Efficient Resource Discovery in Self-organized Unstructured Peer-to-Peer Networks

Lu Liu[1], Nick Antonopoulos[2], Stephen Mackin[3], Jie Xu[4], Duncan Russell[5]

Abstract:

In unstructured peer-to-peer (P2P) networks, two autonomous peer nodes can be connected if users in those nodes are interested in each other's data. Due to the similarity between P2P networks and social networks, where peer nodes can be regarded as people and connections can be regarded as relationships, social strategies are useful for improving the performance of resource discovery by self-organizing autonomous peers on unstructured P2P networks. In this paper, we present an Efficient Social-Like Peer-to-peer (ESLP) method for resource discovery by mimicking different human behaviours in social networks. ESLP have been simulated in a dynamic environment with a growing number of peer nodes. From the simulation results and analysis, ESLP achieved better performance than current methods.

Keywords: Peer-to-Peer, Social Networks, Small World, Search, Simulation

[1] School of Computing, University of Leeds, Leeds, West Yorkshire, United Kingdom
[2] Department of Computing, University of Surrey, Guildford, Surrey, United Kingdom
[3] Surrey Satellite Technology Limited, Surrey Research Park, Guildford, Surrey, United Kingdom
[4] School of Computing, University of Leeds, Leeds, West Yorkshire, United Kingdom
[5] School of Computing, University of Leeds, Leeds, West Yorkshire, United Kingdom

1. INTRODUCTION

Peer-to-peer (P2P) networks attract attentions worldwide with the great success in file sharing networks (such as Napster, Gnutella, Freenet, BitTorrent, Kazaa, and JXTA). As a major design pattern for future systems opposite to the traditional client-server paradigm, research on P2P networks is extremely important and could possibly radically alter the way of day-to-day use of computer systems.

Efficient resource discovery remains a fundamental problem for large-scale P2P networks. In contrast to P2P networks, people in social networks can directly contact some acquaintances that potentially have knowledge about the resources they are looking for. Similarly to social networks where people are connected by their social relationships, two autonomous peer nodes can be connected in unstructured P2P networks if users in those nodes are interested in each other's data. The similarity between P2P networks and social networks, where peer nodes can be regarded as people and connections can be regarded as relationships, leads us to believe that human strategies in social networks are useful for improving the performance of resource discovery by self-organising autonomous peer nodes on unstructured P2P networks.

The theories of small world networks [1-2] in social sciences have been already applied to the system design of P2P overlay networks [3-9]. However, most studies of constructing small world behaviours on a P2P topology are based on the concept of clustering peer nodes consciously into groups, communities, or clusters (e.g. [4-6]). Studies like [7-9] have explored the possibility of building an information sharing system by clustering peer nodes into different groups according to their interests. However, maintaining an additional multilayer structure over dynamic P2P networks will also introduce extra overhead [9]. 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 [10]) and compact data structures (e.g. Bloom Filters [11]).

To address this problem, an Efficient Social-Like Peer-to-peer (ESLP) model for resource discovery is presented in this paper by mimicking different human behaviours in social networks. Once a peer node finds a files it is looking for, the peer node can download it, save it or discard it, which will not be discussed in this paper.

The ESLP is based on our preliminary work on Social-P2P [12], which can be deployed on top of any unstructured P2P networks to improve the performance of resource discovery. Unlike previous models, we do not intentionally construct peer communities. In contrast, ESLP gives peer nodes a social network and let them meet and get to know each other. Peer nodes that have the same interests will gradually connect to each other and form peer communities spontaneously, which can maximally avoid problems of previous community-based P2P systems.

The remainder of this paper is organized as follows: Section 2 discusses related work. Section 3 presents the ESLP algorithm. Section 4 describes our simulation methodology. Simulation results are analysed in Section 5 and we conclude the work in Section 6.

2. RELATED WORK

2.1. Unstructured P2P Search

Existing solutions for resource discovery in P2P systems can be classified into two categories: structured and unstructured P2P systems. Each structured P2P system (e.g. Chord [13], CAN [14], and Pastry [15]) has a dedicated network structure on the overlay network. In contrast, unstructured P2P systems (e.g. Gnutella) do not control data placement and are resilient in dynamic environments. 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. [16]). Peer nodes normally maintain additional indices of files offered by their 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 by reorganising peer nodes into a two-level hierarchy with supernodes and leaves. Supernodes 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 the overlay topology (e.g. [17-18]). In ESLP, 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, Different from self-organizing networks, the search algorithms of Adaptive Probabilistic Search (APS) [19] 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.

NeuroGrid [20] is an adaptive decentralised search system. Unlike previous methods, NeuroGrid utilizes the historic record of previous searches to help peer nodes make routing decisions. In NeuroGrid, peer nodes support distributed searches through semantic routing by maintaining routing tables at each node [21]. 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 the query to peer nodes from the rest of the connected nodes. However, NeuroGrid is effective for previously queried keywords only and is not suitable for networks where peer nodes come and go rapidly [21].

Sripanidkulchai [22] presented a content location solution in which peer nodes loosely organise themselves into interest-based structure. When a peer node joins the system, it first searches the network by flooding to locate content. The lookup returns a set of peer nodes that store the content. These peer nodes are potential candidates to be added to a "shortcut list". One peer node is selected at random from the set and added. Subsequent queries will go through the peer nodes in the shortcut list. If a peer cannot find content from the list, it will generate a lookup with Gnutella protocol.

However, similarly to NeuroGrid, Sripanidkulchai's model is only effective for previously generated queries.

Semantic Overlay Network (SON) is presented in [23], where queries are routed to the appropriate SONs, increasing the performance of recourse discovery in P2P networks. Peer nodes are classified strictly into different SONs by node classifier. Formation and maintenance of SONs are very expensive in distributed P2P networks, which still inherent the same problem of the community-based P2P systems [4-6] discussed in Section 1. Similarly to SON, associative overlay [24] is used to organise the peer node with different guide rules. Search is conducted through guide rules. However, a search within each guide-rule is essentially blind, without a further selection algorithm for the nodes in the same area of interest.

2.2. Small World Networks

TSN [25] 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 [26] 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. ESLP: Efficient Social-Like Peer-to-peer

In this section, we will reify the social theories that can be used in the resource discovery in P2P systems into six social behaviours and then mimick these social behaviours in a P2P network.

3.1. Knowledge Index Creation

Social behaviour 1: in social networks, people remember and update potentially useful knowledge from social interactions, and then random and diffuse behaviours gradually become highly organised [27].

Similarly to people in social networks, each peer node in the ESLP network builds a knowledge index that stores associations between topics and the related peer nodes by the results of searches. As illustrated in Figure 1, when a peer node receives a query, it will first search the local content index to find matched files. If the query needs to be further forwarded, the peer node will use the local knowledge index to find associated peer nodes using the ESLP routing algorithm and multicast the query to these peer nodes.

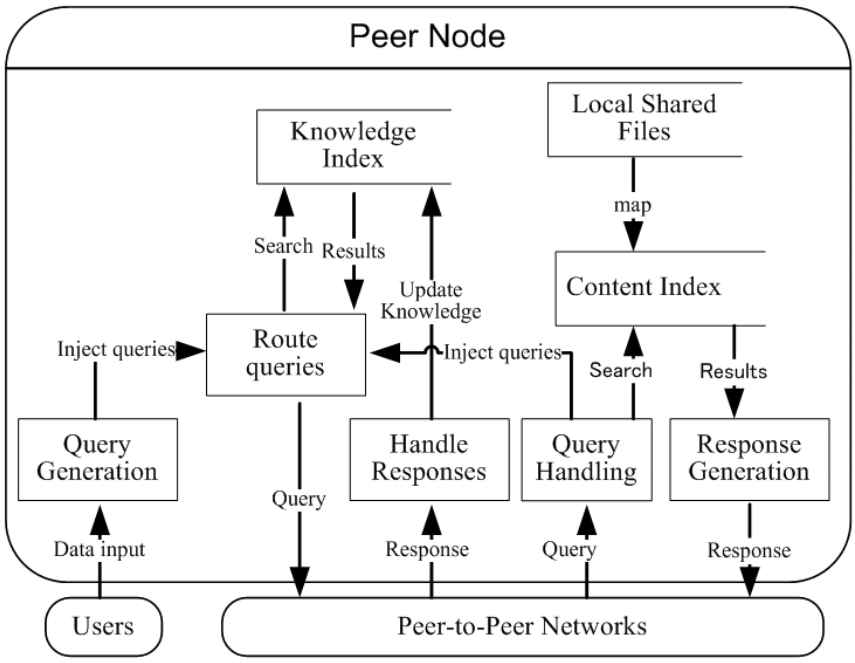


Figure 1. Query initialisation and query processing.

ESLP uses a TTL (Time-to-Live) to prevent infinite propagation. In the ESLP network, a newly joined node will send its first query to a set of randomly selected 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 connects to these nodes. The new obtained knowledge is stored in the local knowledge index. 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.

Social behaviour 2: in social networks, some events with associated people fade from a person's memory with time and a personal network is adjustable with changing environments.

Similarly, in ESLP, peer nodes can update their knowledge on other peer nodes from daily search results. Some old and invalid knowledge is replaced by new obtained knowledge. The invalid knowledge that fails to be updated with daily searches will be dropped to the bottom of the knowledge index. The knowledge index is maintained in a queue using a Least Recently Used (LRU) policy without duplicates, where the most-recently used topic at the top and the least-recently used at the bottom. The advantages of LRU for removing old index items is that it has constant time and space overhead and can be very efficiently implemented. Since the size of the knowledge index of ESLP is finite at each peer node, such oldest knowledge (probably out-of-date) will be removed when the knowledge index reaches a maximum.

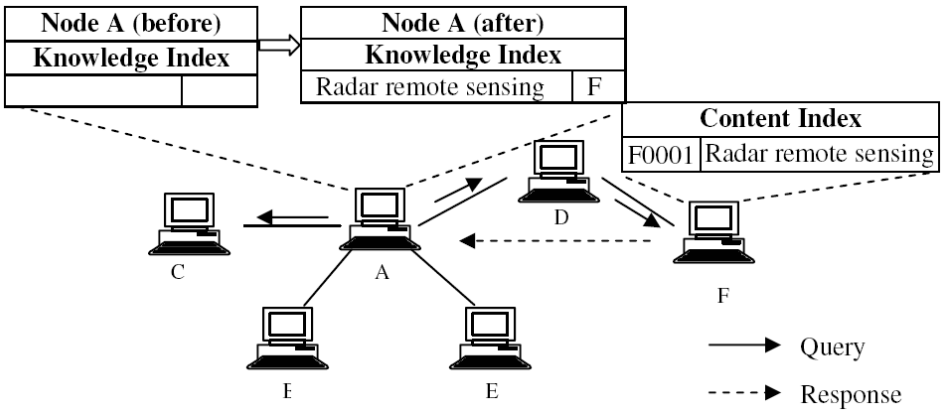


Figure 2. A knowledge collection example.

Figure 2 shows an example of knowledge collection process of a peer node. Node *A* with an empty knowledge index searches for the files on the topic "radar remote sensing". Since no information is cached, node A will send the query to the randomly selected nodes: node *C* and node *D*. Node *D* further forwards the query to its neighbouring node *F*. In this case, the query is successful at node *F*. Thus, node *F* responds to node *A* with the requested topic. Node *A* then associates node *F* with the topic "radar remote sensing" in its knowledge index. For a following query on "radar remote sensing", node *A* can find the node *F* directly associated with the query from the local knowledge index and will preferentially send the query to node *F* rather than node *C* and node *D*.

3.2. Query Generation

In order to efficiently search the network, two kinds of queries will be generated according to the different stages of searches known as ordinary query and active query. For the ordinary query, the target nodes sharing the desired files will respond with the information related to the requested topic only. For the active query, the target nodes will not only respond to the requested topic but also inform the originator of other associated topics it shares in the same interest area. An interest area in ESLP 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 [28], which is the most widely distributed data base of Web content with a common topic structure. The Open Directory Categories have been widely used in popular Internet services, such as Google, Yahoo, ICQ, AOL, MSN.

As illustrated in Figure 3, when the originator generates a query with the topic "radar remote sensing", a target node that shares the desired files will answer the query about "radar remote sensing" as well as the associated topics "optical remote sensing", "laser remote sensing", and "visual remote sensing" in this interest area. The obtained new information will be put into the local knowledge index by the originator for future queries. With these active queries, the originator can gather more pieces of knowledge from each successful query, but additional traffic will be generated for shipping such additional knowledge. The extra traffic could be significant if every node generates all queries in this manner, which is difficult to be handled for bandwidth-limited networks. In contrast, if we search the network only with ordinary queries, each new node accumulates knowledge slowly by gathering one piece of knowledge from each successful query, especially for those peer nodes which are seldom present or query the network.

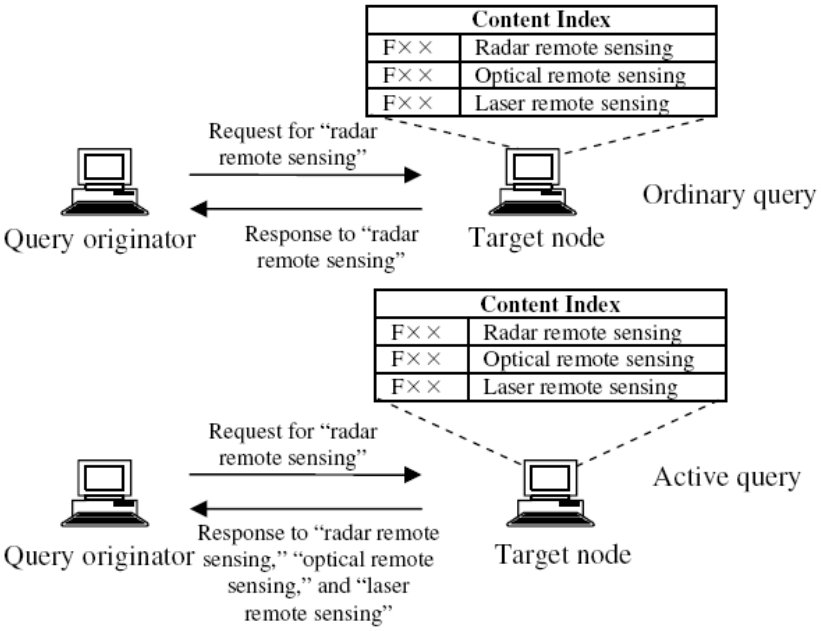


Figure 3. Ordinary query and active query of ESLP.

To address these issues, ESLP utilizes a trade-off solution for the bandwidth-limited networks. The recently joined nodes will utilize active queries to quickly accumulate a large amount of useful information regarding their interests. After the cached knowledge reaches a certain threshold ratio r of the maximum size of the knowledge index $n_{\max\_knowledge}$, the peer nodes will utilize ordinary queries to discover the required files with low traffic cost. If the ratio of cached knowledge $r_{cached} \le r$, active queries will be used to search the network. Otherwise, an ordinary query will be adopted. Moreover, this process is not only applicable for recently joined nodes, but also enables peer nodes to quickly recover from unpredictable knowledge loss.

3.3. Semantic Routing Algorithm

3.3.1 Query Processing

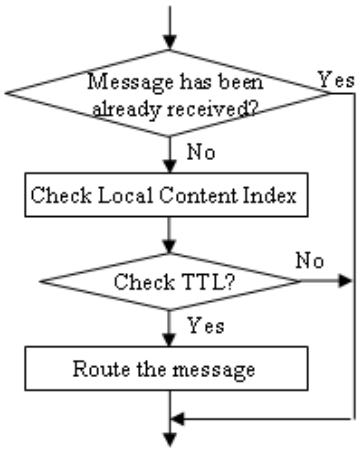


Figure 4. Flowchart of query handling.

When a peer node receives a query, it will first check whether the message has been already received as shown in Figure 4. Redundant messages will be discarded

without further processing. Then the peer node will search the local content index to find matched files. If the query needs to be further forwarded ($TTL > 0$), the message receiver will use the local knowledge index to find associated peer nodes from the local knowledge index using the ESLP routing algorithm and multicast the query to these peer nodes as shown in Figure 2.

Social behaviour 3: in a social network, communities are self-organised with regard to the common interests. In the ESLP network, because links between peer nodes are built according to the results of searches, a node has more probability to link to other peer nodes with the same interests, which have files of interest to him/her, with a high degree of likelihood. Therefore, 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 [2] in social networks.

3.3.2 Adaptive Forwarding

Analogously to social networks, ESLP utilises a semantic approach to route queries to a selected subset of neighbouring nodes on the P2P overlay. In addition, ESLP also involves a dedicated strategy to address the network overload problem of some existing P2P systems (e.g. Gnutella).

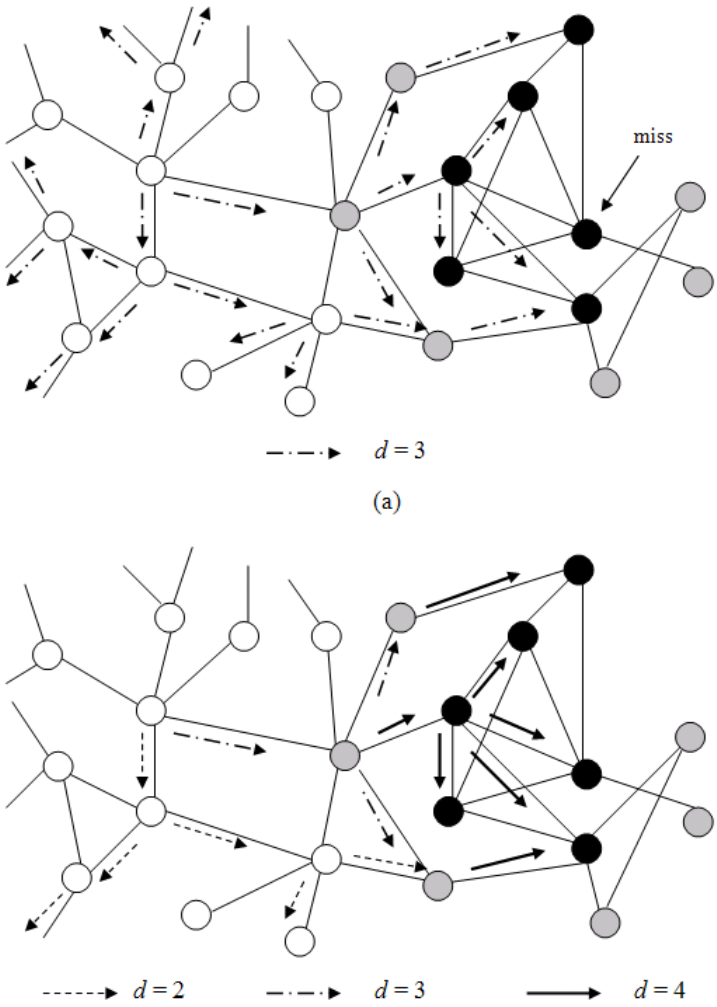


Figure 5: Comparison of static routing and adaptive routing (a) a static routing strategy (b) an adaptive routing strategy

In existing query routing methods (e.g. [21, 29, 30]), the number of peer nodes to be forwarded in each hop is usually set to a static value. A fixed number of peer nodes are selected in each hop neglecting the probability of finding the requested file in these peer nodes. In order to conduct a more efficient search in ESLP, the number d of peer nodes to be forwarded is adjustable according to the correlation degree of the selected node to the query between a minimal number $D_{\min}$ (lower bound) and a maximum number $D_{\max}$ (upper bound) in each hop. When the correlation degree of a selected peer node is high, there is a high possibility that the selected peer node may share a desired file or has the knowledge about who shares the file. The probability of forwarding the query to this peer node should also be high by defining a high cut-off of node selection. In contrast, if the correlation degree is low, a low cut-off should be set to limit the scope of querying.

Figure 5 shows examples of two routing strategies with a static number of receivers per hop ($d = 3$) and an adaptive number of receivers per hop ($d = 2 \sim 4$), respectively. In Figure 5, the black dots represent the peer nodes that share the requested resources, while the grey dots show the peer nodes that are highly correlated with the query and know who has the requested resources. As shown in Figure 5, the adaptive routing strategy achieves better search performance by finding more target nodes with fewer query messages.

3.3.3 Node Selection

Social behaviour 4: 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 has found that 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.

The node selection procedure of ESLP is shown in Figure 6, where n is the number of peer nodes that have already been selected to be forwarded in each hop.

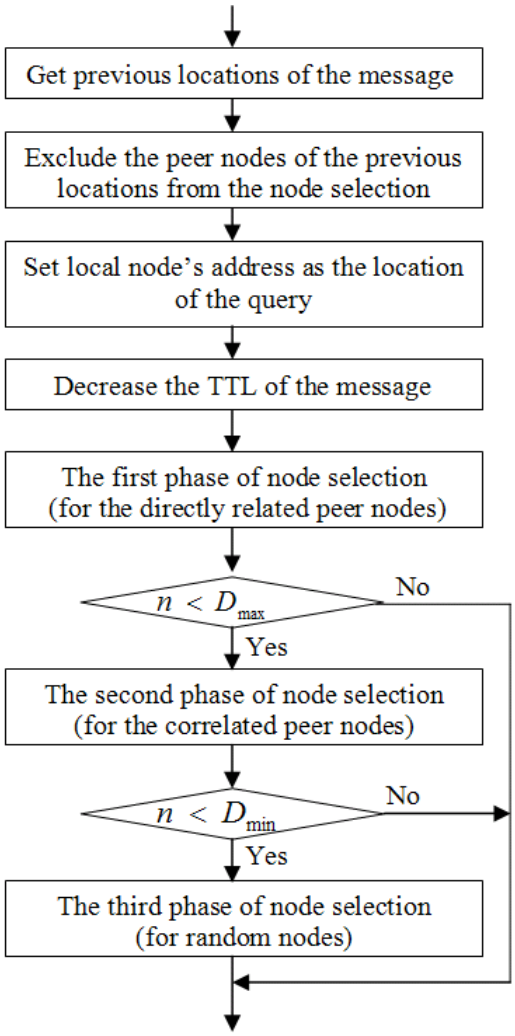


Figure 6. Flowchart of the node selection procedure

The routing algorithm of ESLP involves the following three phases in each hop: searching for the directly related peer nodes, searching for the correlated peer nodes and searching for random peer nodes, from the local knowledge index. When a node receives a query which needs to be forwarded, the node routing algorithm firstly searches for peer nodes directly associated with the requested topic from the local knowledge index and sorts them with the time of receipt. The peer node that is inputted or updated most recently will be selected first. These directly associated peer nodes have the greatest likelihood of finding the requested files. Hence, at most $D_{\max}$ peer nodes will be forwarded.

However, the success probability of finding $D_{\max}$ 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, which searches for the peer nodes sharing content associated with the interest area of the requested topic from the local knowledge index. The corresponding interest area of a specific topic and the other topics in this interest area can be found from the Open Directory Categories. ESLP 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. For a given query, such as “Apple”, Open Directory will return several options, e.g., 1) Computers: Systems: Apple, 2) Home: Cooking: Fruits and Vegetables: Apples, 3) Computers: Companies: Apple Inc.. Users need to select one option to continue the search. If Open Directory cannot provide any satisfactory category for the query, users can also define a category for their own query.

These peer nodes sharing 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 correlation. If two or more nodes have the same correlation degree, we put the peer node that responded most recently first.

Social behaviour 5: searching for a piece of information in social networks is most likely a matter of searching for an expert on the topic together with a chain of personal referrals from the searcher to the expert [31].

Social behaviour 6: in social networks, 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.

In the ESLP network, it is not necessary for a peer node to declare its interest since that has already been implied by its shared files. 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 knowledge or other content 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: $c = n_{matches} / n_{total}$, 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 which can be found in the Open Directory. Different from any other P2P algorithm, the cut-off criteria d for the second phase are different for different peer nodes between $D_{\max}$ and $D_{\min}$. The cut-off criterion d for a peer node 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:

$$d = round\left(c \cdot \left(D_{\max} - D_{\min}\right)\right) + D_{\min}, \quad (1)$$

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 peer node should be low with a low cut-off ($d \approx D_{\min}$). In contrast, when the peer node is highly correlated with the area of the requested topic by matching most topics in this area ($c \approx 1$), the cut-off of the peer node $d \approx D_{\max}$.

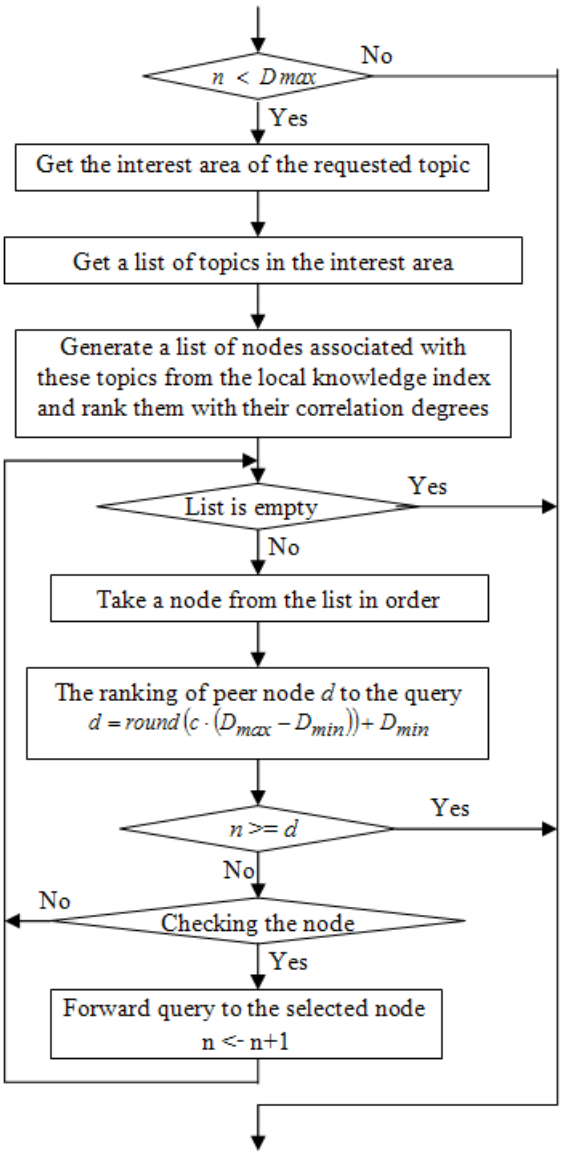


Figure 7. The second phase of semantic routing algorithm.

The flowchart in Figure 7 shows the second-phase of the node routing algorithm used in our simulations. As shown in Figure 7, the peer nodes associated with the interest area of the requested topic are sorted with their correlation degree. Because peer nodes are taken from the list in order, the cut-off d will decrease with the reducing correlation degrees c of the selected peer nodes (according to Equation 1). But n is increased by one for each new node selected. The query will be sent to the peer nodes only if the number of forwarded nodes n is smaller than this peer's cut-off to the query 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 ($c > 0$) have been taken from the list in the second phase but there are still not enough nodes selected $n < D_{\min}$, 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 the associated area ($c = 0$) and forward the query to them, until $D_{\min}$ peer nodes are forwarded ($n = D_{\min}$).

3.4. Query Routing Example

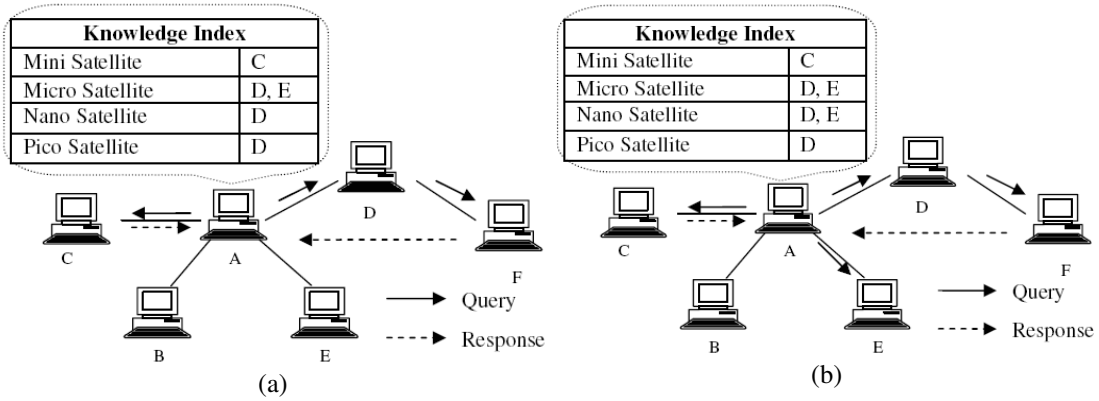


Figure 8. Examples of query routing. (a) Two nodes are queried. (b) Three nodes are queried.

Figure 8(a) illustrates a simple example of query routing with the ESLP algorithm where $D_{\max} = 5$ and $D_{\min} = 1$. Suppose node *A* receives a query with the topic "mini satellite". In the first phase, node *A* finds the node *C* which is directly associated with the topic "mini satellite". Due to $n < D_{\max}$ ($n = 1$, $D_{\max} = 5$), node *A* will further search the local knowledge index for the peer nodes associated with the relevant topics "micro satellite", "nano satellite", and "pico satellite" from the same interest area of the requested topic. In this case, node *A* gets node *D* and node *E* associated with these topics from the local knowledge index. Since node *D* is associated with three topics and node *E* is associated with one topic in the area with four topics, node *D* ($c_D = 3/4$, $d_D = \frac{3}{4} \cdot (5-1) + 1 = 4$) is more correlated with the interest area of the requested topic in accordance with the cached knowledge than node *E* ($c_E = 1/4$, $d_E = \frac{1}{4} \cdot (5-1) + 1 = 2$). Hence, node *D* will be sorted on top of node *E* in the list and the query will be sent to node *D*, because the number of forwarded nodes is smaller than the cut-off of node *D* $n < d_D$ ($n = 1$, $d_D = 4$). Then $n + 1 \rightarrow n = 2$ and the selection procedure will be completed because $n \geq d_E$ ($n = 2$, $d_E = 2$). The actual number of queried nodes in this case is *two*. Node *D* may not have the requested files, but it will use its own cached knowledge to propagate the query further and to find peer nodes sharing the desired files with a higher likelihood. In this example, node *D* knows that node *F* is associated with the requested topic and the requested files are obtained in node *F*.

Figure 8(b) illustrates a similar example, where node *E* is associated with two topics "micro satellite" and "nano satellite" in the same area of the requested topic "mini satellite". node *E* is also selected in this case, because $n < d_E$ (where $n = 2$ and $d_E = \frac{2}{4} \cdot (5-1) + 1 = 3$)) and the actual number of nodes to be queried is *three* in the second example.

4. SIMULATION METHODOLOGY

We evaluate the performance of ESLP by simulations under a dynamic environment. The ESLP simulator is programmed using the Java Language. The main components are illustrated in Figure 9.

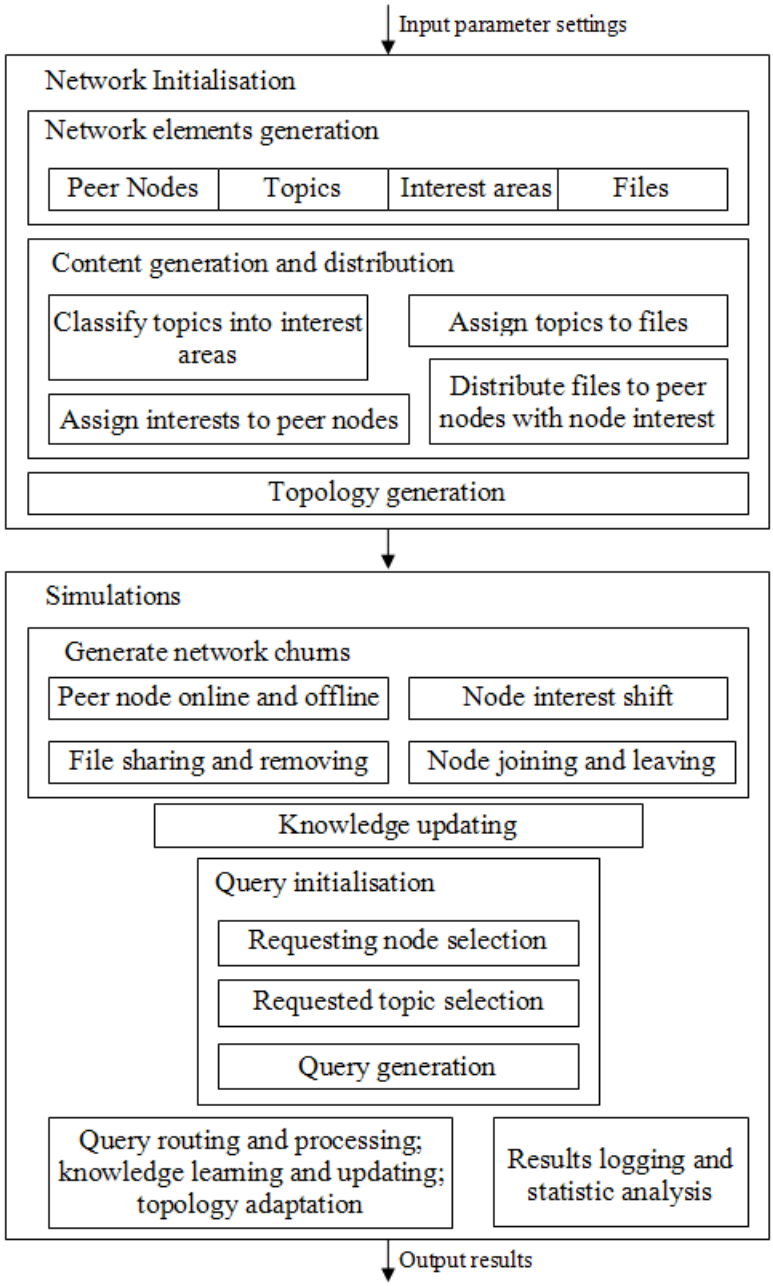


Figure 9. Simulator structure.

4.1. 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. In order to build a near-realistic environment, we preferred to use the data from the observed networks. The previous studies observed that the distribution of keywords in files could be approximated by Zipf's law in the form of

$y \sim \frac{1}{x^{\alpha}}$, where y is frequency, x is rank and α is constant. The estimated distribution in the study [33] has been followed in our simulations to generate topic keyword distribution to files. In each simulation, we generated 1280 topics, distributed them to 10,000 files, and each file was assigned two 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].

The measurement study [23] for the music sharing network on Stanford shows 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 randomly 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 a node with a probability of $P_f = 90\%$, but occasionally from a random area $(1 - P_f) = 10\%$. For the file relevant to the primary interest area, at least one of the topics of the file should be in the interest area of the hosting node. A total of 16 interest areas were generated and each covered 80 topics.

4.2. Search Network

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 current primary interest area of the requesting node with a probability P_s ($P_s = 90\%$, if no other value is specified), but sometimes from a random area with a probability of $(1 - P_s)$. Each query was tagged by a *TTL* to limit the life time of a message to three hops in the simulations. The generated queries were propagated with the ESLP routing algorithm. The performance metrics discussed in Section 4.5 were recorded and statistically analysed. Even though request frequency is variable for different users in different periods, the study [32] observes that each peer node generates an average of two requests each day. This has been implemented in our simulations. Each simulation day is set as 2000 time steps.

4.3. Topology Initialization and Evolution

The studies in [30, 24] 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 characteristics 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 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 in the beginning of each simulation, each peer node kept an empty knowledge index which can contain a

maximum of 1000 entries between topics and associated peer nodes (if no other size is specified). The threshold ratio r of ESLP is defined as 80%.

Some popular P2P networks are growing very fast on the Internet [35]. However, some measurement studies (e.g. [36]) observed that the size of some mature P2P networks stay 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 (30 nodes) joined the network every day in the first month until the network 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 as described in the next paragraph. We ran simulations to trace the results of about two months. Five independent simulation runs are performed in each experiment.

4.4. Network Churn Generation

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 [36] measured network churns by using a user ID instead of an IP address that was used in previous measurement studies (e.g. [34]). 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). Therefore, our simulations followed the availability distribution of peer nodes in the study [36], where about 50% of peer nodes are presented less than 30%.

The research in [37] argued that user interest shift is a vital factor for P2P file-sharing networks, especially in today's dynamic information era. The semantic areas of interests may be stable, but interests of peer nodes are dynamic. 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. 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 share an extra 10% files to the network and 1% of peer nodes to remove 10% of shared files 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 peer nodes could have already been removed from the network.

4.5. Performance Metrics

Performance was evaluated with the following measures:

- Recall: the ratio of the number of found files to the number of all matched files in the network.
- Average path length of searches: the average distance from the query originator to the targeted node which first finds a matched file. If none are found, the average path length of the search is set as four ($TTL+1$) in the simulations. This metric is used to measure the speed of resource discovery.
- Recall per visited node.
- Average number of found files.
- Number of visited nodes.

- Number of query messages.

5. SIMULATION RESULTS

We compared the performance of ESLP with two relevant methods: RAN and NEURO, and two derived methods: OSLP and ASLP:

- OSLP: search the network with ordinary queries only. OSLP is a special type of ESLP with the threshold ratio $r = 0\%$.
- ASLP: search the network with active queries only. ASLP is a special type of ESLP with the threshold ratio $r = 100\%$.
- RAN: a constrained Gnutella-like routing strategy. Received queries are randomly passed to D_{min} connected peer nodes.
- NEURO: NeuroGrid [20] routing strategy.

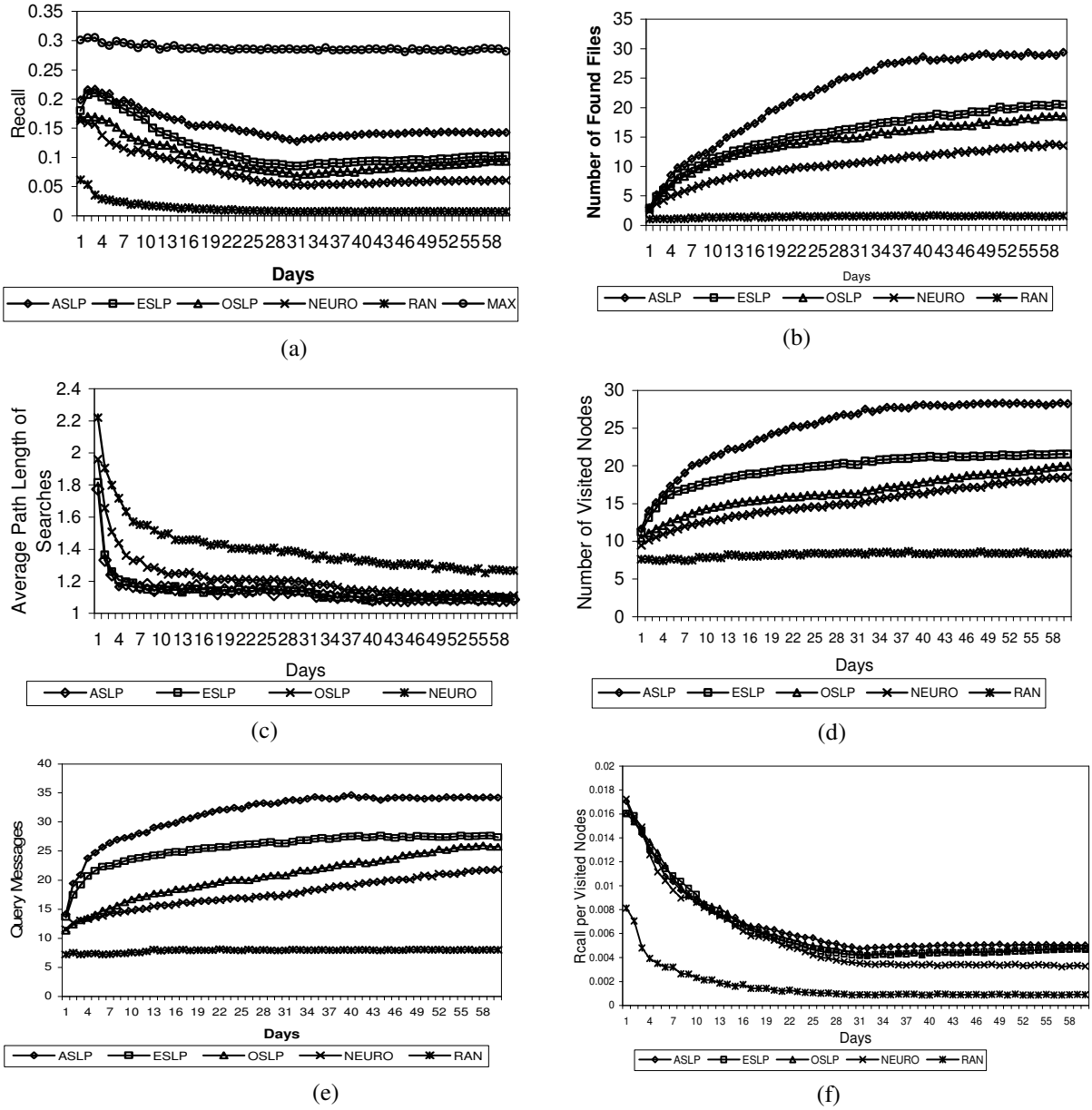


Figure 10. Performance comparison. (a) Recall. (b) Number of found files. (c) Average path length of searches. (d) Number of visited nodes. (e) Number of query Messages. (f) Recall per visited node.

5.1. Performance Evaluation

From the results in Figure 10, OSLP achieved better performances than NEURO and RAN by finding more files (as shown in Figure 10(a) and (b)) more quickly (as shown in Figure 10(c)). In Figure 10(a), the maximum possible recalls are all below 31%, because a large amount of files are available on a large number of offline nodes. Since many new files are added into the network by newly joined peer nodes, the recall decreases during the network growing period, while the number of found files keeps increasing. Because ASLP and ESLP can quickly accumulate a large amount of useful knowledge about the location of the relevant files in a short term, the average path length of searches quickly decreases to just above one, as shown in Figure 10(c).

As shown in Figure 10(d) and (e), OSLP will visit more peer nodes and need a few more query messages introduced by the second step of node selection procedure, but the search efficiency of OSLP is still better than NEURO by achieving higher recalls per visited peer node as shown in Figure 10(f). At the early stage of searches, it is very difficult for peer nodes to find directly associated nodes with the requested topic from the local knowledge index by using either OSLP or NEURO methods due to the limited knowledge cached, but OSLP are capable of retrieving the peer nodes which share associated files with the relevant topics more often. These selected peer nodes which are highly correlated with the interest area of the requested topic have more knowledge about the query than random nodes. Therefore, OSLP 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, OSLP have better search capabilities and better knowledge-collecting capabilities. With these advantages, OSLP achieved better performance than the other methods.

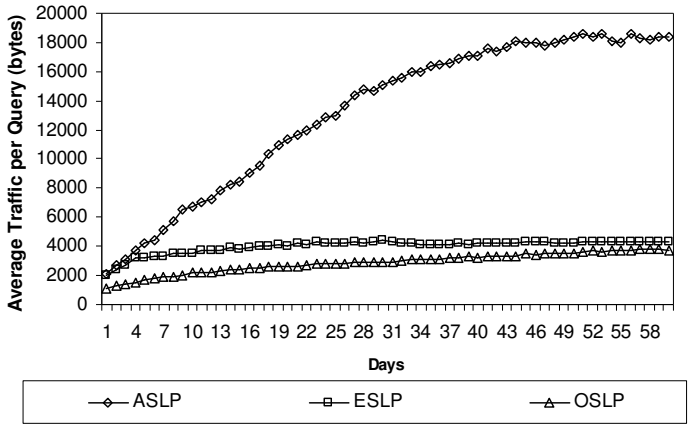


Figure 11. Average traffic generated per query.

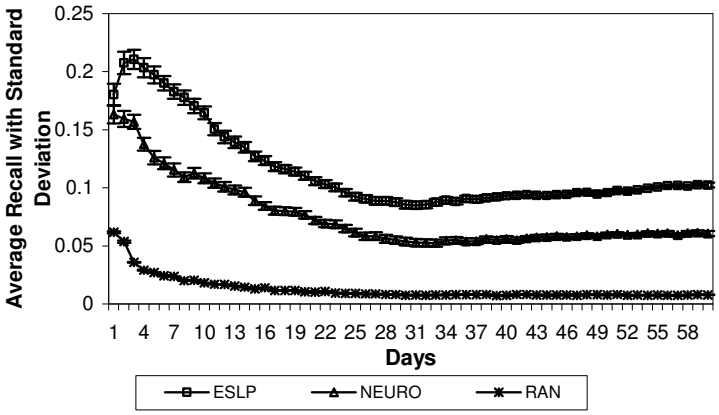
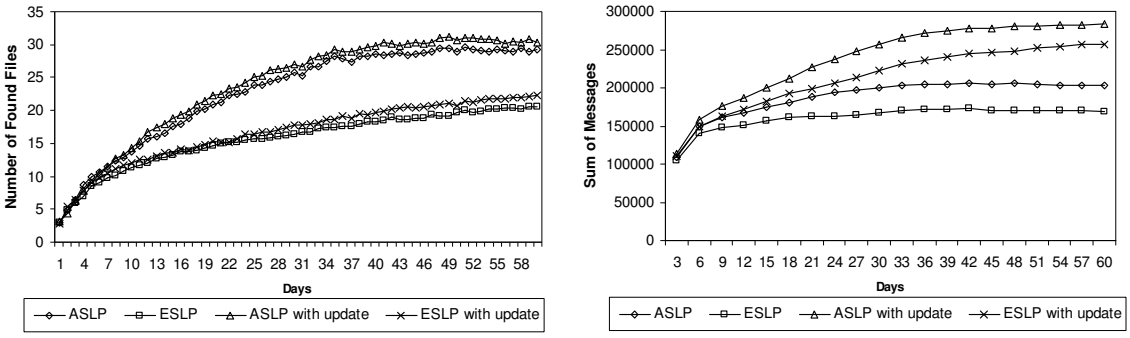


Figure 12. Recall with standard deviation.

With active social behaviours, ASLP and ESLP can more quickly establish a knowledge index than OSLP by gathering more pieces of knowledge from each successful query. Threshold r can be set according to the network bandwidth. ASLP is a special type of ESLP with the threshold ratio $r = 100\%$. As shown in Figure 10(d), ASLP (with $r = 100\%$) is recommended for the network with abundant bandwidth, because ASLP achieved the best performance by finding more requested files as shown in Figure 10(d).

Figure 11 shows the average traffic generated per query, where we obey the structure of Gnutella query and response messages and the length of a topic is set as 50 characters. As shown in Figure 11, the traffic generated by each ASLP query is increasing to a significant value by visiting more peer nodes and transferring more information, which can be a heavy traffic load for the network when many queries occur in the network at the same time. By defining $r = 80\%$, the performance of ESLP has been clearly improved with little more traffic compared to OSLP ($r = 0\%$), especially in the early stage of searches. Therefore, ESLP with threshold $r = 80\%$ is a good trade-off solution for the bandwidth-limited networks which can achieve a good performance at a small traffic cost. Uncertainties in the simulation data of ESLP, NEURO and RAN are measured by standard deviations, which have been shown in Figure 12. Since many peer nodes are joining the network at the first month, the uncertainties of data in the growing network are more significant than that in the mature network.

5.2. Knowledge Updating



(a) (b)

Figure 13. (a) Number of found files with ASLP with and without extra knowledge update. (b) Total of messages generated in every 3 days.

The validity of cached knowledge could be a potential problem due to network churns. Change handling in distributed system is a difficult task though it is often necessary [38]. In this section, an extra periodical knowledge update strategy is applied to the system to address this issue. In the simulations, each peer node periodically updated its local knowledge index to remove invalid knowledge every 3 days by querying all cached nodes in the local knowledge index. As shown in Figure 13(a), the performance of ASLP and ESLP was only improved a little by using the extra knowledge update due to their own self-update capability.

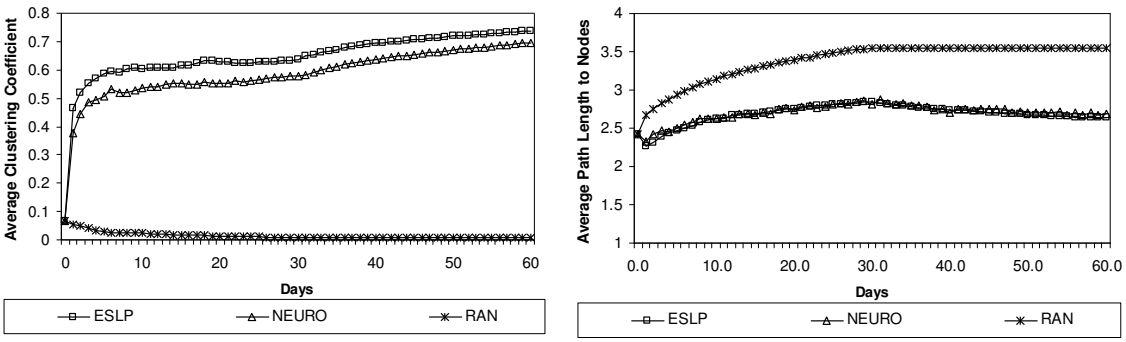
As shown in Figure 13(b), much more network traffic was generated by these knowledge updating messages. With ASLP and ESLP, peer nodes can update their knowledge about other peer nodes from daily search results. Some old and invalid knowledge is replaced by new obtained knowledge. Moreover, some invalid knowledge that fails to be updated will be dropped to the bottom of the knowledge index. With the LRU strategy, such knowledge will be removed when the knowledge

index reaches a maximum. Since the extra knowledge update generates significant traffic with little performance improvement as shown in Figure 13, there is no need for ASLP and ESLP to cooperate with an extra knowledge update in this case.

5.3. Topology Evolution

In Watt'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. These two properties of small world networks were recorded to monitor topology evolution in the simulations:

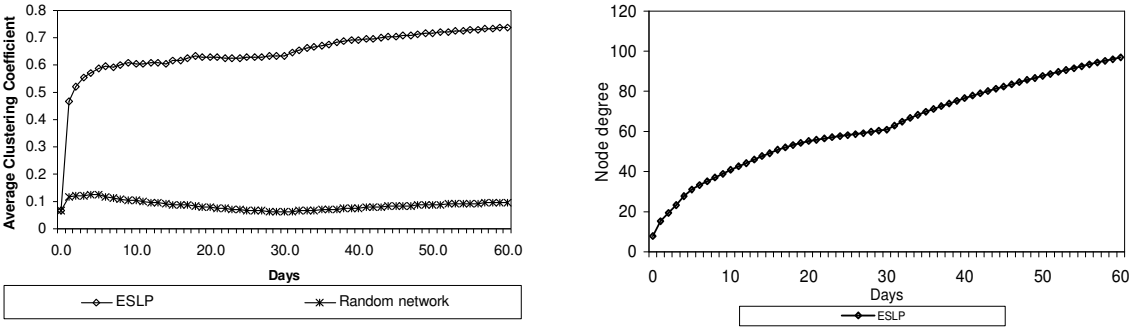
- Average path length to nodes: the average of the shortest distances between any two peer nodes in the simulation network.
- Average clustering coefficient: the average of the clustering coefficients of all nodes in the simulation network. The clustering coefficient of a node is the proportion of the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



(a) (b)

Figure 14. Topology Evolution. (a) Average clustering coefficient. (b) Average path length.

Figure 14(a) and (b) show the clustering coefficient and the average path length to nodes observed in the simulations. As shown in Figure 14(a), the clustering coefficients of ESLP are greater than other routing methods. Figure 15(a) shows the clustering coefficients of ESLP to those of randomly connected networks with the same number of nodes and connections. The clustering coefficients of the randomly connected networks, with the same number of nodes and connections as the ESLP simulation networks, are given by the equation: [39] $C \approx \langle k \rangle / N$, where $\langle k \rangle$ is the average node degree and N is the total number of nodes in the network. As shown in Figure 15(a), the clustering coefficients of ESLP are much greater than those of the randomly connected networks with the same number of nodes and connections.



(a) (b)

Figure 15. (a) Clustering coefficient comparison between ESLP and a random network with the same number of nodes and connections. (b) Evolution of average node degree.

Even though the network size increases quickly at the early stage of the simulations, the average path length to nodes of ESLP and NEURO only increases a little (Figure 14(b)) due to the increasing connectivity of the network (Figure 15(b)). The average path lengths to nodes of ESLP are only slightly smaller than those of NEURO method. However, by using different routing strategies, their search performances are clearly different as shown in Figure 10. From the results shown in Figure 14 and 15, we can see that the small-world phenomenon also appears in the ESLP with a high clustering coefficient of nodes and a short average path length to other peer nodes.

5.4. Knowledge Size

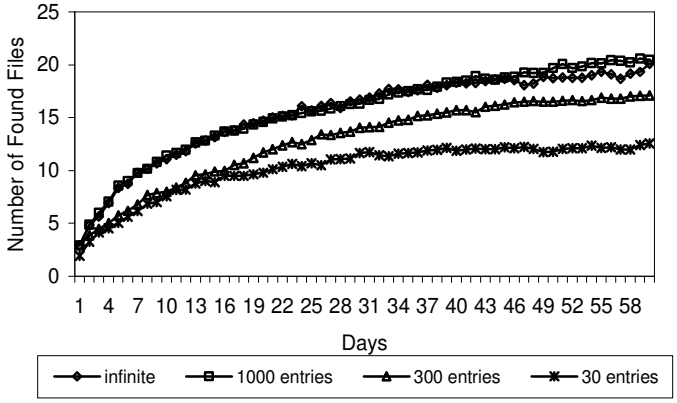


Figure 16. Number of found file with different sizes of knowledge index.

Size of lists in the knowledge index is an important parameter, which determines the storage overhead required for peer nodes. ESLP allows peer nodes to set individual maximum sizes of lists with regard to their capability. However, in order to evaluate the effect of size of knowledge index in the simulations, we define a uniform size in each experiment and then compare the search performance of ESLP with different sizes.

Figure 16 shows the number of found files by ESLP where each node has a knowledge index containing a maximum of 30, 300, or 1000 entries (pairs) between topics and associated peer node addresses. Clearly in Figure 16, the requesting nodes have more difficulty finding the requested files from the nodes with less knowledge. We also compared the search performance by using the knowledge index with a

maximum of 1000 entries and the infinite-size knowledge index (with the same adaptive threshold $r \cdot n_{\max\_knowledge} = 800$ entries). Surprisingly, the search performance by using the knowledge index with a maximum of 1000 entries achieved a slightly better performance than that with the infinite size of knowledge index at the end of the experiment, because the peer nodes with the infinite-size knowledge index could have difficulty keeping all cached knowledge up-to-date with its own self-update in a highly dynamic environment. Hence we defined the knowledge index with a maximum of 1000 entries for other experiments, which contains most of the commonly used topics that can be kept generally up-to-date.

5.5. Request Structure and Data Sharing

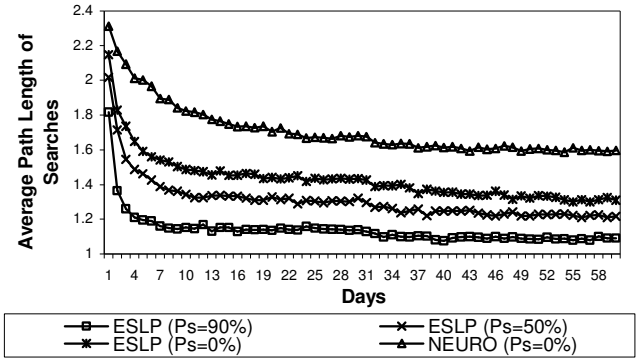


Figure 17. Average path length with different request structures.

ESLP is simulated with different request structures. By using the ESLP simulator, the requested topic is selected from the primary interest area of the query originator with a probability P_s, but is from a random area with a probability $(1-P_s)$. In the case of $P_s = 90\%$, 90% of the requested topics are randomly selected from the interest area of the query originator. On the contrary, in the case of $P_s = 0\%$, a purely random topic is chosen as the requested topic which is the worst case since the message holder cannot benefit from the repeated queries in its interest area.

Figure 17 shows the results of the average path length of searches by ESLP on some representative samples of P_s of 0%, 50% and 90%, respectively. In the simulations, the request scope was enlarged by setting a smaller P_s. Since the probability of matching cached knowledge decreases with P_s, the average path length of each search increases along with P_s. But the performance of ESLP is still better than that of NEURO even in the worst case of $P_s = 0\%$ as shown in Figure 17, because ESLP can still find the peer nodes that are more likely to share the request files even though the directly associated peer nodes cannot be found from the local knowledge index.

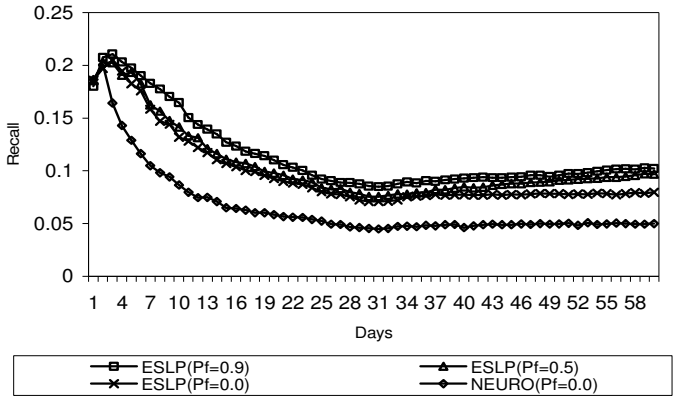


Figure 18. Recall with different structures of data sharing.

The effect of the simulation parameter P_f is also evaluated in the simulations. As shown in Figure 18, the recall decreases by setting a smaller P_f. However, the performance of ESLP is also better than that of NEURO even in the case of $P_f = 0\%$, since ESLP is able to find the peer nodes that potentially have the knowledge about queries. According to the results shown in Figure 17 and 18, peer nodes are generally more difficult to target the requested files in the network where users have very wide interests.

6. CONCLUSIONS AND FUTURE WROK

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 peer-to-peer resource discovery by building a social-like peer-to-peer network. In this paper, we present ESLP for resource discovery by self-organizing autonomous peers with social strategies. ESLP is a self-adaptive algorithm. The number of peer nodes to be forwarded in each hop is adaptive according to the correlation degree of the peer node to the query. Moreover, the different types of queries are also utilised in accordance with different search stages, which enables peer nodes to accumulate knowledge efficiently with low communication cost. ESLP is also a self-organising algorithm, peer communities are formed and maintained spontaneously, where peer nodes are self-organised based on their daily intercommunications. Each peer node can automatically detect potential interests of other peer nodes in the network according to their previous behaviours and preferentially links to the peer nodes that have similar interests. Global behaviours then emerge as the result of all the local behaviours that occur. Finally, the peer nodes with similar interests will be highly connected to each other.

ESLP has been simulated, as well as the derived strategies: OSLP and ASLP, in a dynamic environment with a growing number of peer nodes. From the simulation results and analysis, these social-like peer-to-peer algorithms achieved better performances, more quickly targeted more requested files and more efficiently established a knowledge index about the locations of files than current methods. ESLP achieves a cost-effective performance by quickly finding the desired files with a small traffic cost.

In future work we will further optimize and extend social-like peer-to-peer algorithms to deal with more complex queries and cooperate with semantic languages. The algorithms will also be simulated in a larger-scale peer-to-peer system over longer periods. Since resource and service discovery in Grid networks involves a lot

of elements in common with resource discovery in P2P networks, ESLP is also applicable for service discovery in the Grid networks with numerous service providers. We will try to extend current Grid standards, especially for resource descriptions and annotations, by including additional attributes for peer's specifications. With cooperation of Grid networks, the usage of P2P networks could be broadened from simple file provision to more advanced services, such as sharing redundant computing power for complicated scientific calculation and sharing extra bandwidth for real time video transmission.

REFERENCES

[1] Milgram S., The Small World Problem. *Psychology Today* 1967; 2: 60-67.
[2] Watts D. and Strogatz S., Collective Dynamics of Small-World Networks. *Nature*, 1998; 393: 440-442.
[3] Hong T., Performance. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, Oram (Eds.) A.. O'Reilly, 2001; 203-241.
[4] Kleinberg J., Small-World Phenomena and the Dynamics of Information. *Advances in Neural Information Processing System (NIPS).* 2001; 14: 431-438.
[5] Liu L., Antonopoulos N., Mackin S., Fault-tolerant Peer-to-Peer Search on Small-World Networks*, Journal of Future Generation Computer Systems*, 2007; 23(8): 921-931.
[6] Zhang H., Goel A., and Govindan R., Using the Small-World Model to Improve Freenet Performance. *Computer Networks,* 2004; 46 (4): 555-574.
[7] Cuenca-Acuna F.M. and Nguyen T.D., Text-based Content Search and Retrieval in ad hoc P2P Communities. *Proceedings of the International Workshop on Peer-to-Peer Computing*, Cambridge, MA, 2002.
[8] Khambatti M.S., Ryu K.D., and Dasgupta P., Efficient Discovery of Implicitly Formed Peer-to-Peer Communities. *International Journal of Parallel and Distributed Systems and Networks*, 2002; 5(4): 155-164.
[9] Vassileva J., Motivating Participation in Peer-to-Peer Communities. *Proceedings of the Workshop on Emergent Societies in the Agent World*, Madrid, Spain, 2002.
[10] Karp R., Shenker S., Schindelhauer C., and Vocking B., Randomized Rumour Spreading. *Proceedings of 41st Symposium on Foundation on Computer Science*, Redondo Beach, CA, 2002.
[11] Bloom B., Space/time Tradeoffs in Hash Coding with Allowable Errors. Communication of ACM, 1970; 13(7): 422-426.
[12] Liu L., Antonopoulos N., Mackin S., Social Peer-to-Peer for Resource Discovery. *Proceedings of 15th Euromicro International Conference on Parallel, Distributed and Network-based Processing*, Naples, Italy, 2007.
[13] Stoica I., Morris R., Karger D., Kaashoek M.F., and Balakrishnan H., Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. *Proceedings of ACM SIGCOMM*, San Diego, CA, 2001; 149-160.
[14] Ratnasamy S., Francis P., Handley M., Karp R., and Shenker S., A Scalable Content-Addressable Network, *Proceedings of the ACM SIGCOMM*, San Diego, CA, 2001; 161-172.
[15] Rowstron A. and Druschel P., Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems. *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms*, Heidelberg, Germany, 2001.

[16] Crespo A. and Garcia-Molina H., Routing Indices for Peer-to-Peer Systems. *Proceedings of International Conference on Distributed Computing Systems*, Yokohama, Japan, 2002.
[17] Xiao L., Liu Y., and Ni L.M., Improving Unstructured Peer-to-Peer Systems by Adaptive Connection Establishment. *IEEE Transactions on Computers*, 2005; 54: 176-184.
[18] Chawathe Y., Ratnasamy S., Breslau L., Lanham N., and Shenker S., Making Gnutella-Like P2P Systems Scalable. *Proceedings of ACM SIGCOMM*, Karlsruhe, Germany 2003.
[19] Tsoumakos D. and Roussopoulos N., Adaptive Probabilistic Search for Peer-to-Peer Networks, *Proceedings of the International Conference on Peer-to-Peer Computing*, Linkoping, Sweden, 2003.
[20] Joseph S., NeuroGrid: Semantically Routing Queries in Peer-to-Peer Networks, *Proceedings of the International Workshop on Peer-to-Peer Computing*, Pisa, Italy, 2002.
[21] Joseph S., P2P MetaData Search Layers. *Proceedings of International Workshop on Agents and Peer-to-Peer Computing*, Melbourne, Australia, 2003.
[22] Sripanidkulchai, K., B. Maggs, and H. Zhang. Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems. *Proceedings of IEEE Infocom*, San Francisco, March 2003.
[23] Crespo A. and Garcia-Molina H., Semantic Overlay Network for P2P System, Technical Report, Stanford University, 2002.
[24] Crespo, A. and H. Garcia-Molina. Routing Indices for Peer-to-Peer Systems. *Proceedings of International Conference on Distributed Computing Systems*, Vienna, Austria, July 2002.
[25] Borch, N. Social P2P for Social People. *Proceedings of International Conference on Internet Technologies & Applications*, Wrexham, UK, September 2005.
[26] Pouwelse, J., et al., Tribler: A Social-based Peer-to-Peer System. *Concurrency and Computation: Practice and Experience*, 2007; 19: 1-11.
[27] Newcomb T.M., Social Psychology: the Study of Human Interaction. - 2nd ed. Revised. London: Routledge and Kegan Paul, 1975.
[28] The Open Directory Project. Available: http://dmoz.org/.
[29] Maymounkov P. and Mazi`eres D., Kademlia: A Peer to Peer Information System Based on the XOR Metric, *Proceedings of Internation Workshop on Peer-to-Peer Systems*, Cambridge MA, March 2002.
[30] Lv Q., Cao P., Cohen E., Li K., and Shenker S., Search and Replication in Unstructured Peer-to-Peer Networks, *Proceedings of ACM SIGMETRICS*, Marina Del Rey, CA, June 2002.
[31] Kautz H., Selman B. and Shah M., Combining Social Networks and Collaborative Filtering, *Communications of ACM,* 1997; 40: 63-65.
[32] Krishna P., Measurement, Modelling and Analysis of a P2P File-sharing Workload, *Proceedings of ACM Symposium on Operating Systems Principles*, Bolton Landing, New York, 2003.
[33] Makosiej P., Sakaryan G., and Unger H., Measurement Study of Shared Content and User Request Structure in Peer-to-Peer Gnutella Network. *Proceedings of the Conference on Design, Analysis, and Simulation of Distributed Computing System*, Arlington, Virginia, 2004.
[34] Pauli C. and shepperd M., An Empirical Investigation into P2P File-Sharing User Behaviour. *Proceedings of Americas Conference on Information Systems*, Omaha, Nebraska, 2005.

[35] Distributed Computing Association White Paper, summer 2003. Available: http://dcia.info/About/summer_2003_white_paper.htm
[36] Bhagwan R., Savage S., Voelker G.M., Understanding Availability. *Proceedings of the International Workshop on Peer-to-Peer Computing System*, San Joes, CA, 2003.
[37] Ren Y., et al., Explore the Small World Phenomena in Pure P2P Information Sharing System. *Proceedings of the International Symposium on Cluster Computing and the Grid*, Tokyo, Japan, 2003.
[38] Xu J., A. Romanovsky and B. Randell, Concurrent Exception Handling and Resolution in Distributed Object Systems, *IEEE Transactions on Parallel and Distributed Systems*, 2000; l.11 (10):1019-1032,.
[39] Zhou H., Scaling Exponents and Clustering Coefficients of a Growing Random Network, *Physical Review* 2002; E66:016125.