world phenomenon has been observed in Social-P2P networks with a high clustering coefficient and a short average path length. Social-P2P system has been simulated in a dynamic environment with probabilistic request structure and file sharing. From the results and analysis, Social-P2P achieved better performance, more quickly found more matched files and more efficiently established a knowledge index about the location of files, than current methods.

## References

- [1] S. Milgram, “The Small World Problem,” *Psychology Today* 2, 1967, pp.60-67.
- [2] D. Watts and S. Strogatz, “Collective Dynamics of Small-World Networks,” *Nature* 393, 1998, pp.440-442.

- [3] I. Stoica, et al., "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," in *Proc. of ACM SIGCOMM*, San Diego, CA, 2001, pp.149-160.
- [4] S. Ratnasamy, et al., "A Scalable Content-Addressable Network," in *Proc. of the ACM SIGCOMM*, San Diego, CA, 2001, pp.161-172.
- [5] A. Rowstron and P. Druschel., "Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems," in *Proc. of IFIP/ACM International Conference on Distributed Systems Platforms*, Heidelberg, Germany, 2001.
- [6] N. Antonopoulos and J. Salter, "Efficient Resource Discovery in Grids and P2P Networks," *Internet Research, Vol 14, No 5*, 2004, pp.349-346.
- [7] P. Maymounkov and D. Mazieres, "Kademlia: A Peer to Peer Information System Based on the XOR Metric," in *Proc. of International Workshop on Peer-to-Peer Systems*, Cambridge, MA, 2002.
- [8] B. Yang and H. Garcia-Molina, "Efficient Search in Peer-to-Peer Networks," in *Proc. of International Conference on Distributed Computing Systems*, Vienna, Austria, 2002.
- [9] S. Rhea, D. Gells, T. Roscoe, and J. Kubiatowicz, "Handling Churn in a DHT," in *Proc. the USENIX Annual Technical Conference*, Boston, MA, 2004.
- [10] F.M. Cuenca-Acuna and T.D. Nguyen. "Text-based Content Search and Retrieval in ad hoc P2P Communities," in *Proc. of International Workshop on Peer-to-Peer Computing*, Pisa, Italy, 2002.
- [11] M.S. Khambatti, K.D. Ryu and P. Dasgupta, "Efficient Discovery of Implicitly Formed Peer-to-Peer Communities," *Journal of Parallel and Distributed Systems and Networks*, 5(4), 2002, pp.155-164.
- [12] M. Khambatti, K.D. Ryu and P. Dasgupta, "Structuring Peer-to-Peer Networks Using Interest-Based Communities," in *Proc. of International Workshop on Databases, Information Systems, and Peer-to-Peer Computing*, Toronto, Canada, 2004.
- [13] R. Karp, S. Shenker, C. Schindelhauer, and B. Vocking, "Randomized Rumour Spreading," in *Proc. of 41st Symposium Foundation on Computer Science*, Redondo Beach, CA, 2000.
- [14] B. Bloom. "Space/time Tradeoffs in Hash Coding with Allowable Errors," *CACM*, 13(7), 1970, pp.422-426.
- [15] A. Crespo and H. Garcia-Molina, "Routing Indices for Peer-to-Peer Systems," in *Proc. of the International Conference on Distributed Computing Systems*, Vienna, Austria, 2002.
- [16] JXTA. Available: <http://www.jxta.org>.
- [17] Bearshare. Available: <http://www.bearshare.com>.
- [18] L. Xiao, Y.Liu, and L.M. Ni, "Improving Unstructured Peer-to-Peer Systems by Adaptive Connection Establishment," *IEEE Transactions on Computers* 54, 2005, pp.176-184.
- [19] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutella-Like P2P System Scalable," in *Proc. of ACM SIGCOMM*, Karlsruhe, Germany, 2003.
- [20] D. Tsoumakos and N. Roussopoulos, "Adaptive Probabilistic Search for Peer-to-Peer Networks," in *Proc. of the International Conference on Peer-to-Peer Computing*, Linköping, Sweden, 2003.
- [21] S. Joseph, "NeuroGrid: Semantically Routing Queries in Peer-to-Peer Networks," in *Proc. of the International Workshop on Peer-to-Peer Computing*, Cambridge, MA, 2002.
- [22] C. Tempich, S. Staab, and A. Wraniak, "REMINDIN: Semantic Query Routing in Peer-to-Peer Networks Based on Social Metaphors," in *Proc. of 13<sup>th</sup> WWW Conference*, New York City, 2004.
- [23] SWAP, Available: <http://swap.semanticweb.org/>.
- [24] Jon Kleinberg, "Small-World Phenomena and the Dynamics of Information," *Advances in Neural Information Processing Systems (NIPS) 14*, 2001.
- [25] H. Zhang, A. Goel, R. Govindan, "Using the Small-World Model to Improve Freenet Performance," *Computer Networks* 46(4), 2004, pp.555-574.
- [26] L. Liu, S. Mackin, and N. Antonopoulos, "Small World Architecture for Peer-to-Peer Networks," in *Proc. of IEEE/WIC/ACM Workshop on Peer-to-Peer Networking and Autonomous Agents*, Hong Kong, 2006, in press.
- [27] The Open Directory Project. Available: <http://dmoz.org/>.
- [28] P. Makosiej, G. Sakaryan, and H. Unger, "Measurement Study of Shared Content and User Request Structure in Peer-to-Peer Gnutella Network," in *Proc. of the International Conference on Design, Analysis, and Simulation of Distributed Computing System*, 2004.
- [29] C. Pauli and M. Shepperd, "An Empirical Investigation into P2P File-Sharing User Behaviour," in *Proc. of Americas Conference on Information Systems*, Omaha, Nebraska, 2005.
- [30] A. Crespo and H. Garcia-Molina, "Semantic Overlay Network for P2P System," Technical Report, Stanford University, 2002.
- [31] R. Bhagwan, S. Savage, and G.M. Voelker, "Understanding Availability," in *Proc. of the International Workshop on Peer-to-Peer Computing System*, San Joes, CA, 2003.
- [32] S. Saroiu, "A Measurement Study of Peer-to-Peer File Sharing Systems," in *Proc. of Multimedia Computing and Networking*, San Jose, CA, 2002.
- [33] Y. Ren, et al., "Explore the Small World Phenomena in Pure P2P Information Sharing System," in *Proc. of the International Symposium on Cluster Computing and the Grid*, Tokyo, 2003.
- [34] P. Krishna, "Measurement, Modelling and Analysis of a P2P File-sharing Workload," in *Proc. of ACM Symposium on Operating Systems Principles*, New York, 2003.
- [35] H. Zhou, "Scaling Exponents and Clustering Coefficients of a Growing Random Network," *Physical Review E* 66, 016125, 2002.