Improved P2P Content Discovery by Exploiting User Social Patterns

Reza Farahbakhsh[1], Noel Crespi[1], Ángel Cuevas[1], Neetya Shrestha[1], Mehdi Mani[2], Poompat Saengudomlert[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*
poompats@ait.ac.th

Abstract— **One of the most challenging issues in Unstructured Peer-to-Peer (P2P) network is to efficiently locate information resources. In this paper we propose a novel concept in which peers exploit social relations in order to improve the search success rate. In social networks, people can directly contact acquaintances that have knowledge about the resources they are looking for. However, peers lack these capabilities in a P2P network. We can find different searching mechanism such as flooding, which drastically increases the communication overhead, or random walk (RW) that reduces the message overhead, but since it is a blind sequential search it may take a long time to route a query. In front of this we have proposed a two-hop algorithm that incorporates the social behaviors of peers and processes queries more efficiently [2]. However, there is a strict limitation that the recommended nodes must always have the query resources. In this paper, we propose a one-hop algorithm that uses social behavior patterns. In the proposed algorithm peers establish friendship relations and learn from past experiences to recommend suitable peers that will route queries in an efficient manner. The simulation results show that the proposed one-hop algorithm provides better average success rates compared to both the random walk algorithm and the two-hop algorithm by reducing the search to only one logical hop. In addition, our proposal minimizes the required network memory space by limiting the query record, useful friends and resources at each node in the one-hop algorithm.**

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

I. INTRODUCTION

P2P systems have become very popular over the past decade. Several applications make use of P2P principles, among these are some of the most widely-used applications on the Internet, e.g. Skype, BitTorrent, etc. However, many challenging issues remain to be addressed in P2P networks. One of the most challenging problems is to develop efficient resource discovery algorithms [4, 6], as there is no central entity from which information about resources can be obtained. Several studies have attempted to address this issue in P2P networks, as it promises multiple rewards [1, 6, and 3].

This paper proposes to incorporate social information within the peers to achieve enhanced content discovery algorithms. In particular, we propose a one-hop algorithm to improve searching for resources in unstructured P2P networks, and compare it with the two-hop algorithm [2] and with the random walk algorithm presented in [10].

The rest of this paper is organized as follows. The next section provides the background and some related work. The proposed resource search algorithm is presented in section III, followed by its performance evaluation and comparison with two other algorithms in section IV. We end the paper with conclusion and recommendations for future work in Section V.

II. BACKGROUND AND RELATED WORK

Flooding is a traditional resource location method in unstructured P2P networks. For instance in Gnutella, a peer searching for a resource floods a query to all its neighbouring peers, who in turn will forward it to their neighbours until the query has travelled a certain radius. It is a simple and robust method, even with peers joining and leaving the system, but it is not scalable [6]. Random walk, in which only few query messages are generated, is an improvement over the flooding technique [7].

The social-P2P algorithm, introduced in [7] for resource discovery, is based on using the social behaviours of people in social networks. In this algorithm, peer nodes with the same interests will be connected to new peers as peers learn from past experiences. The interest-based shortcut is a search algorithm based on the principle of interest-based locality, which states that peers with similar interests are likely to have resources that will be needed by each other [3]. The interest-based shortcuts algorithm improves the basic search algorithm, significantly reducing the amount of network traffic compared to Gnutella. Authors in [2] propose a two-hop resource discovery algorithm by using people's social patterns. In this algorithm, directed interest links are created between peers with similar interests. The peer that requests a resource and the peer that replies to the query learn that they have a similar interest and thus a directed link is created from the former to a latter. Peers develop friendships with peers having similar interests. If a peer does not have the requested resource, it keeps a record of the query in its memory. Learning the interests of other peers in the network is a gradual process.

978-1-4673-5288-8/13/$31.00 ©2013 IEEE

There is still room to further improve these search algorithms. For example, decreasing the number of search hops, minimizing the required network storage space and enhancing network scalability are just three of the issues that motivate us to work towards the improvement of search algorithms in P2P networks.

III. Proposed Resource Search Algorithm

Studying existing search algorithms motivated us to work towards optimizing two areas: reduce query path length to find a resource and minimize memory space utilization, Therefore, the final objective of this work is to improve the resource search mechanisms in P2P networks to achieve a high success rate with low overhead, reducing delay in the network and minimizing the required memory space.

This paper proposes a one-hop algorithm that takes advantage of social patterns to improve resource searches in unstructured P2P networks. Each peer develops friendships with peers with which it shares similar interests, increasing the useful value of a friend/peer each time that peer provides a resource. The peers learn from past experiences to route their queries more and more efficiently. The longer a peer stays in the network, the more queries it will receive and route, thus more network knowledge it will captures, and thus, it will be able to better help others by recommending those peers that had looked for the same resources, thereby reducing the network traffic and increasing the average success rate.

A one-hop algorithm implies that, in order to search a resource, a query is transmitted to nodes that are one hop away. One hop refers to one logical hop, which means a query may have to be transmitted across multiple physical nodes. In the two-hop and random walk algorithms a query can be transmitted to TM nodes, being TN is the total number of nodes in the network and TM<TN. The Average Success Rate (ASR) of a query is greater in the two-hop algorithm than the random walk algorithm [10]. In the two-hop algorithm, a query is first transmitted to TM/2 selected neighbours and/or useful friends, and each of them can forward the query to one of its neighbours and/or useful friends, as shown in Figure 1.a. In the random walk algorithm, a query is transmitted to TM nodes sequentially as shown in Figure 1.(b).

The concept behind the proposed one-hop algorithm is to modify these two search mechanisms to obtain a purely parallel search, as shown in Figure 1.(c), reducing the path length from query-to-resource to one logical hop and thereby improving the search time and network traffic compared to the two-hop and the random walk algorithms.

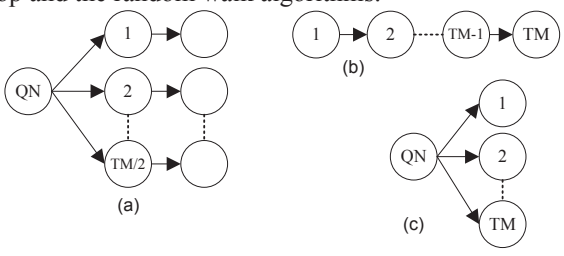

Fig. 1. (a): Two-hop algorithm [2], (b): Random walk algorithm [10] (c): One-hop algorithm

In our approach, Ni refers to node i where i is an integer and can take any value from 1 to TN, Rj is used to name resource j where j can take any value from 1 to TR (total number of resources) and Qk refers to query k that can take any value from 1 to TQ (total queries in the network). To illustrate the algorithm operations we consider the following sequence of events. Initially, the useful friend list and the query record of each node are empty, as shown in figure 2.

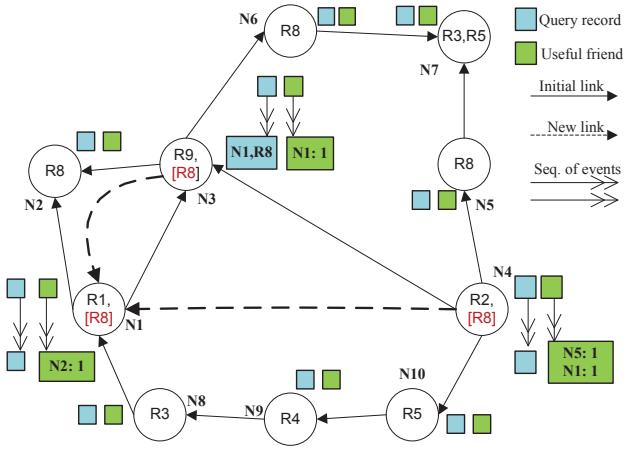

Fig. 2. Resource distribution in initial network and updated UF and QRs

Let us consider TM=2 and the following queries {1}: Q1 (N1, R8), {2}: Q2 (N4, R8), {3}: Q3 (N3, R8). In query {1}, N1 is searching for R8 and it sends query messages to N2 and N3. N1 gets R8 from N2; its resource is updated such that N1 now has R1 and R8. N1 also updates its useful friends list where it gives the useful value of 1 to N2. At the same time, N3 does not have R8, so it updates its query record that N1 was looking for R8. In query {2}, N4 is searching for R8. As shown in Figure 2, its useful friends list and its query record are empty, and it has only resource, R2. N4 can send query messages to two nodes, which it randomly selects from N3, N5 and N10. We assume N3 and N5 are selected. N4 gets R8 from N5, adds R8 to its resource list and updates its useful friends list, giving N5 the value of 1. At the same time, N3 responds that N1 had previously looked for R8. So, N4 establishes a logical link with N1 since it is more than one hop away and receives R8 from N1. N4 also makes N1 a useful friend since it received R8 from N1. N4 receives R8 from both N5 and N1. However, N4 saves only one copy of R8 in its resource list. In query {3}, N3 is looking for R8. It finds from its query record the recommended node (RNs) of N1 then establishes a link with N1 as a recommended node for N3 and gets R8 from N1. It updates its resource and useful friend (UF) lists as shown in figure 2.

The flowchart of the one-hop algorithm is shown in figure 3. When a query node (QN) initiates a query, it can follow three different search mechanisms: Recommended Node Based Search (RBS), Useful Friend Based Search (UBS) and Neighbour Based Search (NBS). A query node initially begins a search using RBS. If it has a record in its query record that shows that the resource it is searching for had been searched for by some other nodes, it receives the resource from one of those nodes, i.e. the recommended nodes. If there is no

recommended node in its query record, it starts the UBS. If a node does not receive the resource from either an RBS or a UBS, it sends the query to TM neighbours to begin NBS. We assume there is a central server that contains all the resources in the network. It is only used if the query node cannot receive the query resource from the network; to ensure that the query node always gets the query resource in our search algorithm.

IV. Performance Evaluation

A. Simulation Methodology

In our algorithm, a random network is generated with unidirectional links between the nodes. Initially, these nodes have no information about other nodes. As the queries in the search algorithm are generated and forwarded, each node learns who its useful friends are. Each query is generated at random by selecting a query node randomly from all TN nodes. The corresponding query resource is also selected randomly from all TR resources in the network. Each query is tagged with the value of TM, which means that the query can be forwarded to up to TM nodes.

We consider two different types of initial resource distribution. In case I, we set as a heterogeneous distribution of resources where each resource can be with a variable number of nodes. Each resource is given a value chosen randomly between 1 and TN/2. The resources are distributed at random among the nodes in the network according to their chosen value. The nodes are chosen randomly and uniformly for this resource distribution. In case II, to see how our algorithm works for homogeneous resource distribution, at the start of each test, we randomly and uniformly distribute each resource to 10 nodes. The Success Rate is defined as the number of times the queries were resolved with the one-hop algorithm via RBS, UBS or NBS, without the need of the central server, out of the total number of queries (TQ) in the network. Mathematically, the success rate is evaluated as follows. The higher the success rate, the better the performance of the one-hop search algorithm.

$$\text{ASR} = \frac{\text{Sum of Success Rate from a number of simulations}}{\text{Total number of simulations}}$$

The Average Success Rate (ASR) is the average success rate obtained from several simulated scenarios with the same parameters.

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

The average path to the resource is the number of overlay hops that a query has to travel to find its resource. It is directly proportional to the time required to locate the required resource in the network. The smaller the average path length to the resource is, the lower the delay to locate that resource. The average number of messages sent and received per node is obtained by summing the total messages received and sent by each node for the total of TQ queries in the network and dividing by the network size TN. The lower the average number of messages sent and received per node, the lower the network traffic. Also the probability that there is a unidirectional link from one node to another in the network is named "p".

A. Results and discussion

We compare the performance of the proposed one-hop algorithm with the random walk [10] and the two-hop algorithm [2]. For initial resource distribution, in case I each resource is given a random value between 1 to TN/2 and distributed randomly to this number of the nodes in the network. And in case II, each resource is distributed to 10 nodes randomly in the beginning. Finally, the probability that there is a unidirectional link from one node to another in the network named p. When the value of TM is increased, a query can be forwarded to a greater number of nodes. Therefore, the ASR increases with the TM. For p=0.1, we observe that our proposed one-hop algorithm is comparable and better than the two-hop algorithm and the random walk algorithm as shown in figure 4, because more information is stored in the one-hop algorithm as queries.

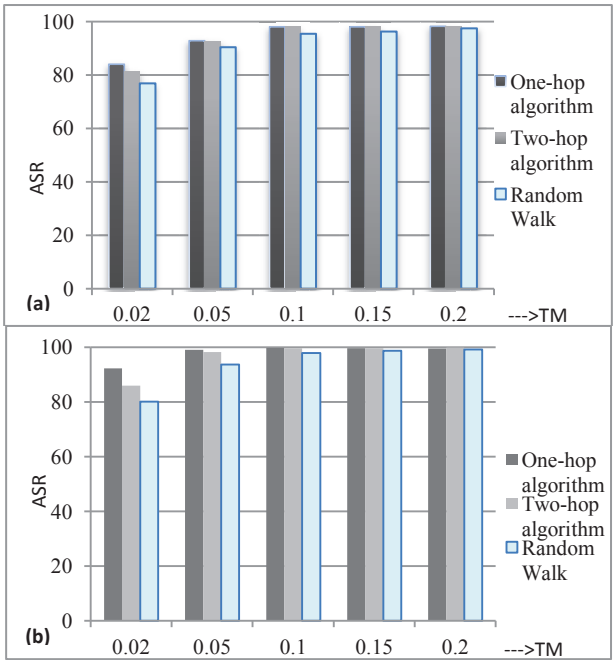

Fig. 3. ASR and resource distribution (a): Case I, (b): Case II

Fig. 5 shows how the average path length to resource varies in the three algorithms. When the value of TM is increased for TN=500, TR=100, TQ=1000 and p=0.1, a query can be forwarded to nodes at further number of hops in the random walk algorithm, increasing the average path lengths to resources. In the two-hop algorithm, a resource may be found within the first or second hop. Therefore, the average path length to a resource is between 1 and 2. In the one-hop algorithm, a query is forwarded only within one hop and a new link is established if the recommended node is more than one hop away. Therefore, the resource is always located at the node in one hop and the average path length to a resource is equal to 1. For resource distribution in case I, there are more resources available. Therefore, a resource can be located in fewer hops when compared to case II.

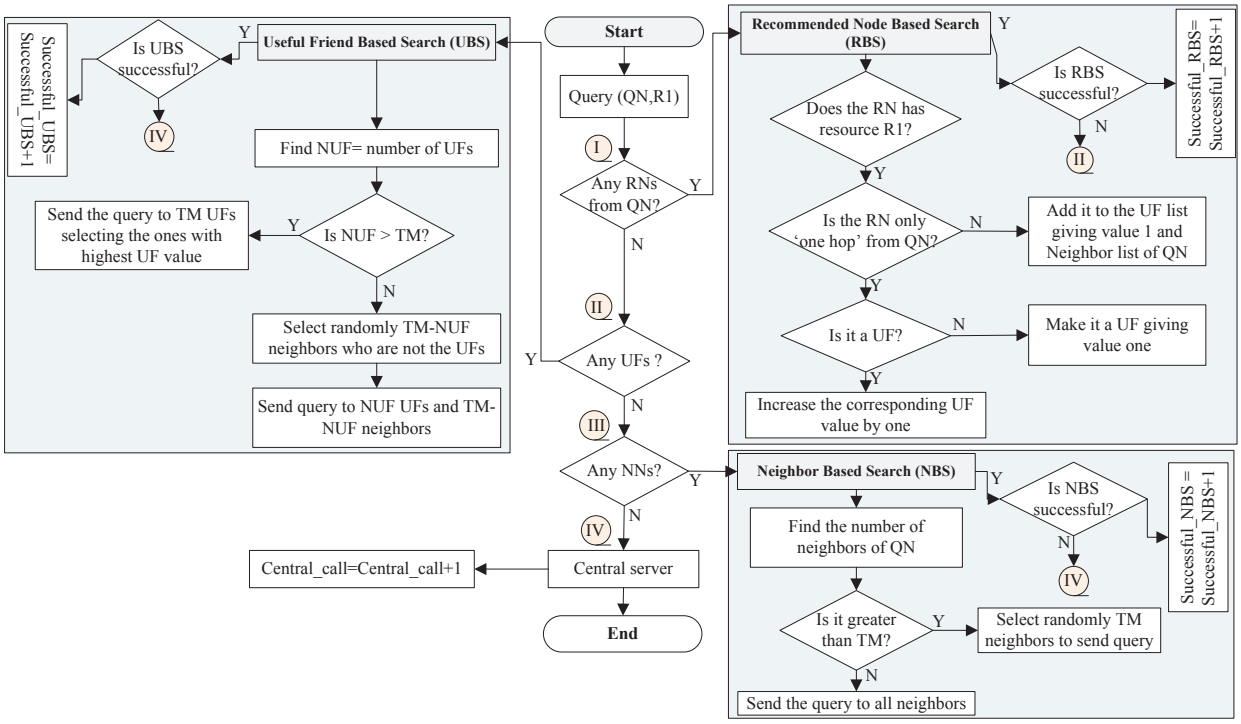

Fig. 4. One-hop algorithm for resource discovery in P2P networks

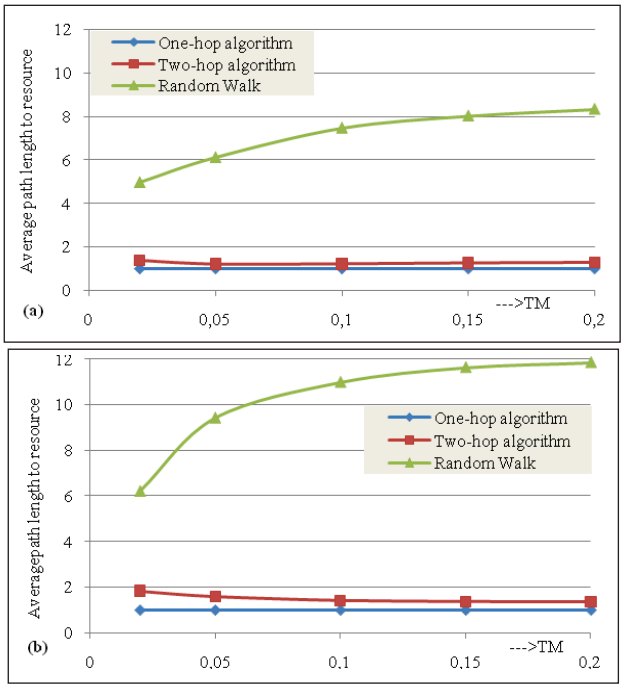

Fig. 5. Average path length to resource (a): Case I, (b): Case II

Fig. 6 shows the average number of messages sent and received per node for all the three algorithms. When TM is increased for TN=500, TR=100, TQ=1000 and p=0.1, the average number of messages per node increases in the random walk algorithm as it can forward the query to more nodes. The average number of messages sent and received per node saturates due to the average node degree in the two-hop algorithm and the one-hop algorithm. For resource distribution of case I, the random walk finds resource in fewer hops, and therefore few nodes participate in the search. Consequently, the average number of messages per node is smaller.

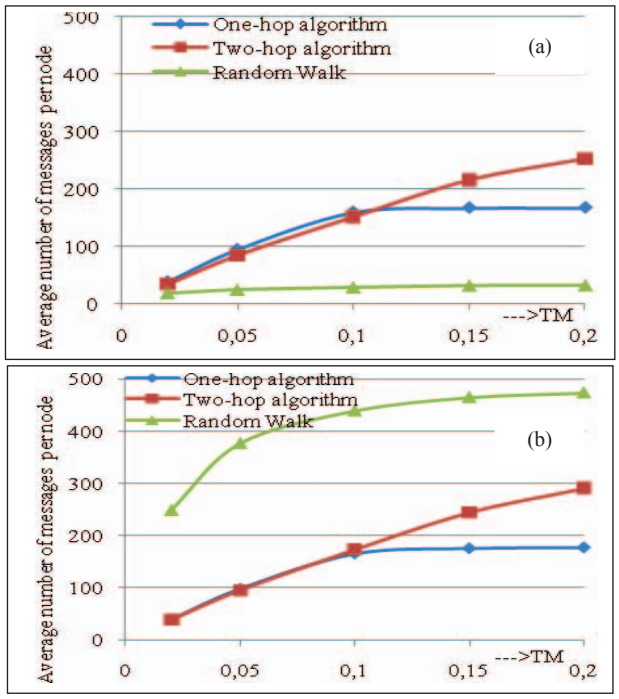

Fig. 6. Average number of messages Sent & Received (a): Case I, (b): Case II

When the network size in a P2P network increases, we have addition in the number of peers participating in the search. The peers not only consume resources but also provide

resources. So the ASR should increase with an increase in the network size. For TM=0.02TN as TN is increased from 100 to 500, the value of TM increases from 10 to 20, 30, 40 and 50. At p=0.1, the average node degree is 50. For TR=100 and TQ=1000, as TN increases, TM increases and a query can be forwarded to more nodes. Therefore, the ASR increases for all the three algorithms as shown in Figure 7.

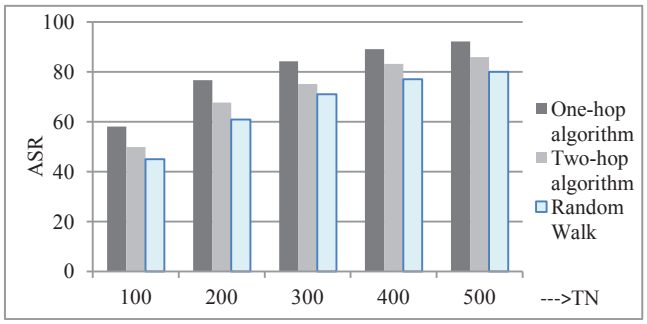

Fig. 7. ASR for TM=0.02TN, p=0.1 and resource distribution in Case I

As the value of TQ increases (size of the query) the useful friends and the resource lists increase in the network and occupy more storage space resulting in an increase in processing time for each query. As Fig. 8 shows, the size of the query record can be limited to 10 a number at which we can achieve the same ASR when the query record is unlimited.

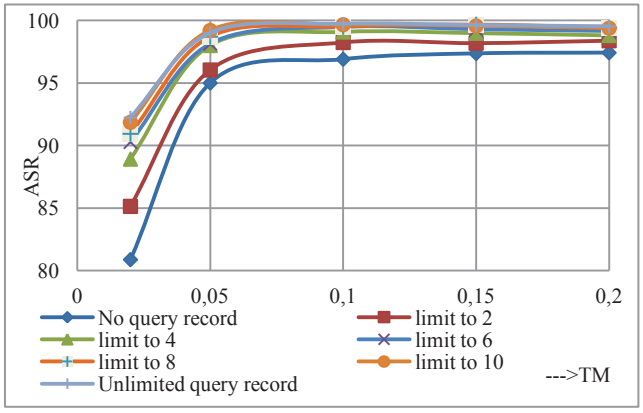

Fig. 8. ASR for TN=500, TR=100, TQ=1000 and p=0.1 with query limitation

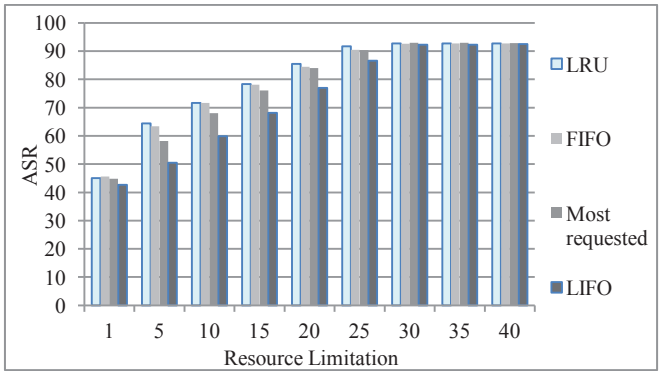

Fig. 9. ASR for TN=500, TM=0.02TN, TR=100, TQ=1000 and p=0.1 with resource limitation

For resource limitation, we use different methods: LRU (Least-Recently Used), FIFO (First-In-First-Out), most-requested and LIFO. We limit the resources to different values to select the smallest values that can give the same ASR as when the number of resources is unlimited. In LRU, to store a new resource, the resource that was the least-recently requested is removed. FIFO and LIFO remove the resource that was stored first and last, respectively. The most-requested method removes the resource that was most requested by its neighbours. Fig. 9 shows that LRU and FIFO are comparatively better than the other two methods and that we can limit the number of resources to 30 to achieve the same ASR as when the number of resources is unlimited.

V. Conclusion and Future Work

In this paper, we have proposed a one-hop algorithm to locate the resources in P2P networks that exploits social behaviour patterns where peers develop friendships with others who have similar interests. Our solution limits the queries to the peers within one logical hop and also minimized the memory space of the network by limiting the query record, useful friends and resources. Simulation results show that the average success rate is higher in compare to other similar algorithms (Random Walk and Two hop algorithm). In addition, the average path length to resources is also the lower which is directly related to the time required to locate the resource or network delay. Furthermore, the average number of messages sent and received per node in the one-hop algorithm is less than the same as in the two-hop algorithm. Finally, the query record, useful friends and resources can be limited to suitable values, reducing the required memory. LRU and FIFO policies provide better average success rates with resource limitation situations in our search algorithm.

Acknowledgment

This work has been supported by the EU ITEA-2 Project TWIRL (Twinning Virtual World Information with Real World Data Sources).

References

[1] S. Buchegger and A. Datta, “A case for P2P infrastructure for social networks - opportunities & challenges”, WONS'09, Feb. 2009.

[2] R. Farahbakhsh, N. Crespi, A. Cuevas, S. Adhikari, M. Mani , “socP2P: P2P Content Discovery Enhancement by considering Social Networks Characteristics”, IEEE ISCC’12, Turkey, 2012.

[3] K. Sripanidkulchai, B. Maggs, “Efficient content location using interest-based locality in peer-to-peer systems,” INFOCOM'03, 2003.

[4] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, “A Survey and Comparison of Peer-to-Peer Overlay Network Schemes”, *IEEE Communications Surveys & Tutorials*, Vol. 7, Issue: 2, P: 72 – 93.

[5] Y. Liu, L. Xiao, X. Liu, L.M. Ni, and X. Zhang, “Location Awareness in Unstructured Peer-to-Peer Systems”, *IEEE Transactions on Parallel and Distributed Systems,* Vol. 16, Issue: 2, 2005.

[6] W. Jianyong, L. Yuling, G. Futing, “A New Strategy of Resource Searching in Unstructured P2P Network”, ICCSN'10, 2010.

[7] L. Liu, N. Antonopoulos, and S. Mackin, “Social Peer-to-Peer for Resource Discovery”, PDP'07, 2007.

[8] Dongyu Qiu and R. Srikant, “Modeling and performance analysis of BitTorrent-like peer-to-peer networks”, SIGCOMM’ 04, August 2004.

[9] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, “Making gnutella-like P2P systems scalable”, SIGCOMM'03, 2003.

[10] 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.