Expert Systems with Applications 36 (2009) 10312–10324

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Improving peer-to-peer search performance through intelligent social search

Stephen J.H. Yang [a,*], Jia Zhang [b], Leon Lin [c], Jeffrey J.P. Tsai [d]

[a] Department of Computer Science and Information Engineering, National Central University, Jhongli, Taoyuan County 32001, Taiwan
[b] Department of Computer Science, Northern Illinois University, USA
[c] Enterprise Business Group of Wistron Corporation, Taiwan
[d] Department of Computer Science, University of Illinois at Chicago, USA

ARTICLE INFO

Keywords:
Intelligent social search
Peer-to-peer
Social grouping
Social network
Social routing

ABSTRACT

As a large amount of information is added onto the Internet on a daily basis, the efficiency of peer-to-peer (P2P) search has become increasingly important. However, how to quickly discover the right resource in a large-scale P2P network without generating too much network traffic remains highly challenging. In this paper, we propose a novel P2P search method, by applying the concept of social grouping and intelligent social search; we derive peers into social groups in a P2P network to improve search performance. Through a super-peer-based architecture, we establish and maintain virtual social groups on top of a P2P network. The interactions between the peers in the P2P network are used to incrementally build the social relationships between the peers in the associated social groups. In such a P2P network, a search query is propagated along the social groups in the overlay social network. Our preliminary experiments have demonstrated that our method can significantly shorten search routes and result in a higher peer search performance. In addition, our method also enhances the trustworthiness of search results because searches go through trusted peers.

1. Introduction

The explosion of Web-based technology has led to an increasingly large volume of information to be added onto the Internet on a daily basis. The traditional client/server model, where a relatively low number of servers handle the communication, has been considered inefficient for resource sharing and network management. Therefore, its alternative peer-to-peer (P2P) network model has been developed rapidly and used extensively (Kim, Kim, & Cho, 2008; Niu et al., 2007). In a P2P network, all nodes are treated as equal peers that simultaneously function as "clients" and "servers" to each other. However, how to find the right information and resource in a large-scale P2P network has remained a critical problem (Kim et al., 2008). Take Gnutella network as an example. Gnutella hosts an average of approximately 2.2 million users, with around 750,000 to one million users online simultaneously at any given moment (Hughes, Coulson, & Walkerdine, 2005). How to quickly locate interested resources in such a large P2P system is not a trivial task.

Currently, flooding is the major search algorithm in Gnutella (Hughes et al., 2005). When a peer initiates a search operation, it sends the query messages to its neighbor peers. If a peer recipient does not possess the content requested, it in turn forwards the query to its neighbor peers. The query propagation continues until relevant information is found or a predefined number time-to-live (TTL) is reached. Studies show that this commonly used search algorithm may cause severe performance problems (Hughes et al., 2005). For example, it may generate a huge amount of network traffic and cause communication congestion and slow response. If the TTL is defined very low, on the other hand, one may fail to find resources although they exist in the network. Therefore, many algorithms and approaches have emerged to enhance P2P search performance, for example, UbiSrvInt (Yuan & Chen, 2007), Freenet (Clarke, Sandberg, Wiley, & Hong, 2001), modified breath-first search (BFS) (Kalogeraki et al., 2002; Tsoumakos et al., 2006, 2003), iterative deepening (Lv et al., 2002; Yang et al., 2002, 2003), expanding ring (Hassan & Jha, 2004), random walk (Gkantsidis, Mihail, & Saberi, 2004), interest cluster (Tong et al., 2005; Borch, 2005; Cohen, Fiat, & Kaplan, 2003), trust-based recommendation (Griffiths, 2006; Li & Kao, 2009), distributed hash table (DHT) Gummadi et al., 2003, JXTA (Juxtapose) (Nottelmann & Fischer, 2007), and grid architecture (Tsai & Hung, 2009). However, as more information is added into the Internet in an increasingly rapid speed, how to facilitate P2P search performance always deserves emphasis and research.

In contrast to most of the other P2P research efforts, which are focused on search process, our research concentrates on how to propagate query messages to relevant nodes and how to locate appropriate content. In this paper, we propose a novel P2P search

* Corresponding author. Tel.: +886 3 4227151x35308; fax: +886 3 4222681.
E-mail address: jhyang@csie.ncu.edu.tw (S.J.H. Yang).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.01.045

method by applying the concept of social grouping and intelligent social search to derive socially related group of peers in a P2P network. Through super-peer architecture, we establish and maintain a virtual social group as an overlay network on top of a P2P network to help P2P search. Our approach contains three core algorithms. First algorithm aims to constitute virtual social groups with peers that have close preferences and high similarity. Second algorithm aims for social routing, so that a query is propagated to the corresponding friend peers in the same social group. Third algorithm aims to dynamically maintain the virtual social groups through social network analysis (Carrington, Scott, & Wasserman, 2005) methods.

Our main contributions are twofold. First, we propose an approach to form and maintain virtual social groups among peers in a P2P network, and use the built social network to assist in exploring resources and knowledge in the P2P network. Second, we have designed and implemented a prototype of social P2P system and proved its ability to enhance P2P search performance. Our experiences and observations of building this prototype system will guide the construction of a social P2P network in the real world.

The remainder of the paper is organized as follows. In Section 2, we explain our research motivations and strategy of applying social networks to P2P search. In Section 3, we discuss related work. In Section 4, we present our social networks-based P2P search method, explaining in detail our algorithms of forming a social network, social routing, and maintaining social relations. In Section 5, we present our experimental settings and results summary and analysis. In Section 6, we make conclusions.

2. Research motivations and strategy

In this research, we propose a novel P2P search method by applying the concept of social grouping and intelligent social search to a P2P network. Social science research has revealed that people build social relationships with each other and these relationships may help people in finding appropriate information or services more effectively (Beydoun, Kultchitsky, & Manasseh, 2007; Yang & Dia, 2008; Carrington et al., 2005; Wang & Chiu, 2008). Instead of randomly broadcasting search requests, one asks its acquaintances for help. If the acquaintances do not have the answer, they will pass the query to their direct connections, and so on. The search process thus becomes more efficient because of the parallelism and search propagation along the social links. Rooted deeply in social science, a social network refers to an abstract network structure that describes the relationships between a set of nodes, each representing a social unit such as person, a group, an organization, or a computer. In its simplest form, a social network can be viewed as a graph comprising relations between contained nodes. Therefore, graph theory may be adopted to investigate the interaction patterns and their influences in a social network.

The reason why we investigate social networks is because the similarities between a social network and a P2P network. The peers in a P2P network can be viewed as the participants in a social network; the edges in a P2P system can be viewed as the social ties in a social network. The social relationships between a pair of peers are evaluated into a similarity degree, based on a vector of quantifying decision factors including peer profiles, preferences, and so on. According to the quantified social relations, peers are assigned to appropriate virtual social groups to facilitate resource discovery processes.

Throughout this paper, we will take Gnutella (Hughes et al., 2005) as an example for our discussions about P2P networks as well as the testbed for our research. The reason is twofold. First is its popularity. Gnutella is ranked as the third most popular system for file sharing, including academic lectures sharing that is the direct goal of this research. Second is its openness. Gnutella is an open, decentralized group membership and search system. In a Gnutella network, each peer plays as both a client and a server, so it is called a servent (i.e., an abbreviation of server and client).

3. Related work

Generally speaking, P2P searching algorithms can be classified into two categories based on the topology structure of the P2P network: unstructured and structured. In a structured P2P network, peers keep connection information of each participant as well as network topology. Upon receiving a query, a peer searches the resources based on the topology rules. It is a guarantee that a target will be found if it exists in the network. Representative searching methods oriented to structured P2P networks include Chord (Gkantsidis et al., 2004; Ratnasamy, Shenker, & Stoica, 2002; Stoica et al., 2001) and CAN (Ratnasamy et al., 2002). In an unstructured P2P network, peers join in the network through loose rules. There is no control over the network topology. Thus, it is easy to create and maintain an unstructured P2P network. However, searching in an unstructured P2P network is not guaranteed even if the target information exists in the network. Representative searching methods oriented to unstructured P2P networks include Gnutella protocol (Hughes et al., 2005). In this research, we focus on information search in an unstructured P2P networks.

3.1. Gnutella search algorithm

Gnutella protocol refers to a set of communication protocols for searching and sharing resources in a Gnutella-like P2P network (Ripeanu, Iamnitchi, & Foster, 2002). Flooding is its major search algorithm (Hughes et al., 2005), which is a breadth-first search (BFS) algorithm that utilizes a time-to-live (TTL) controlled broadcasting mechanism to explore a P2P network through connections and search propagation. Fig. 1 illustrates the strategy. A query is initiated from peer A and is sent to all its neighbor peers (B, C, D, and E) in the P2P network. If none of these peers contain the requested knowledge, each one propagates the query messages to all of its own neighbor peers (e.g., peer E to peers R, S, T, and U), and so on. This query forwarding process will be repeated TTL times, which is predefined, unless the expected resource is found. As shown on the left of Fig. 1, TTL is equal to two. The query flows two layers centered by peer A. On the right of Fig. 1, TTL is equal to five. If a query cannot be satisfied after five propagations (through peers B, H, J, K, and L), it is considered as a failure, and the forwarding is stopped.

Studies show that this commonly used search algorithm may cause severe performance problems. The major reason is that TTL is predefined and used to control the number of hops that a query may be propagated. How to choose an appropriate TTL is not a trivial task. If the TTL is set too high, it will generate a large amount of network traffic flooding over the network thus lead to communication congestion and slow response. On the contrary, if the TTL is set too low, one may fail to find the expected resources although they exist in the network. This formula shows that the flooding algorithm is highly effective for popular items and less effective for rare items. Take the left scenario in Fig. 1 as an example. For simplicity, assume the system adopts a full-degree n-nary peer structure. In the worst case, the flooding algorithm generates the number of traffic units calculated as follows:

$$n(\text{Flood}) = \sum_{t=1}^{\text{TTL}} n^t = n + n^2 + \cdots + n^{\text{TTL}} = \frac{n(n^{\text{TLL}} - 1)}{n - 1} \text{ units}$$

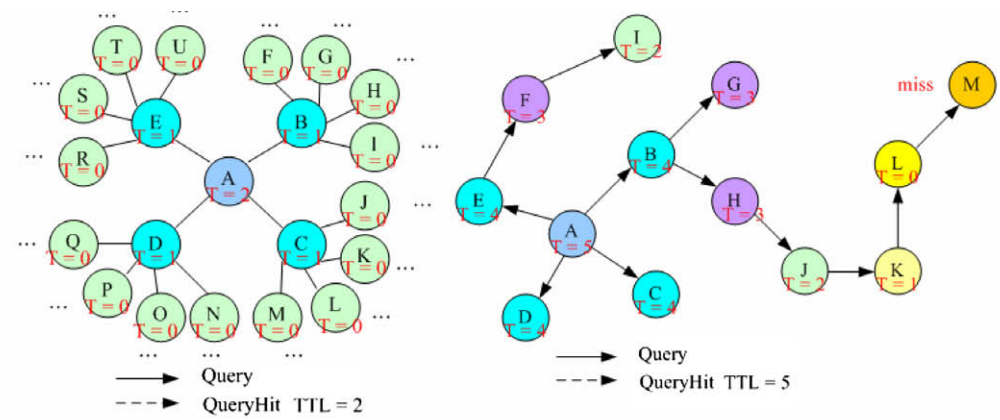

Fig. 1. Flooding search propagation in Gnutella network.

In contrast to the flooding algorithm that sends queries to all connected neighbors, our approach only broadcasts queries to peers with high probability to answer the queries. Therefore, network traffic may be significantly reduced and the result may be quicker to be found.

Many researchers have proposed approaches to improve Gnutella-like P2P system search method. Using the taxonomy proposed in Tsoumakos et al. (2006, 2003), we classify them into two categories: uninformed search method and informed search method. The former strategy intends to propagate query messages to a sufficient number of peers to find appropriate resources. Representative methods include Freenet, modified BFS search, iterative deepening, expanding ring, and random work. The latter exploits resource location information to expedite search process. Representative methods include Napster, super-peer approach, intelligent BFS, interest cluster, and DHT.

3.2. Uninformed search methods

Opposite to the flooding algorithm, Freenet (Clarke et al., 2001) uses a depth-first traversal (DFS) strategy. The query message is first sent to one single neighbor peer with a predefined TTL constraint. This selected peer may in turn forward the query to one of its neighbor peer, and so on. TTL controls the times of query forwarding down the connection path. If within TTL turns, the expected knowledge is not found, another path will be tried. Using Freenet strategy, all search paths are examined sequentially. Since search process can be terminated whenever the query is satisfied, the overall cost (traffic) can be minimized. However, sequential execution may lead to poor response time, with the worst case being exponential of TTL.

With modified breath-first search (BFS) (Kalogeraki et al., 2002; Tsoumakos et al., 2006, 2003) method, one peer sends query messages only to a selected subset of its neighbor peers, instead of to all neighbor peers as flooding algorithm does. The selection is based on past quality responses from previous queries. Compared with the flooding algorithm, this algorithm may cause significantly less traffic and quicker response time. However, how to make an intelligent selection remains a challenge.

Iterative deepening (Lv et al., 2002; Yang et al., 2002, 2003) uses consecutive BFS searches at an increasing depth. Source peer S initiates a BFS of depth a, by sending out a query message to all its neighbor peers. The query will be frozen at all peers a hops away from S. S gathers responses from those peers that have processed the query within a predefined time period W. If the query is not yet satisfied, S will start the next iteration of BFS search with depth $(b - a)$ by sending a *resend* message to all peers a hops away. A peer receiving a resend message unfreezes the query (stored temporarily) and forwards it to its neighbor peers. This process continues in the similar fashion until a predefined depth D is reached. At depth D, the query is dropped. Iterative deepening approach periodically waits to check whether the query is satisfied. Thus, the number of query propagation may be reduced if the query is satisfied at some point. However, this method still does not solve the traffic problem when the expected knowledge is far away from the source peer.

Expanding ring algorithm (Hassan & Jha, 2004) is an extension of the flooding search algorithm. It performs successive flooding searches with an incremental TTL. Initially, source peer sends query messages with a small TTL and waits in a predefined time period for responses. If the query is not satisfied in the time period, the search continues to flood through the P2P network with a larger TTL. The process continues until TTL reaches a predefined upper threshold or the query is satisfied. Random walk algorithm (Gkantsidis,Mihail, & Saberi, 2004) intends to avoid the scalability problem of the flooding algorithm and achieve reasonable search performance. The key mechanism is to forward the query messages to randomly selected neighbor peers at each step, until the target is found. In this way, the number of query messages walking through the P2P network at any time point remains a constant K as the search time elapses. Gkantsidis et al. (2004) have concluded that random walk algorithm performs well under two circumstances: when the overlay topology of a P2P network is clustered and when a client re-issues the same query while its horizon does not change much. However, the algorithm suffers from the fact that its success rate and the number of its hits vary significantly depending on the peers randomly chosen.

3.3. Informed search methods

Gnutella2 (G2) Tsoumakos & Roussopoulos, 2006 proposes a super-peer-based P2P architecture (Wiesner, Kemper, & Brandl, 2004). Some super-peers are chosen to act as servers, each being in charge of a set of regular peers using a star-like fashion and taking care of some typical responsibilities of a server in a centralized P2P system (Yang et al., 2002, 2003; Stoica et al., 2001; Jesi, Montresor, & Babaoglu, 2006; Li et al., 2004; Montresor, 2004). Super-peers manage routing indexes for its controlled peers and neighbor super-peers. Upon receiving a search query, it first forwards the query to its child peers. If the query is not satisfied, the super-peer forwards the query to its connected super-peers. Thus, the super-peer technique eliminates some performance overheads of a primitive P2P network. However, how to select appropriate super-peers remains a big challenge. With the dynamic nature of an Internet-oriented P2P network, this approach bears low flexibility and scalability. In contrast, our proposed method does not select any super-peers. Instead, each peer maintains its relationships with interacted peers. This knowledge can be used to facilitate search propagation process. Therefore, our approach possesses higher scalability and flexibility.

Intelligent BFS (Kalogeraki et al., 2002; Tsoumakos et al., 2006, 2003) algorithm is an informed version of modified BFS algorithm. Every peer maintains a set of tuples (keyword, ID), each representing for a neighbor peer (with ID) that recently replies the query (with keyword). Upon receiving a search query, a peer first identifies all queries similar to the one received using a query similarity metric. Then it forwards the query message to the neighbor peers that have replied most previous similar queries. Whenever a hit occurs, the query takes a reverse path to the original requester peer and updates the local indices for all peers along the path. This method uses dynamically updated tuples to select query routing paths for reducing the amount of traffic messages.

Distributed hash table (DHT) is a mechanism to structure the topology of a P2P network mathematically for higher scalability and determinacy (Gummadi et al., 2003). DHT records peers. Every peer is assigned a unique system-generated key when it joins the network. This key is equally spread mathematically around the neighbor peers, so that all peers in the structure may potentially be used as a small server for a local area. Upon receiving a key, a peer uses some hash function to store the key in its DHT. When a peer needs to forward a request, it shall forward the message the (mathematically) closest peers according to its DHT. Representative search methods upon a DHT-equipped P2P network include Chord (Gummadi et al., 2003; Ratnasamy et al., 2002; Stoica et al., 2001), content addressable network (CAN) (Ratnasamy et al., 2002), and Pastry (Rowstron & Druschel, 2001). The advantage of DHT is that a DHT system may route a message to a peer in $O(\log n)$ hops at worst case, where n is the number of the peers in the system Chord (Gummadi et al., 2003; Ratnasamy et al., 2002; Stoica et al., 2001). The disadvantage of DHT is the overhead of maintain the hash table structure, especially when some peers join or leave the network frequently. In addition, DHT normally only supports keyword-based search (Gummadi et al., 2003; Ratnasamy et al., 2002; Stoica et al., 2001).

One method commonly used to facilitate P2P content location is through the idea of interest cluster (Tong et al., 2005; Borch, 2005; Cohen et al., 2003). Based on semantic meanings, resources are categorized into domains, such as sports, music, entertainment, and so on. Peers can be grouped into interest clusters by calculating the similarities between them according to their carried resources (metadata). Its basic idea is to utilize semantic similarities to facilitate search process and enhance scalability. The basic formula to calculate the degree of similarity between two peers is illustrated as:

$$S_{XY}^{D} = R_{XY}^{D}/Q_{X}^{D}$$

where S_{XY}^{D} is the similarity degree in domain D between peer X and peer Y; R_{XY}^{D} is the number of interactions in domain D; Q_{X}^{D} is the total number of queries initiated by peer X in domain D. The value of similarity is between 0 and 1, where 1 denotes a complete similarity.

In contrast to the interest cluster approach that forms groups using semantic meanings of static content, our approach forms virtual social groups using dynamic interrelationships (e.g., preferences, interaction history, and trust relationship). Therefore, our approach reflects more dynamic feature of an Internet-based P2P network.

3.4. Comparisons

The uninformed search approaches focus on reducing the number of query messages flooding over a P2P network, such as random walker, consecutive BFS searches, and DFS. However, these methods may also suffer from the tradeoff of less hit rate because some peers actually containing the expected information may be missed. The informed search approaches, on the other hand, focus on quick target discovery. They pay the cost of more query messages propagated simultaneously over a P2P network, but usually achieve higher hit rate and higher accuracy.

The major difference between our social search-based P2P network and a typical P2P network is that our model explores and applies social relations between peers. In contrast to a nonsocial P2P network that propagates query messages to a sufficient number of peers or establishes hash tables for routing, our method tries to find trusted or relevant peers to help accomplish search operation. In detail, comparing with existing P2P search approaches, our method stands out in the following three aspects:

(1) Peers in our P2P system are grouped based on similar preferences, interests, or background knowledge. Interest cluster method can be viewed as a special case of our social grouping.
(2) Peers in our P2P system virtually connect to friend peers instead of anonymous peers in a regular P2P system. The degree of trust between friend peers may be higher, because they are more likely to be trusted than anonymous peers.
(3) In addition to similarities and preferences, we monitor the in-degree and out-degree centrality among peers to ensure dynamic group maintenance by adopting the social network analysis (SNA) (Carrington et al., 2005) method.

To our best knowledge, our research is the first attempt to take into consideration of social aspect to enhance P2P search performance. Our approach exploits trusted or relevant peers to help accomplish search more effectively and efficiently. We use social relations between peers, such as friendship and existence of communities, to further enhance the usability and performance of P2P search.

4. Intelligent social search

We apply the concept of the social networks to facilitate information search in a P2P network. Through super-peer architecture, we establish and maintain a social group of peers as an overlay network on top of a P2P network. The interactions between the peers in the P2P network are used to incrementally build the social relationships between the peers in the associated social network. We adopt the multidimensional scaling (MDS, see Kruskal & Wish, 1978) and social network analysis (SNA, see Carrington et al., 2005) techniques to help constitute and maintain the virtual social groups with similar interests or preferences. In such a P2P network, a search query can be propagated along the social groups. We introduce a concept of "friend peers" to represent the peers categorized into the same social group. Friend peers are used to help explore information in the P2P network in an effective and efficient manner.

4.1. Social network construction

Multidimensional scaling (MDS) (Kruskal et al., 1978) is a set of inter-related statistical techniques widely used in many fields (e.g., information science, psychophysics, psychometrics, social science, marketing, and ecology) for data reduction and data mining to explore information hidden in data. In this research, we apply MDS for data visualization to explore similarities and dissimilarities among data. Our core idea is to use data relationships and user preferences to develop a multidimensional space called a

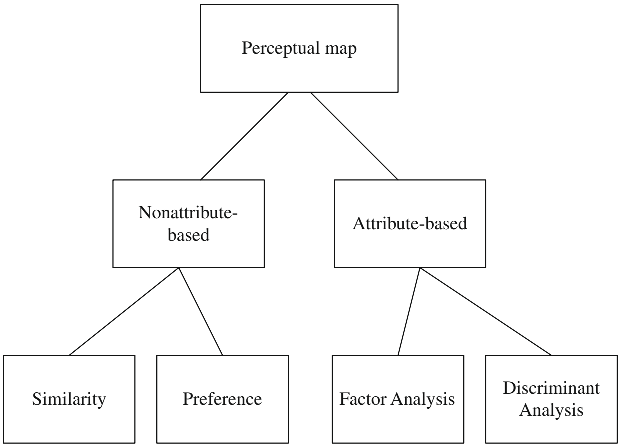

Fig. 2. Classification of perceptual map.

perceptual map, which is a graphics technique that visually displays the similarities or dissimilarities among data.

Fig. 2 illustrates our basic method of how to generate a perceptual map. Without losing generality, we allow two strategies to extend considerations, either an attribute-based approach or a non-attribute-based one. An attribute-based approach adopts some techniques (e.g., factor analysis and discriminate analysis) to analyze the data set under investigation and establish a set of attributes for data visualization. A non-attribute-based approach identifies some subjective considerations (e.g., similarities and user preferences) as dimensions for data visualization.

We use the factor stress (Kruskal et al., 1978) to represent the goodness-of-fit (i.e., consistent matching degree). The less the stress, the better the goodness-of-fit. In this research, we apply the monotone regression method (Schell & Singh, 1997) to calculate the deviation value $\hat{d}_{ij}$ of d_{ij}.

$$S = \left[\frac{\sum\sum\left(d_{ij}-\hat{d}_{ij}\right)^2}{\sum\sum(d_{ij})^2}\right]^{1/2},$$

where S indicates a stress value. d_{ij} indicates the distance between i and j. $\hat{d}_{ij}$ indicates the deviation value of d_{ij}.

Social network analysis (SNA) (Carrington et al., 2005) refers to the methods for observing a social network and providing mathematical and visual analysis of the relationships between the nodes contained in the social network. The relationships are typically mapped and measured into various social relationship ties. Social network analysis may be conducted either from the perspective of a single node or the overall network. Some common social network analysis methods include centrality, intensity, clique, and so on (Carrington et al., 2005).

In a P2P network, each peer may exhibit different features such as connectivity capabilities, available bandwidth, CPU power, and available time. To facilitate P2P communication, we select some peers to serve as so-called super-peers, which act as local servers for handling some local search routing and task dispatch in a centralized manner. Such super-peers also communicate with other super-peers for achieving higher parallelism. In this research, we define a friend peer at a conceptual level. For a specific node, a friend peer may represent an actual friend's node in the real life or a node with which it has good interaction (search) experiences.

4.2. Social groups

In the real life, each person may have relationships (i.e., ties) with many people through their conceptual social networks, either directly or indirectly. Therefore, such a social network may contain hierarchical friend-relationships. For a specific node, it may have a set of friends (F-1s), a set of friends of friends (F-2s), a set of friends of friends of friends (F-3s), and so on. The numbers show the distance of friendships. Fig. 3a shows an example of social network. From the perspective of node A, it has four F-1s (B, C, D, and E) and eight F-2s (F–M). Some of these friend-relationships overlap. For example, C and D are direct friends to each other, while they are both direct friends of A. Fig. 3a also shows that some F-2s may be reached through multiple friends. For example, node M can be reached through two individual F-1s, D, and E, respectively.

In a social network, the concept of social group represents a certain number of people who share some common features (e.g., same interest, close backgrounds, or socially related) and are called friends. The importance of the concept is apparent, since it helps

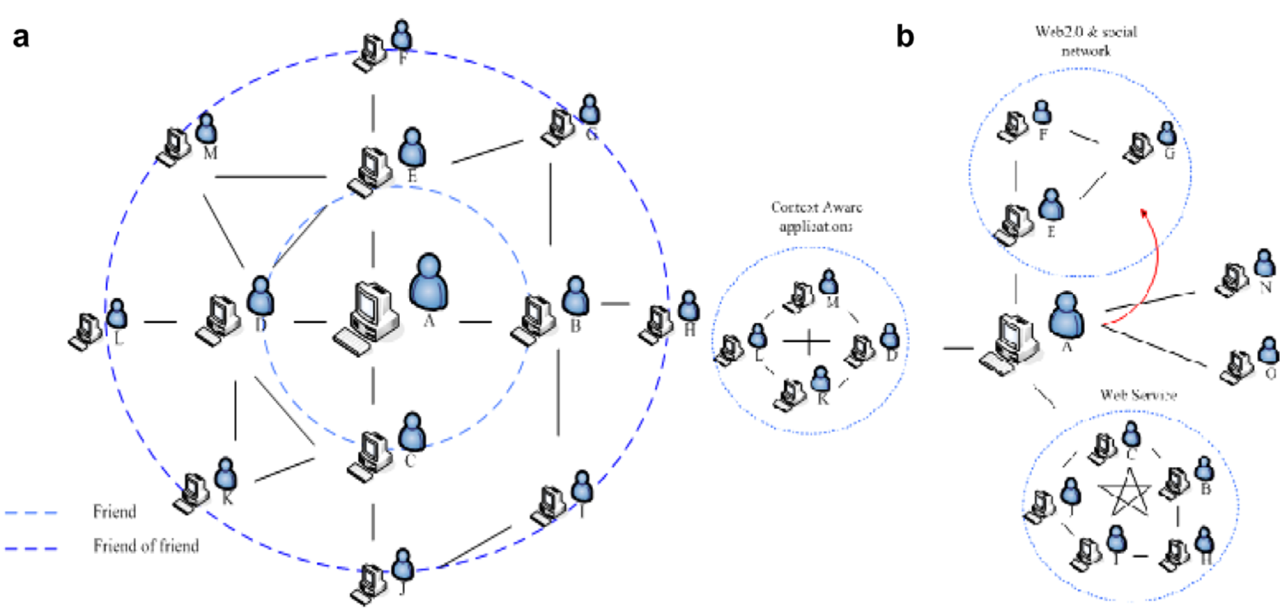

Fig. 3. Concept of social group.

friends to be clustered with virtual connections to help each other. From this sense, a social network can be represented as an overlay network (Yang & Dia, 2008; Wang & Chiu, 2008).

In the real life, a person may belong to multiple groups. For example, Fig. 3 shows three groups: context aware application group, Web2.0 and social network group, and web service group. When participant A intends to ask some questions about Web2.0 and social network, she first queries her direct friends C, D, E, N, and O. Participant E belongs to the Web2.0 and social network group; so she may response. Further, E might also introduce A to her group for further discussions. Eventually, A might even join in the group if she is interested.

Note that many similarities exist between a social network and a P2P network. For example, participants in a social network can be viewed as peers in a P2P network; the social relation ties in a social network can be viewed as the connections in a P2P network. We apply the concepts of social group into a P2P network for improving its search performance. Similar to the fact that friends in a social network may be more willing to help, we exploit friend relations to increase hit ratio of P2P search. Query routing is handled based on social relations; meaning that the queries be routed to the friend peers only, instead of to all participants through a broadcast.

Peers with similarities are identified and clustered into groups based on some criteria, such as same interest or background on a given domain or topic. Fig. 4 shows a P2P network organized under the concepts of social group and super-peer. Each group is represented by a super-peer, which handles other peers using either an asterisk-style or a tree-like structure. With the social relations, the peers in a group are relatively socially close to each other, so searching in the group may shorten the routing length in the network thus reduce bandwidth consumption. If the group does not contain the content, the query will be forwarded to messages to the connected super-peers, which in turn takes charge of gathering the feedback in their groups.

Taking Fig. 4 as an example, assume that peer A intends to search lectures about Web2.0 and social network, so it sends the query message to its super-peer O. Peer O processes the query on behalf of A and forwards the query to the entire group. Not finding the content, peer O forwards the query to its connected super-peer P. Since peer P also does not have the content, it in turn sends the query to its connected super-peers: C, E, and M. Three of them broadcast the query to their own groups. Finally, peer F responses with a positive result. The feedback is thus returned to peer A through the backward path of peer E, P, and O.

4.3. Three phases of social search

The procedure of our social search is divided into three phases: social group formation, semantic social routing, and social group maintenance.

- *Phase 1* is to constitute social groups with peers having close preferences and similarities. Adopting MDS method, we use similar degree to measure and categorize peers and groups. This phase includes three steps: questionnaire, distance measurement, and perceptual map.
- *Phase 2* is semantic social routing. In our approach, each peer keeps a list of its friend peers. Queries are not forwarded to neighbors peers; instead, they are forwarded to friend peers.
- *Phase 3* aims to dynamically maintain the social groups using SNA methods. Observation peers (peers with changes) are identified and adjusted to relevant groups.

The underpinning of our approach is a hypothesis that people with similar preferences and interests may be more capable and

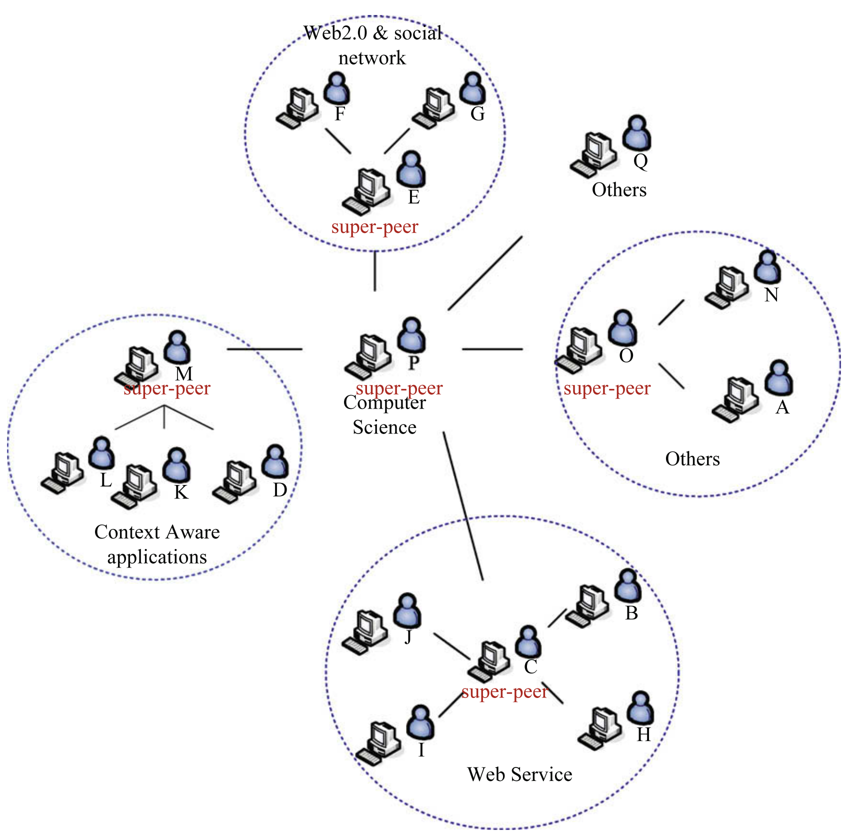

Fig. 4. Social group in a P2P network.

willing to help each other. Based on this hypothesis, we conduct social grouping by building virtual groups containing peers with similar preferences or similarities. The initial knowledge base is established by volunteer questionnaire fulfillment by peers when they join in the network. This information may not be accurate (e.g., some peers may not fill the questionnaire). However, as time goes by, after peers participate in communications and interactions, more information may be obtained and the group forming may be adjusted with more accurate information.

The research challenge underneath is how to enhance P2P search. Our research strategy is to explore social grouping among peers to enhance P2P search efficiency. More specific, we dynamically detect and maintain social relationships among peers to enable feasible social grouping.

4.4. Phase 1: social group construction with MDS

By adopting multidimensional scaling (MDS), our social group construction aims to classify peers according to their preferences and similarities on certain classification bases. In this research, we use ACM Computing Classification System (1998) version as our classification base. Eleven categories (A–K) are identified. Taking category "D. Software" as an example as shown in Fig. 5, it is further classified into six sub-categories: general, programming techniques, software engineering, programming languages, operating systems, and miscellaneous. Similarly, the programming techniques can be further classified into more detailed directions.

Using the classified domain information as dimensions, we create a questionnaire to obtain user preferences. The results become the initial input data for performing multidimensional scaling. Table 1 shows an example of the resulted preference table. The objectives are users and the domains are the items in ACM classification system. The values in the table are measured based on a seven-point scale ranging from (1) strongly dislike to (7) strongly like. For example, Wendy likes domain "Web2.0" very much (with a score of seven); Charles dislikes "social network" topic at all (with a score of one).

Tables 1 is then taken as input data to apply the MDS method. First, we calculate the distance (preference or similarity) between each pair of peers, which is defined using the Minkowski model (Kruskal et al., 1978).

$$d_{ij} = \left[\sum_a |x_{ia} - x_{ja}|^p \right]^{1/p}$$

where x_{ia} denotes the position of value of object i on dimension a. The value of p is preset by users.

Then, we calculate the similarity factor between each pair of peers, by normalizing the distance value as follows:

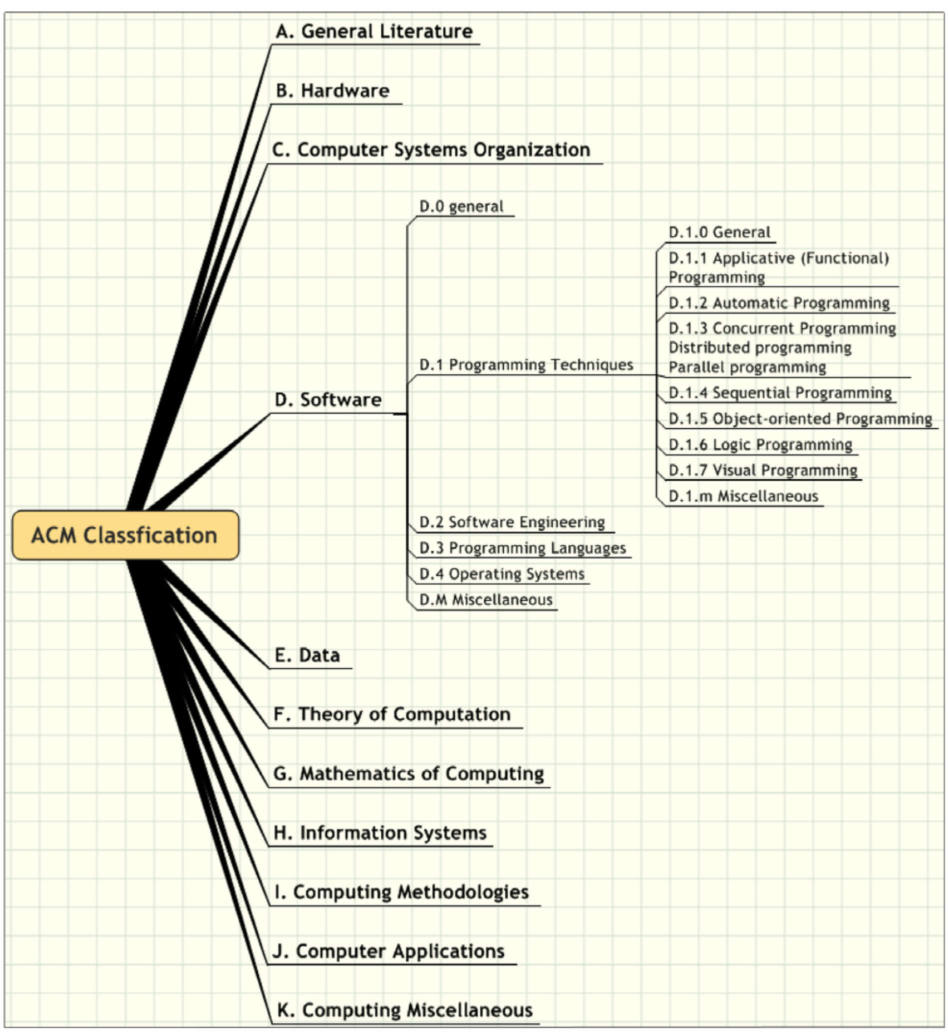

Fig. 5. A partial list of ACM computing classification.

$$S_{ij} = 1 - \frac{d_{ij}}{\text{Maximum distance}} = 1 - \frac{\left[\sum_a |x_{ia} - x_{ja}|^p\right]^{1/p}}{\text{Maximum distance}}$$

Table 2 is the corresponding Minkowski distance matrix resulted from Table 1. Table 2 is thus transformed into a similarity matrix as shown in Table 3.

If the similarity between a peer satisfies $S_{ij} \geqslant \gamma$, where the value of γ is preset, they are put into the same group. How to decide the value of γ depends on users. Basically, a higher value implies that peers in a group share higher similarities. If a peer can be put into more than one group, we could either allow it to be listed in multiple groups or put it into the group with the highest similarity. Without losing generality, we set γ as 0.75 and apply to Table 3. Thus we obtain three groups: (Leon, Justin, Tony), (David, Wendy), and (Charles, Stella, Frank).

Our next step is to utilize the MDS methods to create a perceptual map to provide the visualization of data similarity. Fig. 6 is the perceptual map generated by SPSS, with two representing values, RSQ and Stress. RSQ value is the proportion of variance of the scaling data in the partition. The maximum value of RSQ is one. The higher RSQ is, the better it is. Stress value is calculated by the Kruskal's stress formulation, as illustrated below (Table 4). The stress is 0.08933 and the RSQ is 0.95173. This means that the perceptual map matching is acceptable. Fig. 6 shows the similarity between peers.

Then we proceed on keeping the information of groups in peer profiles. So we can organize similar peers into preference groups on top of P2P networks. Fig. 7 represents the key idea. The first and lowest layer is the Gnutella-like P2P network. The second layer applies the group concept to cluster similar peers to form preference groups. The third layer is super-peer layer, we select a super-peer to take charge of a centralized control. Fig. 7 shows that four peers (A, C, F, and I) form an preference group on top of a P2P network.

Up to this point, an initial preference (interest) group structure is constructed. To maintain the structure afterwards, each peer in the system has to keep certain information in its peer profile, such as its group members, super-peer (group leader), group status, neighbor groups, and friend peers. Table 5 and 6 are two examples of group information and a peer profile.

Here the term of "friend peer" may be objective instead of subjective. One might manually add his/her real life friend as a friend

Table 1
A sample preference table.

Domain\object	Leon	Justin	David	Charles	Wendy	Stella	Harry	Tony	Frank
I.2.6 Knowledge engineering	5	6	3	2	3	4	4	7	4
K.3.1 Context aware learning	5	4	4	3	4	3	2	6	4
C.5.m Context adaptation	3	5	4	7	4	7	3	4	7
H3.4 Distributed systems	7	7	4	4	3	3	7	4	5
H5.4 Mobile multimedia applications	4	3	3	6	4	5	7	3	4
D.2.2 Semantic web service	6	5	6	5	7	4	5	5	5
H.5.3 Social network	7	7	5	1	4	3	3	7	3
D.4.3 Ubiquitous computing	4	4	7	3	5	4	4	3	5
A.0 Web2.0	6	6	4	6	7	5	5	6	5

Table 2
Minkowski distance matrix.

	Leon	Justin	David	Charles	Wendy	Stella	Harry	Tony	Frank
Leon	0								
Justin	2.828	0							
David	5.745	6.083	0						
Charles	8.944	8.718	7.550	0					
Wendy	5.831	6.481	4.123	5.831	0				
Stella	7.681	6.856	5.831	3.606	5.196	0			
Harry	6.083	6.708	6.782	6.083	6.403	6.164	0		
Tony	4.243	4.000	6.708	9.381	6.325	7.141	8.307	0	
Frank	6.403	5.568	4.690	4.359	4.796	2.828	5.831	6.708	0

Table 3
Similarity matrix.

	Leon	Justin	David	Charles	Wendy	Stella	Harry	Tony	Frank
Leon	0								
Justin	0.843	0							
David	0.681	0.662	0						
Charles	0.503	0.516	0.581	0					
Wendy	0.676	0.64	0.771	0.676	0				
Stella	0.573	0.619	0.676	0.800	0.712	0			
Harry	0.662	0.627	0.624	0.662	0.644	0.658	0		
Tony	0.765	0.778	0.627	0.479	0.649	0.703	0.538	0	
Frank	0.644	0.691	0.739	0.758	0.734	0.843	0.676	0.627	0

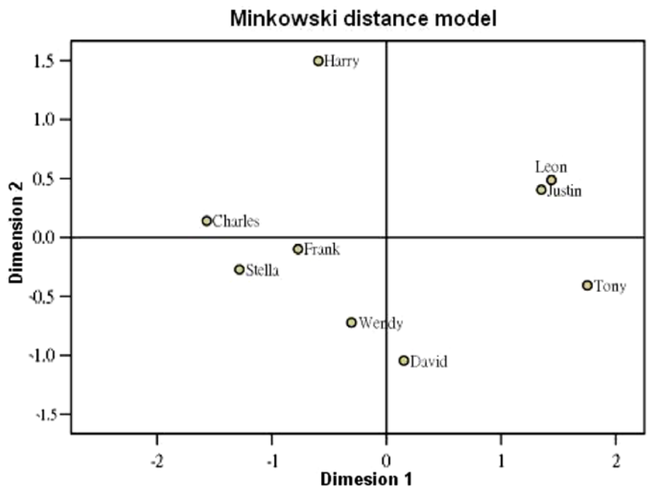

Fig. 6. Perceptual map.

Table 4
Explanations of Kruskal's stress.

Stress	Matching degree
0.200	Poor
0.100	Fair
0.050	Good
0.025	Excellent
0.000	Perfect

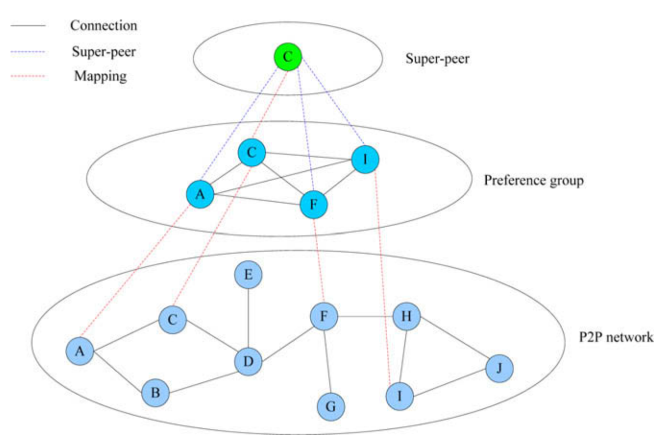

Fig. 7. Preference group structure.

peer, even without actual network interactions before. Our concept of friend peer is analogous to how people form their social networks.

Our way of grouping similar peers is analogous to that of grouping people with similar background knowledge and interest in the real life. People with similar backgrounds often have similar resources. If a peer cannot get a reply from its group, it will ask his friends, similar to the situation in the real life. In other words, we adopt a two-phase approach. The first phase is semantic grouping that groups similar peers based on semantic meanings (e.g., background and preference). The second phase is social grouping that maintains social friend list. In this way we say that our search mechanism combines semantic and social aspects.

4.5. Phase 2: intelligent search with social routing

Our search routing comprises four major steps as shown in Fig. 8. When a peer receives a query:

- *Step 1:* The query is first processed within the group to which it belongs. If the query is satisfied, then the searching process is stopped. Otherwise the flow goes to Step 2 if the peer has friend peers, or Step 3 if it does not have friend peers.
- *Step 2:* The query is forwarded to all friend peers, each processing the query within its own group. If any friend peer responses the query, the searching process is stopped. Each friend peer that does not find requested information returns its group information to the requestor. The flow then goes to Step 3.

Table 5
Example of group information.

Group ID	Member peer	Status
Group 1	Node A	Leaf
Group 1	Node C	Super-peer
Group 1	Node F	Leaf
Group 1	Node I	Leaf
Group 1	–	–

Table 6
Example of peer information.

Friend peer list	Neighbor peer list	Recent past query
Node B	Node H	Node C
Node E	Node J	Node A
Node G	–	–

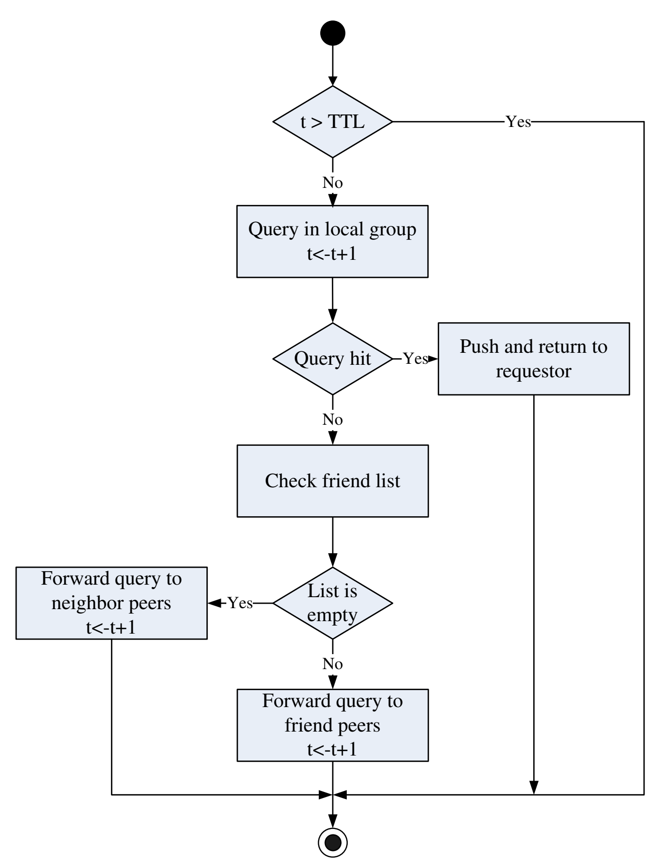

Fig. 8. Procedures of social routing.

- *Step 3:* The query is forwarded to the super-peer of the requestor, which then forwards the query to its neighbor groups. If any peer responses the query, the searching process is stopped. Each group that does not find requested information returns its group information to the super-peer, which in turn forwards it back to the requestor. The flow then goes to Step 4.
- *Step 4:* After the requestor receives the information of the friend of friend's groups (or neighbor's neighbor groups), it selects some of them to repeat Step 2 or 3 accordingly. This process continues until some preset ending condition is met, e.g., TTL reaches zero.

Fig. 9 illustrates a example of our social routing. Assume that peer I in the left interest group intends to find some resource. It first sends the query to its interest group through its super-peer (peer C). If any peer responses the query, the search will be stopped. Otherwise, it sends the query to its friend peers (peers B, E, and G). Each friend peer processes the query in its own interest group. If the answer is still not found or peer I does not have friend peers, peer I asks its super-peer (peer C), which then forwards the query to its neighbor super-peer (peer H). Peer D processes the request in its interest group (the right interest group). If the request is still not satisfied, peer I will select some groups (from its friend peers' residing groups and peer H's group) and repeat the search process.

4.6. Phase 3: dynamic group maintenance with SNA

To further enhance P2P search, we monitor and track search processes to dynamically adjust and maintain semantic groups and social groups based on social network analysis (SNA). We use degree centrality to record and measure a peer's request (In) and response (Out) for queries.

$\mathrm{In}_{I \to J} = 1$ denotes a response from peers I to J;

$\mathrm{Out}_{I \to J} = 1$ denotes a request from peers I to J.

The value of relative in-degree $\mathrm{In}_{I \to J}$ denotes the total number of responses returning to peer J; the value of relative out-degree $\mathrm{Out}_{I \to J}$ denotes the total number of requests sent from peer I. If a peer has high relative out-degree, it means that it has high influence rate (response). If a peer has high relative in-degree, it means that it obtains high support (query).

A peer may switch interest groups if it gets more support (responses) from other groups. For example, consider a peer *o* has a local peer *l* in the same group and a remote peer *g* from another group. At some point of time, assume that the analysis releases the following fact, which shows that peer *o* gets many more responses from *g* instead of from *l*. Then peer *o* should be placed under observation. If the situation stays for a certain period of time, our system will suggest that peer *o* be moved to peer *g*'s group.

$$\frac{\mathrm{In}_{l \to o}}{\mathrm{Out}_{o \to l}} \ll \frac{\mathrm{In}_{g \to o}}{\mathrm{Out}_{o \to g}}$$

Similarly, if peer *o* always responses to a foreign peer *g* more than to a local peer *l*, as shown below, then peer *o* should also be placed under observation. If the situation continues to stay, our system will also suggest that peer *o* be moved to peer *g*'s group.

$$\frac{\mathrm{In}_{o \to l}}{\mathrm{Out}_{l \to o}} \ll \frac{\mathrm{In}_{o \to g}}{\mathrm{Out}_{g \to o}}$$

5. Experiments and discussions

5.1. Simulation design and experimental setup

After building the prototype system, we designed and conducted a set of experiments to evaluate the performance our social network-based P2P search approach. Without losing generality, we built a P2P simulation environment as follows. We simulated a P2P topology graph in a dynamic environment with 500 peers. Each peer randomly connects to one to four peers bi-directionally. A set of randomly selected peers are used as super-peers. Other peers randomly connect to these super-peers. Through our experiments, we generated five different P2P topology graphs.

In our simulated environment, we created 10,000 resources (course materials) distributed on 1287 topics using a normal distribution. Based on the ACM Computing Classification System (1998), each resource has one mandatory topic and two secondary topics. After the resource distribution, we allocated the 10,000 resources to 500 peers based on the preference of each peer. For each peer, we randomly generate its preference on each of the 1287 topics, from 0.00% to 100.00% inclusive. If a peer has a preference of greater than 70% on a topic, we allocate a copy of the resource on the peer. About 70% of resources are allocated using the above algorithm. The rest 30% of resources are allocated randomly on the peers. It can be seen that our simulated resource distribution on

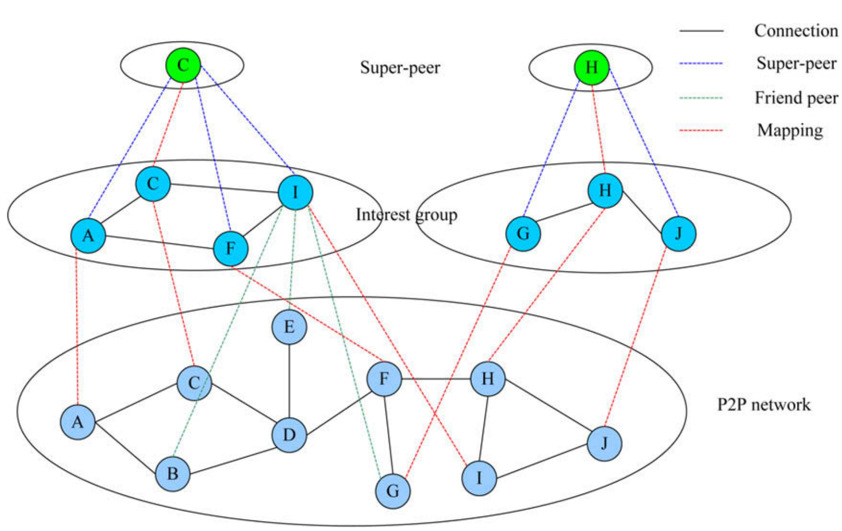

Fig. 9. Semantic-social search mechanism.

Table 7
Experiments results of average hit path length

Search method	Average hit path length
Flooding	3.266
Random super-peer	2.458
Interest cluster	1.914
Our approach	1.796

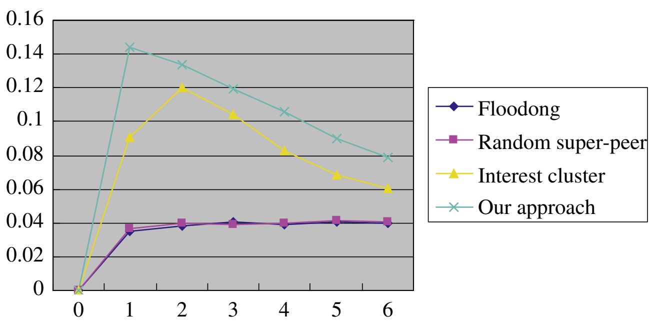

Fig. 10. Precision of search results.

peers is unbalanced, which reflects the actual resource distribution in a real P2P network.

A search scenario starts from a randomly selected peer. We use our social group-based P2P search method (discussed in Section 3.2) to search for results. In our experiments, TTL was set to six, meaning that the search process goes up to six levels deep. We compared our search approach with three other related search algorithms: flooding, random super-peer, and interest cluster. For comparison purpose, TTL in the three methods were also set to six. Since our approach is built on top of super-peer concept, so we also selected the super-peer algorithm for comparison. We randomly chose some peers as super-peers. In interest cluster method, we made peers keep a list of peers that response certain queries, which imply interest clusters. For each of the five P2P topology graphs generated, we generated 2000 queries using each of the four search methods, and calculated its search performance using the measurement approach described in the next section.

5.2. Performance evaluation

For each query using an algorithm, we record its hit path number as its search distance, which is the number of hops (peers) the query travels before it finds a matched resource. For each algorithm, we apply a set of queries and calculate the average path length of search as below, which refers to the average of distances from requesting peers to their corresponding target peers first found with matched resources. This value is used to compare search performance and efficiency of the four search methods.

$$p(a\lg) = \frac{\sum_{\text{query}} \text{hit\_path\_length}}{|\text{query}|}$$

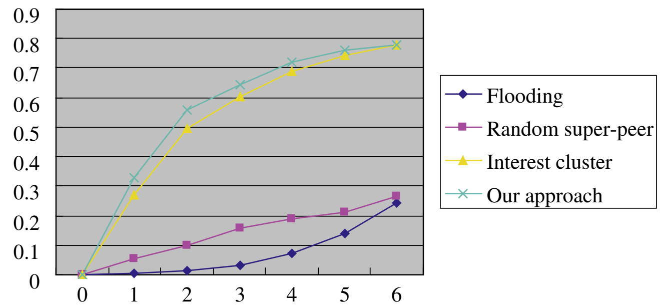

Fig. 11. Recall of search results.

We further measure search accuracy of our search method with that of the other three methods. As shown below, precision refers to the fraction of the retrieved materials that are considered as relevant. Recall refers to the fraction of the relevant materials that are found.

$$\text{Precision} = \frac{|Ra|}{|A|}, \quad \text{Recall} = \frac{|Ra|}{|R|}$$

where A contains a set of peers been reached; $|A|$ is the number of peers in A. R contains a set of peers been found that are considered relevant; $|R|$ is the number of peers in R. Ra contains a set of peers as the intersection of the sets R and A; $|Ra|$ is the number of peers in Ra.

For each search method, we track and record the data for each of the 2000 searches for the five P2P topology graphs. Then we calculate average path length, precision, and recall based on the 10,000 test cases. We repeat the same testing scenarios for all four search methods, and the testing results are summarized and analyzed as below.

Table 7 summarizes the experimental results of the four search methods on average path length. It can be seen that our approach exits the shortest average hit path length and flooding method has the lowest efficiency. With an average of two levels (travel hops), our method finds relevant resources.

Fig. 10 and Table 8 summarize the experimental results on precision. It can be seen that our approach and interest cluster approach exhibit significantly higher precision comparing to flooding and random super-peer approaches. Our approach performs the best. As shown in Fig. 10, the highest precision of our approach is close to 15%, meaning that requesting peer can find relevant resources by querying up to six peers. In contrast to our approach, the precision of flooding and random super-peer approaches are close to 4%, meaning that they had to check many peers and they found few relevant peers.

Fig. 10 also reveals that the highest precision shows at path length one in our approach, meaning that requesting peer can find relevant resources in its preference group with a high probability. We also notice that precision in our approach reduces slowly as the path length increases, which means that number of relevant resources that can be found in other groups is less than that in its preference group. Fig. 11 shows that interest cluster approach shows similar characteristic. However, the difference is that interest cluster approach has poor precision in the beginning of the

Table 8
Experiments results of precision.

Search method\path length	0	1	2	3	4	5	6
Flooding	0	0.034916	0.037907	0.040593	0.03927	0.040467	0.040006
Random super-peer	0	0.036714	0.039935	0.038743	0.039407	0.041658	0.040945
Interest cluster	0	0.091099	0.120108	0.104204	0.082449	0.068644	0.060824
Our approach	0	0.144453	0.133822	0.119098	0.106179	0.089795	0.078937

Table 9
Experiments results of recalls.

Search method\path length	0	1	2	3	4	5	6
Flooding	0	0.004453	0.013531	0.031092	0.070003	0.141222	0.241525
Random super-peer	0	0.053336	0.099764	0.155396	0.189276	0.211388	0.265351
Interest cluster	0	0.268789	0.494891	0.603212	0.688345	0.741458	0.779241
Our approach	0	0.327592	0.558453	0.643042	0.719572	0.758579	0.779345

search process, because it does not have much previous search data in the beginning to help it route to relevant peers.

Fig. 11 and Table 9 summarize the experimental results on recall. It can be seen that the recall of our approach and interest cluster approach increase rapidly when the path length is less than three and increase slowly when the path length is greater than three. The phenomenon shows that these two approaches can find most relevant resources before the length path of three; afterwards, they may query many unnecessary peers. In contrast to these two approaches, the recall of flooding approach increases rapidly after the path length of four. The phenomenon shows that flooding approach can find only few relevant resources before the path length of four; most relevant resources would be found after the path length of four.

Fig. 11 also reveals that our approach and interest cluster approach can find close to 8% of relevant resources and the other two approaches can find only about 25% of relevant resources. Flooding approach performs the worst. From our experiments, we have proved that our approach exhibits effective and efficient P2P search.

6. Conclusions and future research

This research applies the concept and technique of social networks to improve the performance of P2P search. We demonstrate that social networks may help in improving P2P search performance. We present a social network-based search algorithm exploiting the super-peer-based architecture. Specially, we utilize the characters of social networks, such as clustering peers with similar preferences and backgrounds to the same groups and routing through friend peers. To verify that our approach, we simulated a P2P network and our experimental results show that our approach offers effective and efficient P2P search.

The weakness of our current approach is that system performance may become temporarily unstable when new peers join in the network. When a new peer joins the P2P network, it may fill in some questionnaire to provide some initial background information. This information has impact on its original group forming. If the peer provides incorrect information, system performance of our method may decrease. This weakness, however, could be quickly alleviated as time goes by when the peer participates in interactions and obtains credits and feedbacks.

In our future work, we intend to further enhance our search algorithm, especially focusing on semantic aspect. We plan to explore how to analyze search paths and processes to extract data to refine semantic grouping in our method. In addition, we plan to examine peers being useful in the past and seeming to have interest in a given area. We intend to investigate how to use this information to enhance our virtual social network construction and dynamic maintenance.

References

ACM (1998). Computing classification system. Available from: <http://www.acm.org/class/1998/>.

Beydoun, G., Kultchitsky, R., & Manasseh, G. (2007). Evolving semantic web with social navigation. *Expert Systems with Applications, 32*(2), 265–276.

Borch, N. T. (2005). Improving semantic routing efficiency. In *Proceedings of the second IEEE international workshop on hot topics in peer-to-peer systems (HOT-P2P), San Diego, CA, USA, July 21* (pp. 80–86).

Clarke, I., Sandberg, O., Wiley, B., & Hong T. W. (2001). Freenet: A distributed anonymous information storage and retrieval system. In *International workshop on designing privacy enhancing technologies: design issues in anonymity and unobservability* (pp. 46–66).

Cohen, E., Fiat, A., & Kaplan, H. (2003). Associative search in peer to peer networks: Harnessing latent semantics. *The International Journal of Computer and Telecommunications Networking, 51*(8), 1861–1881.

Gkantsidis, C., Mihail, M., & Saberi, A. (2004). Random walks in peer-to-peer networks. In *Proceedings of 23rd annual joint conference of the IEEE computer and communications societies (Infocom), Hongkong, China, March 7–11* (pp. 120–130).

Griffiths, N. (2006). Enhancing peer-to-peer collaboration using trust. *Expert Systems with Applications, 31*(4), 849–858.

Gummadi, K., Gummadi, T., Gribble, S., Ratnasamy, S., Shenker, S., & Stoica, I. (2003). The impact of DHT routing geometry on resilience and proximity. In *Proceedings of the 2003 conference on applications, technologies, architectures, and protocols for computer communications, New York, NY, USA* (pp. 381–394).

Hassan, J., & Jha, S. (2004). Optimizing expanding ring search for multi-hop wireless networks. In *Proceedings of the 47th annual IEEE global telecommunications conference (Globecom), Dallas, TX, USA, November 29–December 3* (pp. 1061–1065).

Hughes, D., Coulson, G., & Walkerdine, J. (2005). Free riding on Gnutella revisited: The bell tolls? *IEEE Distributed Systems Online, 6*(6).

Jesi, G. P., Montresor, A., & Babaoglu, O. (2006). Proximity-aware superpeer overlay topologies. *Lecture Notes in Computer Science, 3996*, 43–57.

Kalogeraki, V., Gunopulos, D., & Zeinalipour-Yazti, D. (2002). A local search mechanism for peer-to-peer networks. In *Proceedings of the 11th international conference on information and knowledge management, McLean, VA, USA* (pp. 300–307).

Kim, J. K., Kim, H. K., & Cho, Y. H. (2008). A user-oriented contents recommendation system in peer-to-peer architecture. *Expert Systems with Applications, 34*(1), 300–312.

Kruskal, J. B., & Wish, M. M. (1978). 07-011, Multidimensional scaling. In *Sage university paper series on quantitative application in the social sciences* (pp. 7–11). Beverly Hills and London: Sage Publications.

Li, Y. M., & Kao, C. P. (2009). TREPPS: A trust-based recommender system for peer production services. *Expert Systems with Applications, 36*(2), 3263–3277.

Li, J., & Vuong, S. (2004). An efficient clustered architecture for P2P networks. In *Proceedings of the 18th IEEE international conference on advanced information networking and applications (AINA), Fukuoka, Japan, March 29–31* (pp. 278–283).

Lv, Q., Cao, P., Cohen, E., Li, K., & Shenker, S. (2002). Search and replication in unstructured peer-to-peer networks. In *Proceedings of the ACM international conference on measurement and modeling of computer systems (SIGMETRICS), Marina Del Rey, CA, USA, June 15–19* (pp. 258–259).

Montresor, A. (2004). A robust protocol for building superpeer overlay topologies. In *Proceedings of the fourth IEEE international conference on peer-to-peer computing (P2P), Zurich, Switzerland, August 25–27* (pp. 202–209).

Niu, C., Wang, J., Shen, R., Shen, L., & Luo, H. (2007). Cooperativeness prediction in P2P networks. *Expert Systems with Applications*. Available online 15 August 2007.

Nottelmann, H., & Fischer, G. (2007). Search and browse services for heterogeneous collections with the peer-to-peer network pepper. *Information Processing and Management: An International Journal, 43*(3), 624–642.

Carrington, P. J., Scott, J., & Wasserman, S. (Eds.). (2005). *Models and methods in social network analysis*. New York, NY, USA: Cambridge University Press.

Ratnasamy, S., Shenker, S., & Stoica, I. (2002). Routing algorithms for DHTs: Some open questions. *Lecture Notes in Computer Science, 2429*, 45–52.

Ripeanu, M., Iamnitchi, A., & Foster, I. (2002). Mapping the Gnutella network. *IEEE Internet Computing, 6*(1), 50–57.

Rowstron, A. I. T., & Druschel, P. (2001). Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science, 2218*, 329–350.

Schell, M. J., & Singh, B. (1997). The reduced monotonic regression method. *Journal of the American Statistical Association, 92*(437), 128–135.

Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., & Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of ACM SIGCOMM conference, San Diege, CA, USA, August 27–31* (pp. 149–160).

Tong, X., Zhang, D., & Yang, Z. (2005). Efficient content location based on interest-cluster in peer-to-peer system. In *Proceedings of IEEE international conference on e-business engineering (ICEBE), Beijing, China, October 18–20* (pp. 324–331).

Tsai, M. J., & Hung, Y. K. (2009). Distributed computing power service coordination based on peer-to-peer grids architecture. *Expert Systems with Applications, 36*(2), 3101–3118.

Tsoumakos, D., & Roussopoulos, N. (2003). A comparison of peer-to-peer search methods. In *Proceedings of sixth international workshop on web and databases (WebDB), San Diego, CA, USA* (pp. 61–66).

Tsoumakos, D., & Roussopoulos, N. (2006). Analysis and comparison of P2P search methods. In *Proceedings of the 1st ACM international conference on scalable information systems (INFOSCALE), Hongkong, China, May 30–June*.

Wang, J. C., & Chiu, C. C. (2008). Recommending trusted online auction sellers using social network analysis. *Expert Systems with Applications, 34*(3), 1666–1679.

Wiesner, C., Kemper, A., & Brandl, S. (2004). Dynamic extensible query processing in super-peer based P2P systems. In *Proceedings of 20th IEEE international conference on data engineering (ICDE), Boston, MA, USA, March 30–April 2* (p. 848).

Yang, W. S., & Dia, J. B. (2008). Discovering cohesive subgroups from social networks for targeted advertising. *Expert Systems with Applications, 34*(3), 2029–2038.

Yang, B., & Garcia-Molina, H. (2002). Improving search in peer-to-peer networks. In *Proceedings of the 22nd IEEE international conference on distributed computing systems (ICDCS), Vienna, Austria, July 2–5* (pp. 5–14).

Yang, B., & Garcia-Molina, H. (2003). Designing a super-peer network. In *Proceedings of the 19th international conference on data engineering (ICDE), Bangalore, India, March 5–8* (pp. 49–60).

Yuan, S. T., & Chen, F. Y. (2007). UbiSrvInt – A context-aware fault-tolerant approach toward wireless P2P service provision. *Expert Systems with Applications, 32*(3), 726–752.