socP2P: P2P Content Discovery Enhancement by considering Social Networks Characteristics

Reza Farahbakhsh[1], Noel Crespi[1], Ángel Cuevas[1], Sraddha Adhikari[1], Mehdi Mani[2], Teerapat Sanguankotchakorn[3]

[1] *Institut Telecom, Telecom SudParis, Evry, France*
{firstname.lastname}@it-sudparis.eu

[2] *ITRON- ITC group*
mehdi.mani@itron.com

[3] *Asian Institute of Technology, Pathumthani, Thailand*
teerapat@ait.ac.th

Abstract— **Content management appears as an essential requirement in order to deploy enhanced P2P networks. In P2P networks, the content that is requested by the query node could be located at different locations/nodes; therefore an efficient search mechanism is required. The proposed search algorithm in this paper, called socP2P, relies on peers' social relationships (friendships, shared interests, shared background and experiences) to improve the content discovery compared to similar solutions. With socP2P nodes can improve searches by using knowledge gained by 'overheard information' during their stay in the network. In addition, our algorithm exploits peers' common interests, friendships, and capability of memorizing experiences (received and routed queries) by them. Simulation results show that socP2P is able to achieve a high success rate, low delay and low overhead. We have verified that our algorithm is not only useful in finding popular contents in the network but also good enough to locate rare files. The obtained results reflect that exploiting social information in P2P networks leads to a more efficient content search mechanism.**

Keywords: Content Discovery, Peer-to-Peer networks, Social Network, Random Walk

I. BACKGROUND AND RELATED WORK

A Peer-to-Peer (P2P) network is a virtual overlay network, built at application layer, on top of a physical network topology. Decentralization and self-organization are key features of P2P networks. AS the number and location of contents in a P2P system have a strong impact on the performance, it is not surprising that content management in P2P networks is one of the research challenges in this area. Peers do not have global knowledge about the network and they have to interact with each other to find the required resource. With the increasing number of peers, the rate of possible interactions may also increase exponentially and hence this strategy is not applicable. In addition, existing unstructured P2P search techniques create huge network traffic or they require high search overhead [4].

Most of the search methods for unstructured P2P networks are focusing on solving their issues by means of blind flooding. Authors in [5] present a new social-like P2P algorithm (Social-P2P) for resource discovery by mimicking human interactions in social networks where peer nodes are people and connections are relationships. Random Walk [6] is one popular searching method in P2P which a query is forwarded to a randomly chosen neighbor hop by hop until enough responses have been found. Also Gnutella [8] appears as one of the most popular P2P file-sharing applications which require nodes to broadcast messages to their neighbors in order to perform content lookup.

Given that recent studies of P2P systems have provided a series of useful solutions to improve the performance of content management in P2P networks, realizing an advanced method of content management by means of social relations of users in OSNs is still an open issue in ongoing research. The objectives, requirements and challenges for P2P social networks have been formulated in [1].

In this paper, we use interest similarity gestures (exhibited by the fact that both nodes are interested in the same content) to establish P2P links rather than relying on random associations or DHT (Distributed Hash Table) rules. The proposed algorithm, named socP2P, uses social relation of peers to improve content management. Finally, we evaluate the performance of the proposed algorithm via simulation in terms of average success rate and path length to desired contents. In addition, we compare socP2P with Random walk algorithm.

The rest of this paper is organized as follows. The proposed resource search algorithm, socP2P, is presented in section II. Section III contains the performance evaluation of the proposed algorithm. Finally, we conclude the paper and present future direction in Section IV.

II. SOCP2P, PROPOSED RESOURCE SEARCH ALGORITHM

A. Proposal Details Overview

As stated earlier, the driving concept in this paper is to extract the social characteristics of the human beings behind a network of peers in a P2P system, and use them to improve the content management. We want to identify nodes with similar interests and enable them to help each other to find desired content. Nodes do not need to declare their interests, because inferring the interest of other nodes in the network is a gradual process for all nodes in the network. The more time they spend in the network the more they learn. Hence, they can route queries towards the right nodes more efficiently. In addition, peers may develop friendship with nodes having similar interest, so peers can generate their friends list and use it to route queries in the network.

In our design, we create new links between peers if we find/detect an interest similarity between them. For example,

978-1-4673-2713-8/12/$31.00 ©2012 IEEE

when node 'A' makes a request for object 'X', and if it gets a successful reply from node B, both nodes learn that they have a matching interest. Our approach differs in detecting interests between nodes compared to [2] that use the concept of providing a strength value to each node in a friend list. Furthermore, a node that generates a query and the node that successfully responds to that query are considered to share similar interest. Thus each interaction between nodes helps to determine if two nodes have interest similarity or not. In addition, each node has the capability of memorizing those received queries even if they are not successful. Therefore, it is also able to recommend a "right node" (i.e. a node that likely stores the searched content).

We have tried to use interest similarity gestures to make P2P links rather than relying on random associations or DHT rules. The links are created in two ways based on interest similarity. In the first method, Directed Interest Links (DIL) is created based on interest similarity between nodes. For simplicity, we set that after nodes A and B learn that they have a matching interest, node A creates an Interest Link with node B. We avoid creating a bidirectional link here, because as it happens in our normal life, relationship links may not always be mutual. For example as OSNs have shown us, some people are very well known and popular in the society, so many people like to create a link with them. However, these kinds of links are not often bidirectional.

The second type of link creation is the Network Suggested Forwards (NSF) and then Created Links model. The above association is the simplest and it is quite limited because it does not exploit the fact that there may be other copies of object X in the network, which node A and node B may not be aware of. The network is usually flooded with queries directed to other nodes for searching purposes. A node may receive queries for the same content more than once. In this case, the node that receives the query has already learned or memorized that some other nodes were looking for the same content earlier and instead of flooding the query in the network or forwarding the query in a random way, the query node can be recommended to get the content directly from the node that was looking for the same content earlier. Moreover, it is now possible to learn that the two nodes share interest profiles and so there is a possibility for them to create an interest-link which otherwise would have not been discovered. A network node keeps records and uses search queries to automatically adapt and make note of objects that indicate related interest relationships.

Fig. 1 shows an NSF model example in which node A is looking for content X. Node A sends query to node E, which memorizes the received query and since node E does not have the content X, it forwards the query to node C. Node C has the content X, thus it sends back the requested content to querying node A. Later in the time, node B looks for the same content X. Then, it sends its query to node E, which has previously encountered the same query. Thus, it recommends node A to node B assuming that node A might have the content X as it was looking for it earlier. Node E also forwards node B's query to node D. Since node D has the content X, it gives back the desired content to node B. The recommended node A also provides the content to node B. In this scenario, node A and node B are interested in same content X. Node C and node D already had the content X. Thus this gives us the clue that all four of these nodes (A, B, C and D) may be interested in the same contents. Thus, Directed Interest Links (DILs) can be created among these four nodes, as shown in Fig. 2.

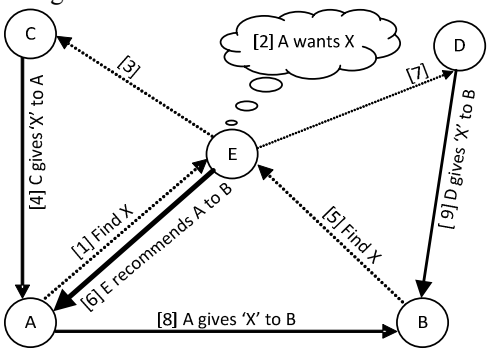

Fig. 1. Network Suggested Forwards and thus Created Links (NSF)

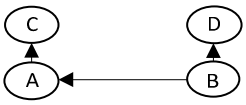

Fig. 2. Directed Interest Links (DIL) created between nodes with interest

B. Three Ways Approach to Search

In our proposed design, we establish a restriction that a query node can send its query directly or via intermediate nodes to a maximum of 'M' nodes. Our objective is to restrict the number of content queries injected in the network for a particular content to M. However, there are two main options for forwarding the queries: in parallel or sequentially. We decide to use a parallel approach since it leads to a better response time [3]. If the query node is unable to find the content in the network (with this restriction of M number of nodes) it will get it directly from the central server. A query node can start its network search in three different ways as shown in figure 3.

The initial method and preferred one is named Recommended Based Search (RBS). The first action of the query node is to check its query record and check if it has already encountered a query for the same content that it is looking for. If it finds that some other node was looking for the same content earlier, it will try to get the content from that node, which we call a Recommended Node. The second way is by making use of useful friends, which is called Friends-Based Search (FBS). Useful friends is a list of the friends of each node that have previously helped that node to find content, either by directly providing the desired content or by forwarding the query to the right node (meaning a node that had the requested content). The third method called Neighbor-Based Search (NBS). If a node doesn't have any recommended node for the looking content and also it does not have any useful friend(s), it will send the query to its neighbors. If the number of Neighbors is greater than M, it randomly chooses M neighbors.

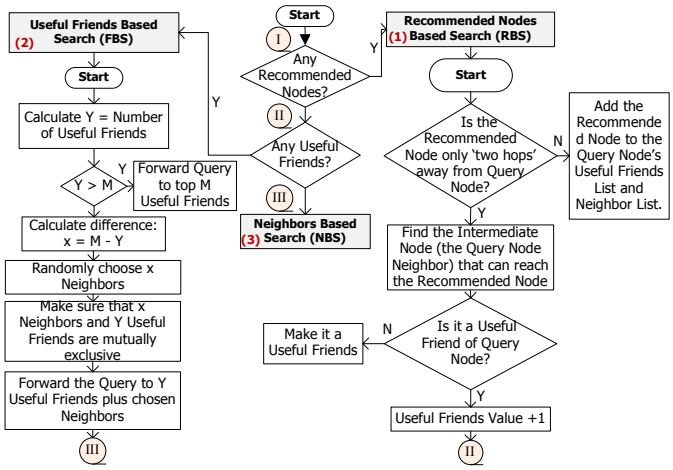

Fig. 3. Three Ways Approach to Search

C. Handling Unnecessary Traffic

In the example of Fig. 4, node S sends a query to nodes L, M and P. Nodes L and P are unaware that M has already received the query from node S. Therefore, they forward the query to M again. Similarly, neither L nor M, are aware that they have already received the query from node S, thus they both forward the query to each other. Therefore, M could receive the same message up to four times which we consider as unnecessary traffic in the network. It is clear that eliminating the logical links LM, MP, LQ and MQ does not reduce the search scope of node S. In our algorithm, when S forwards a query to M, it also appends information about the total list of nodes that the query has been forwarded to. Thus, when node L receives a query from node S, it also gets the information. Thus L does not forward the same query again. This technique has two advantages: (i) it avoids to forward queries again and again to the same node, thus eliminating possible loops and unnecessary traffic from the network; (ii) it increases searching area by forwarding queries to new nodes.

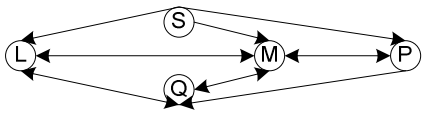

Fig. 4. An Example of P2P Overlay [7]

D. Discovering New Friends

P2P systems are dynamic, which means that peers join and leave the system and also their interest changes over the time. Given this dynamism, nodes always have a need to discover new friends. In our model, nodes add a new node in its friend list if this node successfully replies to a node's query. This update is realized because there is an interest similarity between the two nodes. However, there is no guarantee that a friend will be always useful to the node since (i) the interest of a node or its friend might change, and (ii) some content might be so popular that it is stored by a major portion of the nodes, thus a query for this content could be replied successfully by many nodes. In our model, a rank (or weight) is provided to each friend based on their usefulness. The rank of a friend is increased every time they prove to be useful. To maintain a fix maximum number of friends, the friends that become the least useful based on their ranking are removed from the list when a new useful friend is found.

III. PERFORMANCE EVALUATION

In this section we present the simulation methodology as well as the performance evaluation of socP2P algorithm. In addition, we also present a summary that shows how socP2P compares with the Random Walk algorithm and Interest Based Shortcuts [9].

A. Simulation Methodology

We create a random topology by means of random bidirectional links between peers. Initially peers don't have information regarding to interest of others, thus they keep an empty Friend List. In addition, initially peers present an empty query record, since nodes have not encountered any queries.

The number of resources in the network is denoted by F. Initially, content/resources are distributed randomly in the network based on a popularity parameter of each resource. These popularity parameters are unique and randomly assigned at the beginning. The queries are generated sequentially by random nodes. Then, a query node randomly chooses one of the contents as its query resource (the content that query node is looking for). Each query includes: information about the query node and the query resource, a TTL value of M. Then a query is forwarded to M/2 nodes at first. These M/2 nodes could be useful friends or neighbors. These nodes could forward the query to one neighbor or Useful Friend of theirs. Thus in total, query could be forwarded to M (1 x M/2 + M/2 x 1) nodes. The Success Rate is the rate at which our algorithm resolves queries. In other words, it is how often a query node has found the content it is looking for in the network. The success rate is calculated using the following formula in our algorithm:

$$\text{Success Rate} = \frac{\text{Success NBS} + \text{Success FBS} + \text{Success RBS}}{\text{TQ}} \times 100\%$$

B. Results and discussion

The plot in fig. 5 shows that ASR is increasing with increasing value of M. From this analysis, we can determine a suitable value of M according to our success rate requirements.

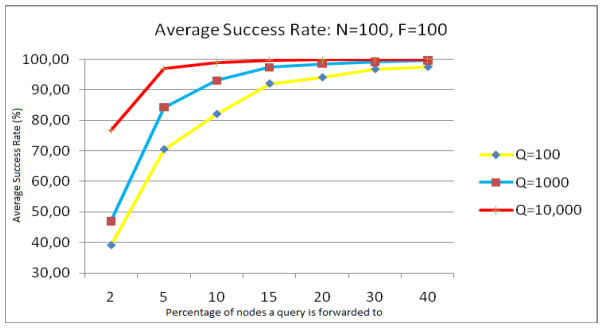

Fig. 5. ASR for N=100, F=100 and different values of Q

As indicated in Fig. 6, if the number of resources (F) increases in the network, the ASR tends to decrease slightly. However, for higher values of M, the effect is not significant.

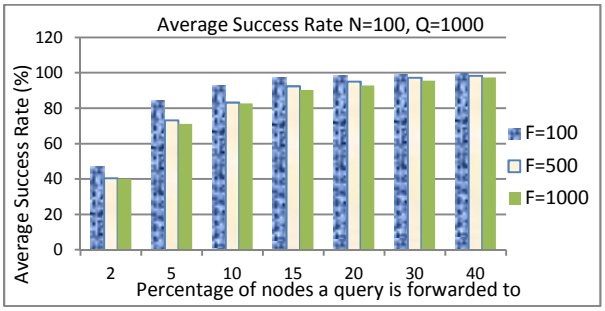

Fig. 6. ASR for N=100, Q=1000 and different values of F

Fig. 7 shows the average success rate with respect to the total number of valid queries in the network (Q).

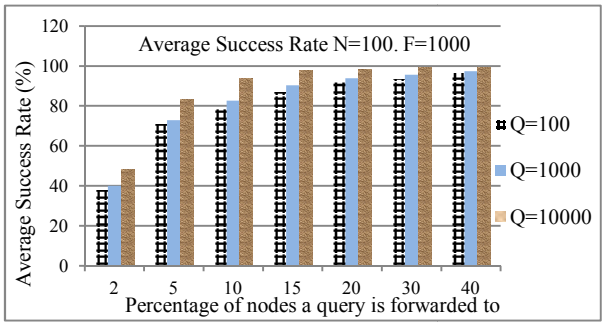

Fig. 7. Average Success Rates for N=100, F=1000 and Q=100, 1000, 10,000

C. *Comparison of socP2P and Random Walk*

Results in figure 8 show that socP2P has the highest success rate for all values of M. The improvement shown by socP2P is especially significant for lower values of M (i.e. 2%), verifying that socP2P can achieve a high success rate without flooding the network with query messages.

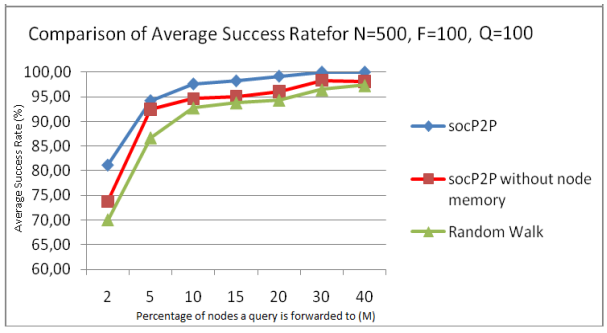

Fig. 8. Comparison of Average Success Rate for N=500, F=100, Q=10,000

Fig. 9 depicts the Average Success Rate versus the Popularity of a file.

Finally, we also found that the average path length (in terms of overlay hops) across several simulated scenarios is 3.5 hops in case of using Gnutella, whereas it is reduced to only 1.27 hops away in case of Gnutella with shortcuts. However, socP2P further improves both results by reducing the path length to only 1.08 hops.

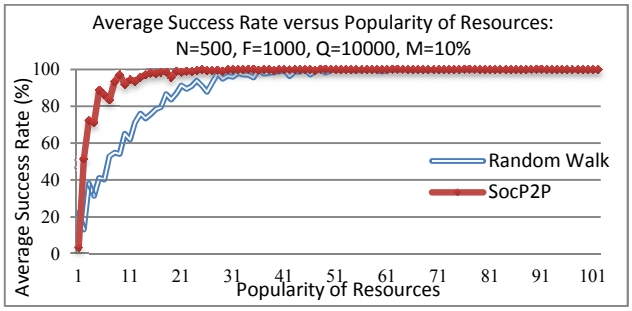

Fig. 9. ASR vs. Popularity of Resources for Random walk and socP2P

IV. Conclusion and Future Work

In this paper a new search algorithm to efficiently locate or discover required content in P2P networks, named socP2P, is proposed. It exploits the properties of social networks and interest links are created between nodes that exhibit similar interest. The approach also exploits the option of learning from received and routed queries keeping a friend list so that queries can be routed intelligently towards them. The simulation results evaluate the performance of the proposed algorithm and demonstrate that socP2P improves previous well-known search mechanisms. It achieves a higher success rate than Random Walk and at the same time, it generates low message overhead. Finally, socP2P is able to efficiently locate not only the popular content, but also the rare ones.

Many possible directions for future research beckon based on this advance. Creating mutual interest links between nodes that have interest similarity, increasing the search success rates of less-popular files, include overhearing part in a forwarded node query and also reducing the search range from two hops to one are some of our possible future endeavors.

Acknowledgment

This work has been supported by the EU ITEA-2 project 10028 “Web-of-Object” (WoO) founded by DGCIS.

References

[1] S. Buchegger and A. Datta. “A case for p2p infrastructure for social networks - opportunities & challenges”, WONS '09, 2009.

[2] Y. Upadrashta, J. Vassileva, W. Grassmann, “Social Networks in Peer-to-Peer Systems”, Proceedings of the 38th Annual Hawaii International Conference on System Sciences, HICSS'05, 2005.

[3] A. Crespo, H. Garcia-Molina, "Routing indices for peer-to-peer systems", 22nd ICDCS, 2002.

[4] Lu Liu, N. Antonopoulos, Jie Xu, and D. Russell. “Investigation of research towards efficient social peer-to-peer networks”, in Computing in the Global Information Technology, ICCGI ’08, 2008.

[5] L. Liu, N. Antonopoulos, and S. Mackin. “Social peer-to-peer for resource discovery”, 15th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP'07. 2007.

[6] N. Bisnik, A. Abouzeid, "Modeling and Analysis of Random Walk Search Algorithms in P2P Networks," Second International Workshop on Hot Topics in Peer-to-Peer Systems, 2005.

[7] L. Xiao, Y. Liu, & L.M. Ni, “Improving Unstructured Peer-to-Peer Systems by Adaptive Connection Establishment”, *IEEE Transactions on Computers,* 2005.

[8] M. Ripeanu, "Peer-to-Peer Architecture Case Study: Gnutella Network", First International Conference on Peer-to-Peer Computing (P2P'01), 2001.

[9] K. Sripanidkulchai, B. Maggs, H. Zhang, “Efficient content location using interest-based locality in peer-to-peer systems,” INFOCOM'03.