Distributed Social-based Overlay Adaptation for Unstructured P2P Networks

Ching-Ju Lin
Institute of Networking and Multimedia
National Taiwan University, Taipei, Taiwan
cjlin@cmlab.csie.ntu.edu.tw

Yi-Ting Chang, Shuo-Chan Tsai, Cheng-Fu Chou
Dept. of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan
{seashell, r92069, ccf}@cmlab.csie.ntu.edu.tw

[1]

Abstract— **The widespread use of Peer-to-Peer (P2P) systems has made multimedia content sharing more efficient. Users in a P2P network can query and download objects based on their preference for specific types of multimedia content. However, most P2P systems only construct the overlay architecture according to physical network constraints and do not take user preferences into account. In this paper, we investigate a social-based overlay that can cluster peers that have similar preferences. To construct a semantic social-based overlay, we model a quantifiable measure of similarity between peers so that those with a higher degree of similarity can be connected by shorter paths. Hence, peers can locate objects of interest from their overlay neighbors, i.e., peers who have common interests. In addition, we propose an overlay adaptation algorithm that allows the overlay to adapt to P2P churn and preference changes in a distributed manner. We use simulations and a real database called *Audioscrobbler*, which tracks users' listening habits, to evaluate the proposed social-based overlay. The results show that social-based overlay adaptation enables users to locate content of interest with a higher success ratio and with less message overhead.**

I. Introduction

The widespread use of P2P systems has made sharing multimedia content, such as music and video files, more efficient. In a social network, people with similar tastes in multimedia content (e.g., people who like jazz music) form a community to share their experience and knowledge. Like people who form social networks, some users of P2P networks have a preference for various types of multimedia content, which may affect the way they query and download content. Although P2P users normally exchange multimedia content with other users who have similar tastes, the architecture of most well-known P2P systems is based on physical network constraints only and does not take user preferences into account. To remedy the situation, in this paper, we propose a social-based P2P overlay that can leverage social phenomena to improve the efficiency of content sharing in P2P systems.

In sociology, a social network [1] is comprised of a set of actors (nodes) that may have relationships (ties) with one another. Sociologists normally use graphs to represent information about relationship patterns between social actors. Such graphs are also called "socio-grams" in sociology. The design of social-based P2P overlay networks is motivated by the concept of socio-grams. More specifically, the objective of the proposed social-based P2P network is to build a socio-gram as an overlay topology for the P2P network.

In a socio-gram, an edge between a pair of nodes indicates that a tie exists between two adjacent nodes; for example, if we are interested in each node nominates which nodes as friends, an edge can be used to represent a friendship tie. In a P2P network, a user hopes to obtain objects of interest from peers who have similar tastes and can provide the requested objects. The key to efficient and scalable searches in unstructured P2P systems is to cover nodes holding the requested objects as quickly as possible and with as little overhead as possible. Actually, the only way to find objects of interest is to continue visiting peers until one that holds the requested object is found.

In this paper, instead of building a friendship socio-gram, we build a similarity-based socio-gram as a P2P overlay topology, where a similarity tie between two peers exists if they have common interests in specific types of multimedia content. Hence, in the proposed social-based overlay, peers sharing similar interests can be connected by shorter paths so that they can exchange multimedia content efficiently. Specifically, whenever a peer requests an object of interest, it can locate the object among its neighboring peers, i.e., the peers who have similar tastes and are more likely to hold the requested object.

The following factors determine the efficiency of a social-based overlay. (1) Similar peer selection: in decentralized P2P systems, it is challenging to define user preferences and identify peers who have similar tastes. (2) Distributed overlay adaptation: a system can collect information about all users to estimate the similarity between peers. However, the centralized method is not scalable, and it can not cope with changes of users' preferences and network dynamics, i.e., churn (defined as the dynamics of peers joining or leaving [2][3]). Therefore, a distributed adaptation algorithm is required so that each peer can discover its similar peers and maintain overlay links distributedly and dynamically.

The goal of this paper is to model a distance measure that quantifies the similarity between peers; hence, peers can form an effective social-based overlay based on the proper similarity measure. On the other hand, we propose an overlay adaptation algorithm that uses a random walk technique to sample the population and discover similar peers from the randomly

[1] This work was partially supported by the National Science Council and the Ministry of Education of ROC under the contract No. NSC95-2221-E-002-103-MY2 and NSC95-2622-E-002-018.

selected samples, instead of collecting detailed information about all P2P users. Because the random walk technique reduces the overlay update overhead significantly, each peer can exploit this method to handle dynamic churn and adapt to changes of users' tastes efficiently and distributedly. Finally, we use a database called *Audioscrobbler*, which tracks users' listening habits, to evaluate the performance of the proposed social-based P2P network.

The remainder of the paper is organized as follows. Section II presents related works on random-walk-based P2P systems and social-based P2P systems. Section III describes the proposed social-based overlay construction algorithm in detail. Section IV evaluates the performance of the social-based overlay via simulations. Then, in Section V, we present our conclusions.

II. Related Works

Decentralized P2P systems are typically classified into two categories: structured P2P systems and unstructured P2P systems. In structured P2P systems, i.e., Distributed Hash Table (DHT) systems, both data placement and the overlay topology are tightly controlled. However, although DHT systems balance the workload and improve query efficiency, most DHT systems must repair the architecture for each node failure; hence, they can not handle churn efficiently. In unstructured P2P systems, such as Gnutella [4], each node incurs a reasonable overhead to build overlay links and repair link failures dynamically according to some loose rules. In addition, querying multimedia content by keywords has become increasingly popular in P2P systems. Unlike DHT systems, which incur extra overlay maintenance costs for providing a keyword search service [5][6][7][8], users of an unstructured system can forward query messages as a sequence of keywords by flooding to find objects that partially match the query keywords. Because of the robustness and flexibility of unstructured systems, we adapt the Gnutella system to social-based unstructured P2P networks, in which the overlay topology is based on the social relationships between peers.

In decentralized unstructured P2P systems, the use of a flooding scheme for overlay construction or content queries induces a scalability problem. Hence, some approaches [9][10][11] use a random walk technique, rather than flooding, to reduce the message overhead. However, the lack of flow control and topology control is one of the weaknesses of random-walk-based Gnutella-like systems. A number of works [2][12][13] balance the load among peers by controlling the number of outlinks and inlinks explicitly based on the bandwidth capability of each peer. Consequently, in graphs that control the node degree, a node with higher capacity will have more overlay links so that there is a higher probability that it will donate its bandwidth resources.

However, random-walk-based Gnutella-like systems can not guarantee that queries will be handled efficiently. Since overlay construction based on random walk does not take user preferences into account, a node may not be able to locate objects of interest from its overlay neighbors. Thus, it may need to visit more peers to locate the requested objects, and thereby generate more message overhead. To address this problem, some works [14][15] have proposed social-based P2P systems, which build an overlay topology that mimics social phenomena. The objective is to connect peers based on their social relationships so that a peer can obtain content efficiently from its neighboring overlay nodes.

In [14], each peer establishes overlay links with peers who have similar preferences. The similarity of peers is measured by comparing their preference lists, which record a number of the most recently downloaded objects. However, this method causes a *new user problem*; that is, a new user who has only made a few downloads can not get an accurate similarity measure. In [15], a central server collects the description vectors of all users, and establishes overlay links based on the distance between each pair of users. One limitation of the centralized methods is that they can not handle churn in P2P systems efficiently, since they generate a heavy traffic load when exchanging information in a large-scale network. In addition, [15] does not explicitly define the description vector, which has a significant effect on the accuracy of the similarity measure.

In this work, we propose a novel social-based overlay for unstructured P2P networks. Our contribution is twofold: (1) we define a quantifiable measure of the similarity between each pair of peers; and (2) we propose an overlay adaptation algorithm that enables each node to establish ties with similar nodes in a distributed and dynamic manner based on a random walk technique. The proposed method uses social relationships to improve the performance of content search, and exploits the advantages of the random walk method to reduce the overlay construction overhead.

III. Distributed Social-based Overlay Construction

In this section, we present an overview of the social-based overlay topology, and then describe in detail how to construct a distributed social-based overlay network based on a random walk technique.

A. Overview of a Social-based Overlay

A social-based overlay for P2P networks clusters users who have similar preferences for multimedia content. Thus, we build a similarity-based socio-gram, denoted as G_s, in which a tie between two peers exists if they have a common interest in specific types of multimedia content. To determine whether two nodes should be connected by a *similarity* tie, the system needs to compile user profiles containing information about users' preferences, and then measure the degree of similarity between the profiles. From a real-world perspective, the objects held by a peer typically reflect the characteristics of that peer, since a peer has limited storage capability and may not keep objects that are not of interest. Therefore, users can be distinguished by the objects they hold. Many works focus on techniques that extract low level or semantic metadata from multimedia objects. An object can be described by multiple attributes, which can be associated with the extracted metadata. For example, a music file can be associated with several keywords (i.e., metadata), such as *genre*=“`Jazz`”, *artist*=“`Pat Metheny`”, *title*=“`Bright Size Life`”. Thus, objects can be categorized based on the tagged keywords. Preferences for different categories of objects can be used to distinguish the characteristics of each peer. Specifically, let $Profile(c_i)$ be the profile of user c_i. $Profile(c_i)$ is defined as a vector of weights $\vec{w_i} = (w_{i,k_1}, w_{i,k_2}, \cdots, w_{i,k_n}, \cdots)$, where the weight w_{i,k_n} denotes user c_i's preference for the objects described by the keyword k_n, as shown by

$$w_{i,k_n} = \frac{|\mathcal{O}_{i,k_n}|}{|\mathcal{O}_i|}, \tag{1}$$

where $\mathcal{O}_i$ is the set of objects held by user c_i and $\mathcal{O}_{i,k_n}$ is a subset of $\mathcal{O}_i$ containing the objects tagged by the keyword k_n. We then use the cosine similarity measure [16][17] to quantify similarity $sim(c_i, c_j)$ between two peers, c_i and c_j , as follows:

$$sim(c_i, c_j) = cos(\vec{w_i}, \vec{w_j})$$

$$= \frac{\vec{w_i} \cdot \vec{w_j}}{\|\vec{w_i}\|_2 \times \|\vec{w_j}\|_2} = \frac{\Sigma_{k=1}^{K} w_{i,k} w_{j,k}}{\sqrt{\Sigma_{k=1}^{K} w_{i,k}^2}\sqrt{\Sigma_{k=1}^{K} w_{j,k}^2}}, \tag{2}$$

where K is the total number of keywords. If c_i and c_j have similar tastes in certain styles of multimedia content, then $sim(c_i, c_j)$ returns a smaller value.

In the proposed social-based P2P network, each peer finds d similar peers (so called buddies) distributedly, and establishes overlay links with them. However, constructing a similarity graph G_s does not guarantee the connectivity of a P2P overlay network. Hence, in the proposed social-based overlay topology, we merge G_s with a weak graph, denoted as G_w, which connects two peers named the consecutive identifiers. In other words, all peers are connected as a ring topology in G_w to avoid partitioning the overlay topology. Thus, in the social-based overlay, each node builds $(d+1)$ overlay outlinks, d for G_s and one for G_w.

The proposed similarity measure can resolve the new user problem because a new user can also provide his/her multimedia content in the buffer space. Hence, the profile for a new user can be created based on the objects stored in the buffer. The other unexpected advantage of the proposed user profiling method is that it discourages freeriders in P2P systems. If a peer does not offer content in its public storage space for other users, its preference (i.e., user profile) can not be compiled precisely, so it can not find buddies based on its user profile. Therefore, the proposed similarity measure can inherently provide incentive for users to share their resources.

B. Distributed Overlay Adaptation

The overlay topology is the component that connects all peers in an unstructured P2P network. The overlay topology must be updated efficiently so that it can react to dynamic churn. Hence, we propose an overlay adaptation algorithm that allows each peer to determine its buddies in a distributed manner. When a new user joins a P2P network, it uses bootstrapping mechanisms, similar to those used in Gnutella, to locate other peers in the overlay topology. It then builds temporary overlay links with those peers to connect to the P2P network, exchanges information with neighbors, and compiles its buddy list distributedly.

Given a set of peers, a new peer can use certain strategies to collect information about the peers to determine their relationships and compile a buddy list. The strategies can be categorized into two types [1]; *full network* methods and *snowball* methods. *Full network* methods collect the user profiles of all peers in a central server, and rank $sim(c_i, c_j)$ for any pair of peers, c_i and c_j, in the system. The method allows the central server to analyze the social structure explicitly and cluster peers who have similar preferences; however, it can be very expensive to collect full information as the network scales up. In contrast, the *snowball* method collects information via epidemic protocols, i.e., a peer can know *friends-of-friends* through its friends. Because the snowball method only samples the target population, the information exchange overhead is much lower, which resolves the scalability problem. Hence, we propose a distributed overlay adaptation algorithm that enables each peer to compile its buddy list distributedly based on the concept of snowball sampling.

In the following, we present the proposed distributed overlay adaptation algorithm, which involves two phases: *distributed buddy selection* and *buddy list update*.

1) Distributed Buddy Selection: To reduce the message exchange overhead, each node can locate buddies with similar tastes from a subset of overlay nodes (called candidates hereafter). Each peer, c_i, can find M candidates from the overlay distributedly and randomly, and calculate the cosine similarity measure, $sim(c_i, c_j)$, for any candidate c_j. Therefore, each peer can maintain a list of the d most similar buddies, i.e., the candidates that yield smaller values of $sim(c_i, c_j)$, and establish the overlay links with the peers in the buddy list. In this method, the effectiveness of the buddy list depends on the efficiency of the candidate selection mechanism. The most efficient way (i.e., the method that generates the lowest message overhead) is to select the d most similar peers from the M neighboring overlay nodes. However, when locating neighbors, the bootstrapping procedure does not consider the characteristic of peers, so a new peer may not be able to find any peers who have similar tastes or interests to itself under this procedure. To resolve this problem, we need an unbiased sampling mechanism that can randomly select a set of candidates from the overlay topology.

Random walk is a typical unbiased sampling technique that forwards a request to a randomly selected neighbor with a probability p at each step, or stops in a visited node with a probability $(1 - p)$. The technique reduces the message overhead significantly, since each request takes its own random walk and generates only as many messages as the length of the path it traverses. In contrast to the flooding method, in the random walk method, the number of messages does not

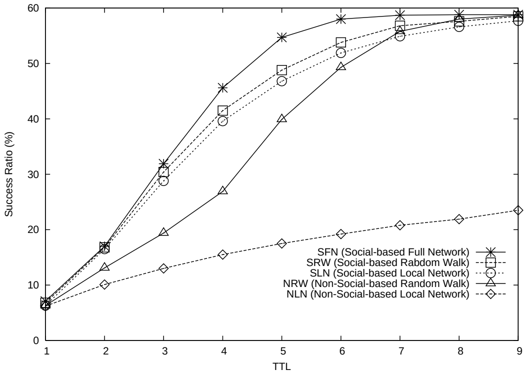

Fig. 1. TTL vs. Success Ratio

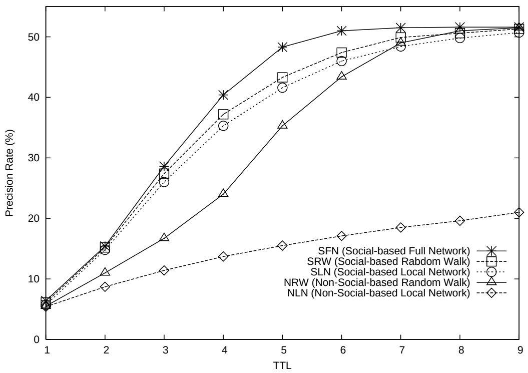

Fig. 2. TTL vs. Precision Rate

increase exponentially with the number of outlinks of each traversed node. In addition, assuming the location of a peer in an overlay is independent of the tastes of that peer, we can exploit an advantage of the random walk mechanism whereby M candidates can be selected randomly and unbiasedly from the overlay network. Hence, to reduce the information collection overhead and avoid biased candidate selection, each node can use the random walk method to select candidates distributedly. However, to strike a balance between the message overhead and unbiased candidate selection, we let each peer designate $\lfloor \frac{M}{2} \rfloor$ nearest neighbors as candidates and also start $\lceil \frac{M}{2} \rceil$ walks to discover $\lceil \frac{M}{2} \rceil$ candidates randomly. Then, a peer can select d buddies distributedly from the M candidates.

2) Buddy List Update: A peer may lose its outlinks if its buddies fail or leave the network. Besides, since peers' tastes may change over time, a peer's tastes may no longer be similar to those of the peers on the buddy list. Hence, a peer must update its buddy list in the following cases: (1) its outlinks are lost, or (2) its user profile is changed. In the second case, user profile modification only occurs in the following situations: (1) when a peer retrieves new objects from other peers, or (2) objects cached in the buffer are deleted by the user or dropped because of buffer overflow. Based on the concept of social phenomena, each peer can exploit the snowball method to locate more buddies through friends-of-friends, since users with similar preferences are usually clustered in a community. When a peer decides to update its buddy list, it locates $2d$ candidates: d friends-of-friends and d peers chosen by random walk. Then, it ranks all candidates and the original buddies in order of their similarity measure $sim(c_i, c_j)$, and updates its buddy list with the d most similar peers.

IV. Performance Evaluation and Discussion

In this section, we use simulations to evaluate the performance of the proposed distributed social-based overlay construction algorithm. To validate the proposed algorithm, we use log-based user profiles collected from *Audioscrobbler*[2], a database that tracks listening habits by collecting the play-lists of users' media players (for instance, Winamp, iTunes, and XMMS). The profiles are used in our simulations to mimic social relationships in the real world. We collect profiles for $1,355$ fans who have listened to five popular styles of music (i.e., rock, metal, pop, punk, and jazz) the most. The number of the fans selected for a specific music style is proportional to the popularity of that style. For each fan, the data set records the 50 songs that he/she listens to the most. Thus, there are $31,005$ objects in our simulations. To simulate a P2P overlay network, we use *brite* [18] to generate the physical network, in which $1,355$ nodes are distributed in a topology of Autonomous Systems (ASes). Then, the 677 nodes (fans) are randomly selected from the physical network to join the P2P overlay network. Each overlay node establishes four outlinks: three for the similarity graph and one for the weak graph.

We classify the 50 music files held by each node into five groups according to genre, and the user profile is defined as $\vec{w} = (w_{\texttt{rock}}, w_{\texttt{metal}}, w_{\texttt{pop}}, w_{\texttt{punk}}, w_{\texttt{jazz}})$. Each overlay node has a buffer that can cache 45 music files. If the buffer is overloaded, cache replacement is based on a popularity-driven algorithm, i.e., the song listened to the least is dropped first. For cross-validation, we randomly divide each node's 50 favorite songs into a training set (40 songs) and a test set (10 songs). Let the training set of songs be cached in each node's buffer. Each node then requests songs in the test set to evaluate the performance of the content query service in the proposed social-based overlay. To evaluate the performance of the keyword search service in the social-based overlay, we let each peer query an object by the tags associated with that object.

We compare three variations of social-based overlay construction methods and two non-social-based methods as follows. (1) Social-based full network (SFN): each peer collects the profiles of all other nodes, and selects d buddies. (2) Social-based random walk (SRW): each peer collects $2d$ candidates (d selected by random walk and d selected from local neighbors) to compile a list of d buddies. Each walk visits a randomly selected neighbor with a probability of 0.5, or stops in a visited node with a probability of 0.5. (3) Social-based local network (SLN): each peer selects d buddies from the $2d$ local neighbors. (4) Non-social-based random walk (NRW): each node starts d random walks, and establishes overlay links with d destination nodes. (5) Non-social-based local network

[2] http://www.audioscrobbler.net/

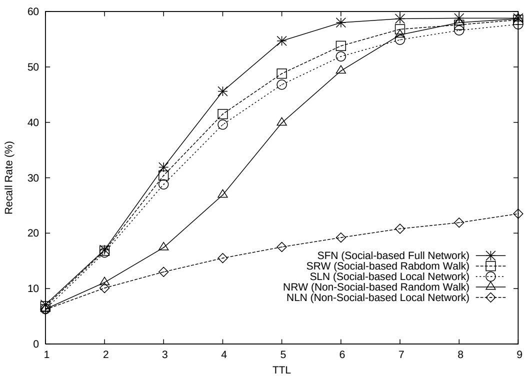

Fig. 3. TTL vs. Recall Rate

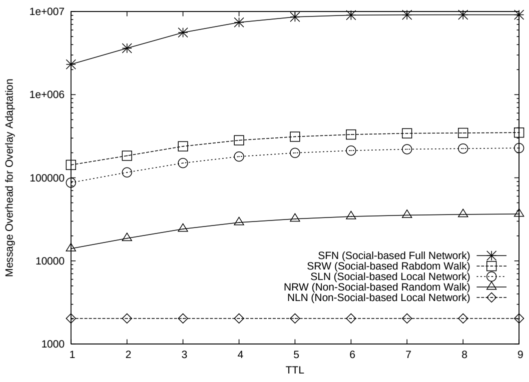

Fig. 4. TTL vs. Overhead of Overlay Adaptation

(NLN): each peer establishes d overlay links with the peers who have consecutive identifiers.

A. Performance Comparison in Static Environments

In this simulation, we compare the performance of the above five schemes in terms of the following performance metrics: (a) the success ratio: the ratio of the number of successful searches to the total number of requests; (b) the precision rate: the number of target objects on the returned list divided by the total number of objects on the returned list; (c) the recall rate: the number of target objects on the returned list over the number of replicas of the target object in the system; and (d) the overlay adaptation overhead: the number of messages used to construct and update the overlay topology. To verify the impact of overlay construction on the performance of the content query service, overlay nodes are not allowed to leave the P2P system during this simulation.

Gnutella-like systems use TTL (Time-To-Live) to control the number of hops that flood a query. This simulation evaluates the performance of all five schemes for various numbers of TTL. Generally, if the TTL is low, peers may not be able to locate the requested objects, even though a copy exists in the overlay network. Conversely, if the TTL is high, peers can discover more overlay nodes and locate the requested objects in the overlay. Figures 1, 2, and 3 show that all schemes achieve better performance in terms of the success ratio, precision rate, and recall rate as the number of TTL increases. However, message overhead caused by flooding increases as TTL increases. To reduce the overhead, a good overlay topology should enable a node to locate the requested object with limited TTL. The figures show that the social-based overlay construction methods, i.e., SFN, SRW, and SLN, outperform the non-social-based methods as TTL is limited to 5. This is because the former methods take user behavior into account and connect peers who have similar tastes. In a social-based overlay, since two buddies can be connected by a shorter path, they can obtain objects of interest with limited TTL from the peers with similar interests. The NLN method performs worst because peers who have consecutive user identifiers are clustered together; hence, the query can not be forwarded to other peers.

The figures also show that the SFN scheme is the best of the three social-based methods. This is because it enables each node to obtain complete information about other peers by collecting all users' profiles to compile a precise buddy list. However, as shown in Figure 4, collecting all user profiles by flooding generates a large message overhead while constructing or updating the overlay topology. The other two social-based methods, SRW and SLN, can perform as well as the full network method, but only incur a small amount of overhead to maintain the overlay links. Because the random walk and local network methods only collect $2d$ candidates' user profiles, they reduce the message overhead of overlay adaptation significantly.

B. Performance Comparison in Dynamic Environments

This simulation evaluates the performance of the distributed overlay adaptation algorithm in dynamic environments, similar to the simulation scenarios in [13], as follows:

1) Churn: Initially, an $\frac{N}{2}$-node (677-node) overlay is built. There are N churn-events during the simulation period. A churn-event is either a single node joining with a probability of 0.5 or a single node leaving with a probability of 0.5. The expected network size after a sequence of events is $\frac{N}{2}$.
2) Shrink: Initially, an N-node (1355-node) overlay is built. Then, 30% of the nodes leave the system during the simulation period.

To simulate the dynamic of churn over time, we distribute all events uniformly over the simulation period, i.e., 40 minutes.[3] The query arrival pattern of each peer follows a Poisson distribution. Specifically, a random variable, X, is used to represent the interarrival times of two queries, and the probability distribution function of X is an exponential distribution with mean $1(/minute)$. When a peer fails to locate an object of interest, it re-issues the query after $1(/minute)$. Each query event is deleted until the request is matched. Because some peers may join or leave the P2P system, a peer that fails to locate an object in the current step may be able to find it in subsequent steps if new users holding the requested object

[3] We use the minute as the time unit. However, we believe that the trend of simulation results will be consistent as the time scale varies.

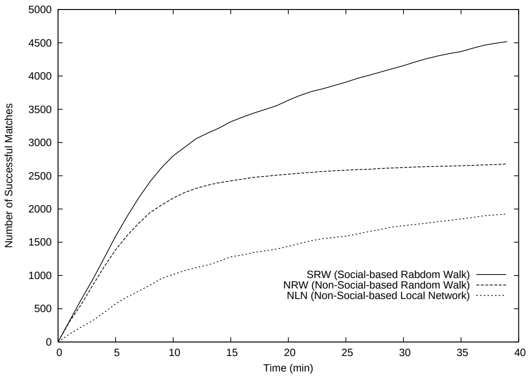

Fig. 5. Number of Successful Matches in Churn Scenario

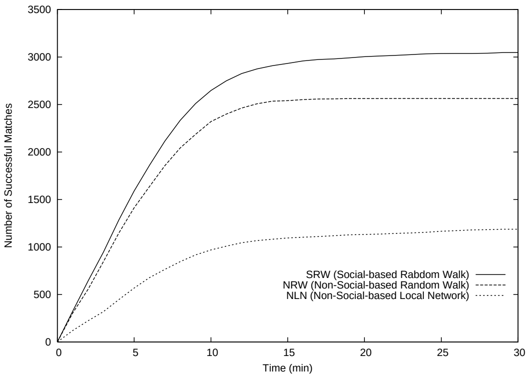

Fig. 6. Number of Success Matches in Shrink Scenario

join the system. In this simulation, we set the TTL to 5, and evaluate the performance of the overlay adaptation algorithms in terms of the cumulative number of successful queries over time.

Figures 5 and 6 show the number of successful matches in the churn and shrink scenarios respectively. In the churn scenario, there are at most about $2,000$ successful matches within 10 minutes in the non-social-based schemes, whereas the proposed distributed social-based overlay adaptation method generates $3,000$ successful matches within 10 minutes. This is because dynamic social-based adaptation allows each peer to update its buddy list, i.e., overlay links, if it or its original buddies change their listening habits. On the other hand, the non-social-based schemes can successfully match at most $2,700$ queries within the simulation period. Because the non-social-based overlay construction algorithm only updates the overlay links based on some random mechanisms, users can not locate objects of interest from their neighboring overlay nodes. In other words, TTL must be increased so that users can locate objects of interest by visiting more peers. Clearly, the social-based method also performs better than the non-social-based methods in the shrink scenario.

V. Conclusion

We have proposed a social-based overlay construction algorithm. We have also defined a user profiling method based on the characteristics of the objects held by each user, and proposed a distance measure to quantify the similarity between peers. The results show that a social-based overlay built according to the proposed similarity measure can improve the performance of the content query service in terms of the success ratio, precision rate, and recall rate. We have also proposed a random-walk-based sampling method to select buddies from unbiased sample candidates. Because the random walk method reduces the overhead of buddy selection significantly, each peer can maintain its overlay links distributedly and dynamically if overlay links fail or user preferences change. The simulation results also illustrate that, even in dynamic environments, the proposed social-based overlay adaptation algorithm can update the overlay topology dynamically and, thus, improve the efficiency of the content query service.

References

[1] R. A. Hanneman and M. Riddle, *Introduction to social network methods: Table of contents*, A. Oram, Ed. http://www.faculty.ucr.edu/ hanneman/nettext/, 2005.

[2] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making gnutella-like p2p systems scalable," in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003, pp. 407–418.

[3] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *IMC '06: Proceedings of the 6th ACM SIGCOMM on Internet measurement*, 2006, pp. 189–202.

[4] Gnutella: http://www.gnutella.com.

[5] O. Gnawali, "A keyword set search system for peer-to-peer networks," June 2002, master's thesis, Massachusetts Institute of Technology.

[6] P. Reynolds and A. Vahdat, "Efficient peer-to-peer keyword searching," in *Proceedings of International Middleware Conference*, Jun 2003.

[7] L. Liu and K.-W. Lee, "Keyword fusion to support efficient keyword-based search in peer-to-peer file sharing," in *CCGRID '04: Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid*, 2004, pp. 269–276.

[8] Y.-J. Joung, C.-T. Fang, and L.-W. Yang, "Keyword search in dht-based peer-to-peer networks," in *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, 2005, pp. 339–348.

[9] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Bhuberman, "Search in power-law networks," *Physical Review E*, vol. 64 46135, 2001.

[10] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," *Lecture Notes in Computer Science*, vol. 2009, pp. 46–66, 2001.

[11] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *SIGMETRICS '02: Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 2002, pp. 258–259.

[12] C. Law and K.-Y. Siu, "Distributed construction of random expander networks." in *INFOCOM 2003*, 2003.

[13] V. Vishnumurthy and P. Francis, "On heterogeneous overlay construction and random node selection in unstructured p2p networks," in *INFOCOM*, April 2006.

[14] J. A. Pouwelse, P. Garbacki, J. W. A. Bakker, J. Yang, A. Iosup, D. Epema, M.Reinders, M. R. van Steen, and H. J. Sips, "Tribler: A social-based based peer to peer system," in *5th Int'l Workshop on Peer-to-Peer Systems (IPTPS)*, February 2006.

[15] P. Androutsos, D. Androutsos, and A. Venetsanopoulos, "Small world distributed access of multimedia data: an indexing system that mimics social acquaintance networks," *Signal Processing Magazine, IEEE , vol.23, no.2pp*, pp. 142– 153, Mar, 2006.

[16] R. Baeza-Yates, B. Ribeiro-Neto, *et al.*, *Modern information retrieval*. Addison-Wesley Harlow, England, 1999.

[17] G. Salton, *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.

[18] Brite: http://www.cs.bu.edu/brite/.