

LINK RECOMMENDATION IN P2P SOCIAL NETWORKS

Yusuf Aytas
Dept. of Computer Engineering
Bilkent University
yusuf.aytas@bilkent.edu.tr

Hakan Ferhatosmanoglu
Dept. of Computer Engineering
Bilkent University
hakan@cs.bilkent.edu.tr

Özgür Ulusoy
Dept. of Computer Engineering
Bilkent University
oulusoy@cs.bilkent.edu.tr

ABSTRACT

Social networks have been mostly based on a centralized infrastructure where the owner hosts all the data and services. This model of “fat server & thin clients” results in many systems and practical problems such as privacy, censorship, scalability, and fault-tolerance. While a P2P infrastructure would be a natural alternative for implementing social networks, it has surprisingly not attracted enough attention yet. Significant research is needed to develop a P2P social network system. From an algorithmic perspective, most graph algorithms for social networks assume that the global graph is available. These need to be revisited in a P2P setting where the nodes have limited information with connectivity to only their neighbors. Following these observations, in this paper, we focus on social network link recommendation problem in a P2P setting. We investigate methods to recommend links to improve social connections as well as the efficiency of the overlay network. We evaluate our methods with respect to measures developed for P2P social networks.

Categories and Subject Descriptors

C.2.4 [Distributed Systems]: J.4 [Social and Behavioral Sciences]:

General Terms

Algorithms, Management, Measurement, Performance, Design, Experimentation, Security, Social Network.

Keywords

Social Network, P2P Network, Link Prediction, Measurement.

1. INTRODUCTION

Online social networks have drawn attention in the last decade with growing number of people using social platforms like Facebook, Twitter, LinkedIn and so on. Social network providers offer variety of services, which results in rich content and linkage data. The data has been typically administered by a single owner. However, this setup is counter-productive in both systems and practical perspectives. Users do not have the power to safeguard themselves from misuse of their data [1]. The owners of social networks can apply censorships and other exercises of central authority [2]. These problems arise mainly because of the classical paradigm of thin clients and a fat server. P2P systems can be an alternative to the client-server model for social networks, where data is maintained in all peers and the users define their level of privacy.

From a systems perspective, a P2P implementation of social networks is expected to scale better. Social networking companies make investments to handle scalability problems employing large data centers. The consumption of resources of energy and time may not be necessary if a P2P implementation could be made possible.

P2P social networks introduce significant challenges in terms of P2P storage, topology and local graph algorithms [1]. The connections and sharing in a P2P social network need to be a continuous process, unlike a typical P2P setting where the connections are constructed based on a file being searched at the time. Most graph models and algorithms for social networks in the literature assume that the global network is available at hand, hence only the owner can truly employ them. This assumption is reasonable in a single administration setting, however needs to be revisited for a distributed infrastructure.

In this paper, we address the problem of link prediction in a P2P social network. The P2P link prediction problem differs due to the distributed nature, where each node has only local information about the network. We propose an approach by adapting a top-k search algorithm to P2P link recommendation. We also present scoring methodologies for nodes in a P2P social network, considering accuracy and efficiency of the underlying topology.

2. BACKGROUND

2.1 Social Networks and Link Prediction

Centrally administered social networks made a significant impact on business and social life. From computer science perspective, they introduced problems such as community detection, influence analysis, ranking, node classification, and link prediction [3]. These problems can be revisited in a P2P setting. In this paper, we focus on link prediction as an essential problem. The link prediction problem was introduced by Nowell and Kleinberg as estimating new interactions between the nodes of a social network [4]. Methods for link prediction rely on content shared among nodes and topology of the network. Latter is preferred especially due to privacy issues. Topological methods are based on paths between nodes and neighborhoods [4]. Paths between nodes approach uses shortest path, ensemble of paths or their variants to handle the link prediction problem. Likewise, Bakstrom and Leskovec use network structure and node/edge attributes to predict new interaction by the help of random walks [5].

Neighborhood approaches, such as Common Neighbors, are also used in link prediction. For example, Adamic and Adar use weighted neighborhood information to find relationship between individuals [6]. Intuition behind Common Neighbor is that a node is more likely to interact with another node if the number of overlaps between their neighbors is high. It is a simple heuristic that can often outperform complex heuristics [7].

While most current link prediction methods work on a global network, we aim to build a localized algorithm in a P2P setting.

2.2 P2P Infrastructures

P2P systems enable sharing data and resources between the peers. File sharing applications are best-known realization of P2P systems. Some successful applications are Napster, Gnutella and BitTorrent. P2P frameworks can also be used for distributed computation, collaboration and communication between peers [8].

P2P systems can be categorized according to the degree of centralization: purely decentralized, partially centralized and hybrid decentralized [8]. In purely decentralized systems resources are shared among all peers uniformly. In partially centralized systems, super peers handle coordination of resources. Lastly, hybrid systems have a central server that keeps track of indexes. Hybrid systems offer an effective tradeoff for implementing P2P social networks, since fully decentralized systems have disadvantages like recalling peer addresses. In a case where all peers are offline, all information can be lost in a decentralized system while we at least expect one super peer can take care of peer addresses. DNS like protocols can be easily implemented in hybrid system using the power of super peers.

In a P2P environment, the link prediction problem becomes different since we do not have control on the full network. Each node has partial information about the network. Thus, there is a need for distributed algorithms to find predicted links in P2P infrastructure. One can utilize a Common Neighbor approach to locally gather and merge link strengths from neighbors. We approach this merging problem as a top-k query processing [9] and implement a set of distributed top-k link recommendation algorithm.

2.3 Related Work

The closest work to our study is the CNP (Common Neighbor Predictor) method proposed by Zhang et al. [10]. While that work does not consider a social network and does not consider P2P performance issues, it uses a distributed algorithm to predict future links in a P2P infrastructure. They express their ground truth through the following formula:

$$\alpha = \frac{|E_{pre} \cap E_2|}{|E_{pre}|}$$

where E_{pre} is the set of predicted links by the algorithm between two time instants t_1 and t_2 . Furthermore, E_2 is the set of links that are newly created by the network between times t_1 and t_2 . First, NCNP (Neighbor's Common Neighbor Predictor) is proposed that considers neighbors' Common Neighbor when predicting a new link, when at least two neighbors of a node share the same node in common as a neighbor. Based on this intuition, they run their prediction algorithm for a given number of cycles, λ . Later, the algorithm is refined as Refined or Popularity aware NCNP. This version considers the weights of the possible links. As a result, a link is predicted if its probability is higher than a given threshold. Our approach is different from Zhang et al.'s methods since we return top-k neighbors while decreasing the number of communications with other nodes.

Another related work is SoCS (Social Coordinate Systems), an algorithm proposed by Kermarrec et al. for link prediction in decentralized social networks [11]. While SoCS is a distributed algorithm, it considers neither P2P environment nor related performance issues. SoCS uses force based graph embedding that depends on iterative forces that are attractions and repulsions.

In Figure 1, nodes c , b and e apply attraction to node a and all the nodes except a itself apply repulsions to node a . Each neighbor of the selected node applies an attraction force to that node and all nodes apply a repulsion force to it.

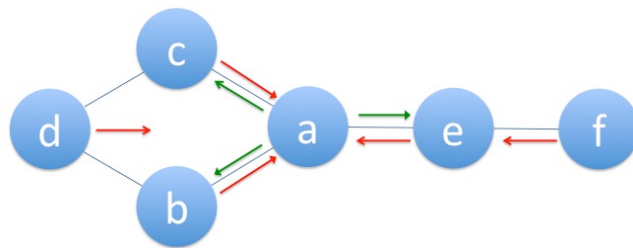


Figure 1 - Example of a friendship social network

The SoCS algorithm represents each node by a vector. After the graph is initialized, repulsions and attractions are calculated for each node in the network. To do so, each node retrieves neighbors of its neighbors by calling a function of its neighbors. The algorithm then calculates velocity and direction of each node. As a result, it finds new position of each node. This process is applied while the graph is unstable which is decided by a convergence function. At last, the algorithm calculates the distance between the node and its neighbors' neighbors and returns the ones that are less than or equal to the acceptable range. SoCS does not consider a P2P environment, while our approach is built on a P2P setting.

We utilize Fagin's top-k query processing for middleware that filters conditions to get top-k objects [12]. Since optimality is often achieved on the worst case, Fagin et al. proposed another approach, TA (Threshold Algorithm), which terminates earlier [13]. Top-k query processing is also discussed in [14] for unstructured P2P networks that focus on challenges of dynamic structure.

Buchegger et al. discuss the requirements of a P2P infrastructure for social networks including distributed storage of data, networking, security, and privacy [1]. In a P2P social network environment, providing a reliable and secure platform is an important challenge. This can be achieved by encryption of data and digestion of access authentication [15]. A potential solution is to use available metadata information, which has some potential side effects [16].

3. P2P LINK RECOMMENDATION

In this section, we discuss how the link prediction problem can be adapted to a P2P setting where each node has only local information about the network. Every node is able to initiate a local link recommendation process of its own. We define 'link recommendation', as suggesting a new link to a peer by considering not only social neighborhood but also network weight information to improve the network performance. While one could assume a uniform weighting of nodes, a more appropriate solution should assign weights based on a P2P social network criteria. A global weight can be assigned to each node in a distributed fashion. Alternatively, each node can assign weights to its neighbors locally utilizing methods from the P2P literature.

The efficiency of a link recommendation algorithm can be measured simply by the cost of communication between nodes. The accuracy of the result can be measured based on both the traditional social network measures (difference between predicted vs. actual) and how the result improves the underlying P2P infrastructure.

3.1 Common Neighbor Recommendation

To solve the P2P link recommendation problem, we start by adapting a Common Neighbor based approach which produces a recommendation to a node x based on highly weighted common neighbors of x 's neighbors. To minimize the communication cost while merging the weights of recommendations from the neighbors, we utilize Fagin's top-k algorithm (FA) and an approximate threshold algorithm (TA) that were originally proposed for top-k Common Neighbor processing.

In FA, each node initializes its neighbor map and enters a loop until finding k neighbor recommendations retrieved from all its neighbors. For each iteration i , neighbor recommendations are retrieved from each neighbor, which is the i^{th} top neighbor of the requested neighbor. Then the neighbor map is updated and the node checks if k recommendations have been retrieved from all its neighbors. If so, it stops and calculates top-k neighbors by assessing weights; otherwise it starts another iteration. The threshold algorithm is the modified version of FA differing for its stopping condition [13]. Even though FA is optimal, it can result in the worst-case scenario with high probability. To avoid this problem TA was proposed where it stops at least as early as FA. Thus, TA calculates a threshold value at each iteration for the last retrieved recommendations. If there are k neighbors that have higher rate than the threshold value, the algorithm terminates.

```
01. Initialize neighborMap
02. WHILE(TRUE)
03.   FOR each neighbor in node
04.     Request recommended node
05.     ADD recommended node to neighborMap
06.   ENDOLOOP
07.   Calculate threshold by last recommendations
08.   Remember top-k highest and discard the others
09.   IF(all neighbors are higher than
10.     threshold in neighborMap)
11.     BREAK
12. ENDOLOOP
13. RETURN neighborMap.neighbors
```

Figure 2 – Top-K TA Common Neighbor algorithm

Figure 2 shows the P2P TA algorithm. Similar to FA, TA also initializes the neighbor map. For each neighbor, it requests the next recommendation, which is the i^{th} top neighbor. At the end of each recommendation, an approximate threshold is calculated by the last retrieved recommended nodes. k neighbors that have the highest weights are remembered at each step, and the others are discarded. If the remembered neighbors' weights are all higher than the threshold that is calculated, it ends up with the neighbors in the neighbor map. The algorithm uses weights of nodes in pruning and decision of final recommendation. We discuss next how these weights can be assigned.

3.2 Node Scoring

While a simple Common Neighbors approach would consider no weights of nodes, one can actually assign weights representing each node's strength in topology, social network, or P2P structure. There is a significant body of work that assigns scores to nodes in a social network, or in a P2P structure. One can borrow those methods to deduct a combined weighting for P2P social networks. We now present such strategies based on the properties of the social networks, P2P systems and their combination.

Node Importance. *Centrality* is defined as the relative importance of an individual node in the network [17]. There are several types of centrality such as betweenness, closeness, and eigenvector [18]. Betweenness centrality is related with the number of shortest paths that pass through a node. Closeness is inverse of the average distance from all the other nodes to a particular node. Eigenvector centrality promotes high-scoring nodes more than the others.

Influence is another measure commonly used to determine how to select an initial set of k users such that they influence the largest number of users at the end [19]. *Reputation* and *authority* are other measures that are similar to influence score.

Clustering Coefficient is a measure related with the number of neighbors a node has and the edges between its neighbors [20].

Reputation Scoring. *Trust* is one of the most challenging tasks in P2P systems since a node can appear and disappear instantly [21]. Trust/Reputation models are based on the values that are assigned between nodes such that node i assigns a trust value to node j , and vice versa. There are a lot of trust models, including CuboidTrust [22], EigenTrust [23], BNBTM [24], GroupRep [25], etc.

P2P Systems Measures. The lifetime of a peer determines its *availability*. The simplest way of implementing availability is waiting up for a given time and marking the node as online or offline [26]. *Latency* is the time spent transmitting a file from one node to another. *Bandwidth* is the speed of the connections to the Internet where bottleneck can be both the receiving peer and the sending peer. Thus, bottleneck bandwidth depends on the slowest peer in the process.

Composite Measures. For scoring the nodes in a P2P Social Network, one can simply combine the scores from social network and P2P systems. For example, a "trusted centrality" score can be developed by combining P2P trust and graph centrality measures. We involve both centrality and trust to recommend new links for the peers to trust and provide higher access to other nodes:

$$tc = t * \alpha + (1 - \alpha) * c$$

where tc stands for trusted centrality, t is the normalized trust score, c is the normalized centrality score and α is the constant for adjusting the weights of centrality and trust. We used this measure for our results in the experimental section.

Another example for composite measures is "available authority", combining availability score from the P2P network and the authority score from the network topology. Recommended peers are reachable most of the time and they are authority for the communities that the peer is participating in.

For link recommendation, we utilize a weighting strategy that unifies the social and network connections. While this simplifies the process, it may result in forcing friendship connections to improve the P2P topology. One can separate the problems of social connection and network connection and run the common neighbor algorithm with separate weights for each.

4. EXPERIMENTAL RESULTS

To evaluate the proposed algorithms, we simulated a P2P network where each node knows only its neighbors and their global or local weights. We also have been building a P2P social network application, which we refer to as SOWHOO [27].

We used both real data (Gnutella and Wikipedia), and synthetic data for the experiments. For synthetic experiments, we generated

different classes of network structures using uniform distribution, the small world model of Watts and Strogatz [28], and finally the clustering model of Holme and Kim [29]. Please note that while a graph can have a uniform edge distribution, the weighting scheme can be non-uniform. We evaluate the efficiency of the methods using the communication cost (the number of messages) as our performance measure. The accuracy of the results depends on the weighting used. The recommended links are not necessarily the common neighbor. The node that collects a higher collective weight would be preferred to a common neighbor with low weight.

We first compare the Top-K Common Neighbor, Top-K FA Common Neighbor and Top-K TA Common Neighbor with respect to the communication cost. We present the cost of each algorithm as the number of edges in the underlying graph changes. During the experiments, both global and local weights are assigned to each node. The global weight for each node is single and does not change according to any node. It can be assigned in a distributed fashion as well. Examples of such global weights are clustering coefficient or influence score. A local weight of a node is its value with respect to another node. The value is not known by any other node except for the one that made the assignment. Examples for local weights are the rating assigned by a node to its neighbor or the number of interactions between two nodes.

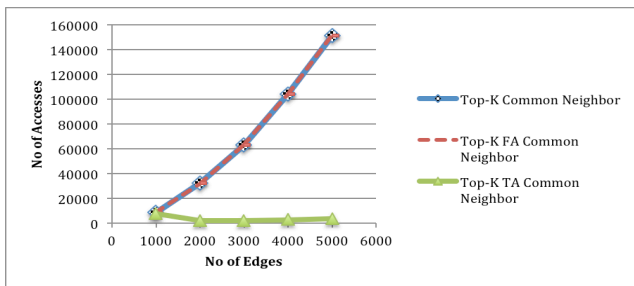


Figure 3 - No of accesses vs. no of edges for uniform graph

We first test the algorithms on the uniform graph where weights of the nodes are assigned locally with power-law distribution. Figure 3 illustrates performance results in terms of the number of messages as the number of edges in the graph increases. Top-K FA Common Neighbor and Regular Top-K Common Neighbor algorithms perform the same. Top-K TA Common Neighbor algorithm outperforms, by far, the other two algorithms. Note that we adapted an approximate version of the original TA algorithm to our context. As the number of edges in the network increases, the performance benefit increases since it terminates in very early iterations compared to the other two.

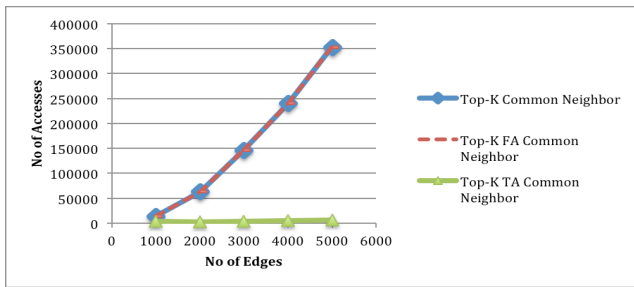


Figure 4 - No of accesses vs. no of edges for small world graph

Small World Model. Figure 4 presents the results for the small world model graph. The edges of the graph are generated according to a ring structure. Weights of the nodes are assigned

with power-law distribution locally. Similar to the previous experiment, as the number of edges in the network increases, Top-K TA Common Neighbor performs better than the other two algorithms, due to its termination in early iterations.

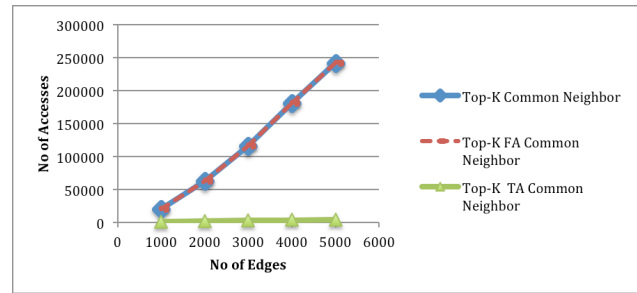


Figure 5 - No of accesses vs. no of edges for clustered graph

Clustered Model. We tested the on clustered model graph where the edges are generated in clusters. Weights assigned to the nodes follow power-law distribution. The performance results of the three algorithms follow a similar pattern as before (Figure 5).

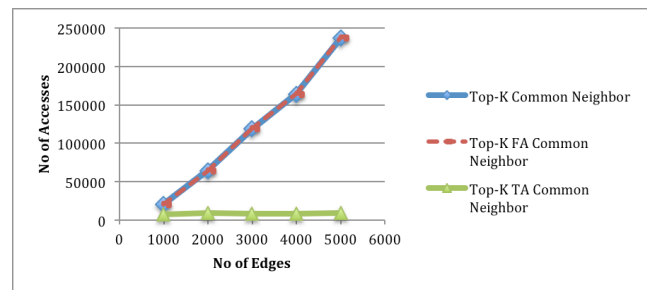


Figure 6 - No of accesses vs. no of edges for global weights

Global Weighting. The performance of the algorithms follow a similar pattern also with global weights, where each node is assigned a global value with power-law distribution. Top-K TA Common Neighbor with global weights performs better than its local weight case. Figure 6 illustrates these results for the clustered graph model.

Real Data. We compared the performance of the algorithms using the Gnutella P2P network [30] and Wikipedia vote network [31]. Gnutella data is one of the snapshots of Gnutella network in 2002. In this snapshot, there are 6301 nodes and 20777 edges with an average clustering coefficient of 0.0150. Wikipedia vote network data set includes small part of the Wikipedia contributors voting each other to become administrator. Wikipedia voting data is extracted from this election data and vote history having 7115 nodes and 103689 edges with average clustering coefficient of 0.2089.

In Gnutella data set, the Top-K TA algorithm does 605,433 accesses while Top-K FA does 676,430 accesses under uniform weight distribution setting. Under power law weight distribution, Top-K FA does 676,432 accesses and Top-K TA does 326,227. Top-K TA is better than Top-K FA in both settings while the difference between two is larger in power law setting.

In Wikipedia data set, the Top-K TA algorithm outperforms Top-K FA with 377,929 accesses under power law weight distribution setting where Top-K FA does 28,670,223 accesses. Under uniform weight distribution setting, Top-K TA results in 14,687,511 accesses while Top-K FA does 28,671,824.

Accuracy vs. Efficiency. To understand the accuracy and efficiency tradeoff, we use a performance measure that involves

both accuracy and the number of accesses to a node. Our measure, ω , is specified using the following formula, where a denotes the accuracy and n denotes the number of accesses. Maximum of n is reached when all the nodes are accessed.

$$\omega = (a - (n / \max(n)))$$

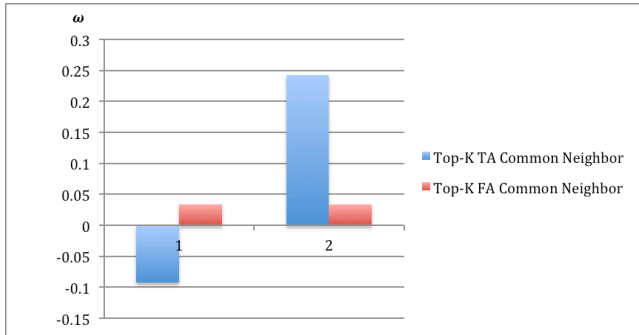


Figure 7 - Uniform (left) and Power-law (right) distribution for Gnutella data set

We have assigned the weights according to both the uniform distribution and power law distribution. As shown in Figure 7, when the uniform distribution is applied (left hand side), FA becomes a good choice; however, when the power law distribution is applied (right hand side graph), TA seems to be a better choice for Gnutella data set.

In Figure 8, we can see that Top-K FA behaves similar to its performance obtained with the Gnutella data set. On the other hand, Top-K TA works better when uniform distribution is applied contrary to the Gnutella results. Consequently, Top-K TA is the better choice regardless of weight distribution with the Wikipedia data set.

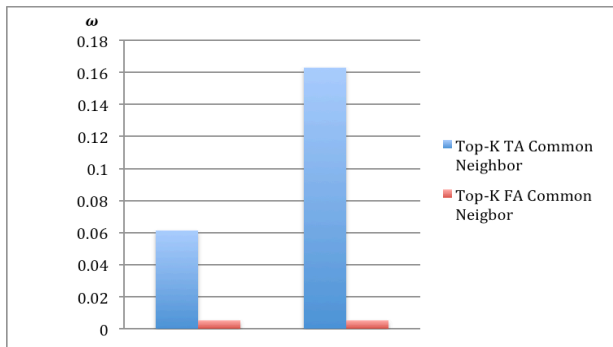


Figure 8 - Uniform (left) and Power-law (right) distribution for Wiki Vote data set

Trusted Centrality. We also implemented the trusted centrality weighting that we propose. To compute this measure, we use the betweenness centrality proposed by Newman for each peer [32]. Since we do not have a fully working P2P infrastructure yet, we assign the trust scores according to the power law distribution as suggested by Zhou and Hwang [33]. We have obtained similar results with the power law distribution in small-world network and clustered network for both Top-K TA and Top-K FA. This result can be due to the small number of edges and the small network.

5. DISCUSSION

We presented our ongoing work on the interesting problem of link recommendation in P2P social networks. We studied two algorithms, Top-K FA and TA Common Neighbors to find recommended links for a node. We also presented weighting methods for link recommendation and proposed trusted centrality combining P2P trust and social network centrality.

The TA algorithm is significantly more efficient than the FA algorithm. This is mainly due to the observation that FA performs mostly in its worst-case in this new setting. Hence FA performs as a regular common neighbor approach since finding k common nodes in all neighbors is not likely in real social networks. One can improve FA by removing the requirement of k nodes retrieved from all neighbors. One can approximate this by not retrieving nodes from all neighbors but from a certain number of neighbors. TA and FA show a tradeoff between accuracy and efficiency as illustrated in Figure 7.

For a complete P2P social network, a large body of research and implementation is needed, such as authentication, privacy, publish/subscribe services and notifications. We are currently building a simple P2P social network application [27]. This application will enable experimenting our algorithms on a real P2P infrastructure with global or local weights assigned according to the underlying activities. There is a wide range of design alternatives one can follow to implement such a system. An example is a hybrid P2P infrastructure, where there are simple peers and super peers. Each peer in the system has a super-peer to provide other peer addresses such as neighbor peers. In order to provide peer addresses, super peers are designed to have a DNS like protocol in which each super-peer delegates address inquiry message to parent super-peer if peer-address is not found in local repository. As a result, there must be at least one super-peer, which has permanent address for system start-up. This super-peer would keep track of super-addresses and if there is no other super-peer, it would also keep track of simple-peers.

We consider a partitioning algorithm for distribution of peers to super-peers. Each super-peer would have a self-balancing mechanism to hold similar number of addresses to have more uniform network. This would prevent the system from depending so much in particular super-peers, which may result in overload for that particular super-peer.

6. REFERENCES

- [1] Sonja Buchegger and Anwitaman Datta. A case for P2P infrastructure for social networks - opportunities and challenges. *6th International Conference on Wireless On-demand Network Systems and Services*, Snowbird, Utah, USA, 2009.
- [2] Philip E. Agre. P2P and the Promise of Internet Equality. *Communications of the ACM*, Vol.46, No.2, pp.39-42, 2003.
- [3] Charu C. Aggarwal, *Social Network Data Analytics*, Springer Publishing Company, Incorporated, 2011.
- [4] David Liben-Nowell, Jon Kleinberg, The link prediction problem for social networks. *12th international conference on Information and knowledge management*, 2003, New Orleans, LA, USA.
- [5] Lars Backstrom, Jure Leskovec, Supervised random walks: predicting and recommending links in social networks, *4th ACM international conference on Web search and data mining*, 2011, Hong Kong, China.

- [6] L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211--230, 2003.
- [7] Sarkar, P., Chakrabarti, D., Moore, A.W.: Theoretical justification of popular link prediction heuristics. *23rd Annual Conference on Learning Theory*, 2010, Haifa, Israel.
- [8] Ö. Ulusoy, Research Issues in Peer-to-Peer Data Management (Keynote Paper), *International Symposium on Computer and Information Sciences (ISCIS'07)*, November 2007, Ankara, Turkey.
- [9] Ihab F. Ilyas, George Beskales, Mohamed A. Soliman, A survey of top-k query processing techniques in relational database systems, *ACM Computing Surveys (CSUR)*, v.40 n.4, p.1-58, 2008.
- [10] Yaodong Zhang, Guobin Shen, Young Yu, LiPS: Efficient P2P Search Scheme with Novel Link Prediction Techniques in *IEEE International Conference on Communications (ICC '07)*, 2007.
- [11] Anne-Marie Kermarrec, Vincent Leroy, Gilles Trédan, Distributed Social Graph Embedding, *20th ACM international conference on Information and knowledge management* Pages 1209-1214.
- [12] Ronald Fagin, Combining fuzzy information: an overview, *ACM SIGMOD Record*, v.31 n.2, June 2002.
- [13] Ronald Fagin, Amnon Lotem, Moni Naor, Optimal aggregation algorithms for middleware, *20th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, p.102-113, 2001, Santa Barbara, California, USA.
- [14] Zhitao Guan, Guangwei Yan, Heqing Huang, A Novel Top-k Query Scheme in Unstructured P2P Networks, *9th IEEE International Conference on Computer and Information Technology - Volume 02* Pages 16-21.
- [15] Oleksandr Bodriagov, Sonja Buchegger. Encryption for Peer-to-Peer Social Networks, *3rd IEEE Third International Conference on Social Computing* 9-11 Oct. 2011, Stockholm, Sweden.
- [16] Benjamin Greschbach, Gunnar Kreitz and Sonja Buchegger. The Devil is in the Metadata – New Privacy Challenges in Decentralized Online Social Networks. *4th International Workshop on SECURITY and SOCIAL Networking*, 2012, Lugano, Switzerland.
- [17] Freeman, L.C.: Centrality in social networks. *Social Networks* 1, 215–239 (1979).
- [18] Arun S. Maiya and Tanya Y. Berger-Wolf, Online Sampling of High Centrality Individuals in Social Networks, *PAKDD'10: 14th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining - Volume Part I*.
- [19] Amit Goyal, Francesco Bonchi, Laks V.S. Lakshmanan, Learning influence probabilities in social networks, *3rd ACM international conference on Web search and data mining*, 2010, New York, New York, USA.
- [20] D. J. Watts and S. H. Strogatz. Collective dynamics of “small-world” networks. *Nature*, 393:440–442, 1998.
- [21] Felix Gomez Marmol and Gregorio Martinez Perez, "State of the Art in Trust and Reputation Models in P2P Networks," in *Handbook of Peer-to-Peer Networking*, Xuemin Shen et al., Eds.: Springer, 2010, pp. 761-784.
- [22] R. Chen, X. Chao, L. Tang, J. Hu, Z. Chen, CuboidTrust: A Global Reputation-Based Trust Model in Peer-to-Peer Networks, in: *Autonomic and Trusted Computing*, No. 4610 in LNCS, *4th International Conference, ATC 2007*, Springer, Hong Kong, China, 2007, pp. 203–215.
- [23] Sepandar D. Kamvar, Mario T. Schlosser, Hector Garcia-Molina, The Eigentrust algorithm for reputation management in P2P networks, *Proceedings of the 12th international conference on World Wide Web*, May 20-24, 2003, Budapest, Hungary.
- [24] Yong Wang, Vinny Cahill, Elizabeth Gray, Colin Harris, Lejian Liao, Bayesian network based trust management, *Proceedings of the Third international conference on Autonomic and Trusted Computing*, September 03-06, 2006, Wuhan, China.
- [25] H. Tian, S. Zou, W. Wang, S. Cheng, A group based reputation system for P2P networks, *Proceedings of the Third international conference on Autonomic and Trusted Computing*, September 03-06, 2006, Wuhan, China.
- [26] Saroiu, S., Gummadi, P., and Gribble, S. 2002. A Measurement Study of Peer-to-Peer File Sharing Systems. *MMCN*, San Jose, CA, USA January 2002.
- [27] <https://github.com/yusufaytas/sowhoo>.
- [28] Watts, D.J.; Strogatz, S.H. (1998). Collective dynamics of 'small-world' networks *Nature* 393 (6684): 409–10.
- [29] P. Holme and B. J. Kim, Growing scale-free networks with tunable clustering, *Phys. Rev. E*, vol. 65, pp. 026107–1–4, 2002.
- [30] <http://snap.stanford.edu/data/p2p-Gnutella08.html>.
- [31] <http://snap.stanford.edu/data/wiki-Vote.html>.
- [32] M.E.J. Newman, A Measure of Betweenness Centrality Based on Random Walks, *Social Networks*, vol. 27, no. 1, pp. 39-54, Jan. 2005.
- [33] Runfang Zhou, Kai Hwang, Trust overlay networks for global reputation aggregation in P2P grid computing, *Proceedings of the 20th international conference on Parallel and distributed processing*, p.29-29, April 25-29, 2006, Rhodes Island, Greece.