Inferring Peer Centrality in Socially-Informed Peer-to-Peer Systems

Nicolas Kourtellis and Adriana Iamnitchi
Department of Computer Science and Engineering, University of South Florida, Tampa, FL, USA
nkourtel@mail.usf.edu, anda@cse.usf.edu

***Abstract*—Social applications implemented on a peer-to-peer (P2P) architecture mine the social graph of their users for improved performance in search, recommendations, resource sharing and others. In such applications, the social graph that connects their users is distributed on the peer-to-peer system: the traversal of the social graph translates to a socially-informed routing in the peer-to-peer layer.**

In this work we introduce the model of a projection graph that is the result of mapping a social graph onto a peer-to-peer network. We analytically formulate the relation between metrics in the social graph and in the projection graph. We focus on three such graph metrics: degree centrality, node betweenness centrality, and edge betweenness centrality. We evaluate experimentally the feasibility of estimating these metrics in the projection graph from the metrics of the social graph. Our experiments on real networks show that when mapping communities of 50-150 users on a peer, there is an optimal organization of the projection graph with respect to degree and node betweenness centrality. In this range, the association between the properties of the social graph and the projection graph is the highest, and thus the properties of the (dynamic) projection graph can be inferred from the properties of the (slower changing) social graph. We discuss the applicability of our findings to aspects of peer-to-peer systems such as data dissemination, social search, peer vulnerability, and data placement and caching.

I. Introduction

Socially-aware applications and services have leveraged out-of-band social relationships for diverse objectives such as improving security [1], inferring trust [2], providing incentives for resource sharing [3], and building overlays [4] for private communication. Online social information has been used to rank Internet search results relative to the interests of a user's neighborhood in the social network [5], to favor socially connected users in a BitTorrent swarm [6], and to reduce unwanted communication [7].

All such applications use social data of users, collected and managed in the form of a social graph within an application domain. A user's social data, which consists of the user's direct relations with other users in the application's social graph, could be stored on systems with various types of architectures. The two extremes are fully centralized, such as in Facebook or Twitter, and fully decentralized, with each user's information stored on the user's mobile device [8], [9], [10], [11]. There is a large spectrum in between, in which the social information can be stored on a decentralized, P2P architecture, where multiple users can have their social information mapped on the same peer [12], [13], [14], [15], [16].

The P2P architectural approach has significant benefits. It can provide better user control over privacy of social data compared to centralized systems and provides better service availability for social applications mining the social graph than mobile devices. However, it also poses the following question: *How does the topology of the social graph affect the routing in the P2P system?*

The answer to this question has direct consequences for the performance of the applications that mine a social graph distributed on a P2P system. For example, social search [5] is a method of connecting searchers to context-relevant content made available by their friends. A social search follows contextually relevant social edges over multiple social hops. Depending on the mapping of the requester's neighborhood on the P2P system, such a search could visit several peers, some more socially resourceful than others. Identifying the more resourceful peers—for example, in terms of the number of social connections between users mapped on different peers, formally measured as degree centrality—can improve significantly the search performance [17]. Similarly, a socially-aware information dissemination application can target the peers through which most of the social traffic passes for fast dissemination and high coverage. Consequently, the performance of applications that traverse the social graph is inevitably affected by the graph properties of *the projection* of the social graph on the P2P network.

This projection graph, which is the subject of this study, is an undirected, weighted graph whose nodes are peers contributed by users and responsible for a set of users in the social graph, and whose edges connect peers whose users are socially connected. The weights on the projection graph edges are the number of edges in the social graph that connect users mapped on the end peers. We focus on three representative metrics, known in social analysis as centrality measures: degree centrality, node betweenness and edge betweenness centrality. Degree centrality shows how many peers can be contacted directly with a message broadcast and is typically used to identify hubs. Node betweenness centrality quantifies the extent to which a peer controls communication over indirect routes and can be used to create data caches for reduced latency to locate data. Edge betweenness centrality quantifies how much a connection between peers is utilized during communications across separate parts of the network and can be used to enhance fault tolerance to malicious attacks by monitoring and blocking malware traffic.

In this paper, we investigate how the three measures in the projection graph correlate with the measures of the social graph. A social graph is typically slowly changing [18]: apart from infrequent events such as moving to a new place or joining a new community, people rarely change their social relations. The typical churn of a P2P system translates into a much more dynamic graph, and thus a dynamic projection graph. Being able to calculate (or estimate) these centrality metrics in the projection graph, independently of the rewiring of the P2P overlay but only based on the more stable social graph, can have significant benefits.

The contributions of this study are the following:

- It presents a formal model for the projection graphs in P2P systems.
- It extracts the analytical relations of the three social network metrics between projection and social graphs.
- It examines experimentally on real networks the association between projection and social graphs and estimation methods for the centrality metrics, when considering various configurations of multiple users storing data on a peer.
- It outlines a set of lessons that connect previous work on social graphs and P2P systems with the projection graph model and shows how our findings can be applied in the design of socially-aware applications and P2P systems.

The rest of the paper is organized as follows: Section II presents in more detail the projection graph model and Section III describes how this model applies to existing works in P2P systems. Section IV formally defines the social network metrics and their association between projection and social graphs. Section V describes the experimental methods used to extract projection graphs from real networks and Section VI presents the experimental results. We conclude this study in Section VII with a set of lessons and how they can be applied to existing social applications and P2P systems.

II. Projection Graph Model

We consider a social graph as an undirected and unweighted graph $G = (V_G, E_G)$, where V_G is the set of users and $E_G \subseteq V_G \times V_G$ is the set of edges that represent the social ties between users. The existence of an (unweighted) edge between two users u and v is denoted by $e(u, v) = 1$.

The projection graph in a P2P system emerges when the social graph is distributed on the P2P network. The projection graph is an undirected, weighted graph whose nodes are peers responsible for a set of users in the social graph and whose edges represent the social ties between the users mapped on different peers. We refer to a user u as "mapped" on a particular peer when the peer stores u's *social data* (the set of all edges in the social graph originating from u). The weight of an edge in the projection graph is given by the number of edges in the social graph that connect the users mapped on the end peers. Formally, a projection graph is represented by P=(V_P, E_P). V_P is the set of peers in the P2P network. For each peer $P_i \in V_P$, $P_V(i)$ is the set of users mapped on P_i. $E_P \subseteq V_P \times V_P$ is the set of edges in P, to which we refer to as P2P edges. A P2P edge $P_E(i, j)$ is the equivalent of the set of social edges between the users mapped on peer P_i and the users mapped on peer P_j. Consequently, $P_E(i, i)$ represents the set of edges between users mapped on the same peer P_i. The weight of an edge between two peers P_i and P_j is denoted by $E(P_i, P_j) = |P_E(i, j)|$.

In this model, a user's social data is stored on *at least* one peer, and each peer stores *at least* one user's social data. Each peer maintains the union of the social data of the users mapped on it. Depending on the social relationship of these users, the union can be anywhere from a disjoint set of edges, as proposed in [12], [13], [14] to a connected subgraph, as proposed in [16].

Figure 1 presents one such scenario, in which users A–O store their data on peers 1–5. The P2P edge $P_E(2, 4)$ has the weight $E(2, 4)$=3 given by the social edges $e(D, K)$, $e(D, L)$ and $e(D, M)$.

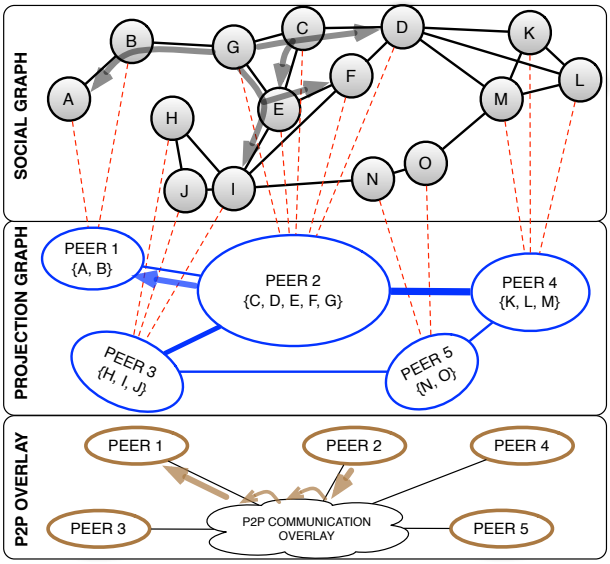

Fig. 1. An example of a social graph distributed on a set of peers which are organized in a P2P overlay. Users A–O are shown in small grey circles, peers 1–5 are shown in large blue circles in the projection graph and brown circles declare the peers organized in a P2P overlay. Users are connected with social edges illustrated with black lines, blue lines correspond to P2P edges with different weights (declared by their width) and red dashed lines correspond to mappings of users onto peers storing their data.

The projection graph is independent from the P2P overlay, as explained in the following. An application searching for the 2-hop neighborhood of user G will traverse the graph over the social connections this user has with the rest of the users (black arrows following social edges in the social graph, Figure 1). Since the social graph is distributed on top of a P2P network, these requests will be routed from peer to peer in a manner informed by the topology of the social graph. Therefore, the traversal of the social graph dictates peer 2 sending a message to peer 1 (blue arrow in projection graph) to request information regarding the 1-hop connections of user B. This application request might translate into multiple routing hops between the peers in the P2P communication overlay (e.g.,

DHT) before the destination peer is located and the request is delivered (brown arrows in the P2P overlay). We call such P2P systems *socially-informed* because the communication pattern between peers is determined by the topology of the social graph and how it is projected on the peers, and can be seen independently of the P2P overlay.

III. Related Work

The management of social data in a P2P architecture has been addressed in systems such as PeerSoN [12], Vis-à-Vis [13], Safebook [14], LifeSocial.KOM [15] and Prometheus [16]. In some cases (PeerSoN, Vis-à-Vis, Safebook, LifeSocial.KOM), the information of a user is isolated from other users, and peers access them individually. Thus, the social graph is fragmented into 1-hop neighborhoods, one for each user, and distributed across all peers, with potentially multiple fragments stored on the same peer that cannot be combined. In contrast, in Prometheus, a peer can mine the collection of social data entrusted to it by a group of (possibly socially connected) users. In all these systems, regardless of the way peers are organized in the P2P architecture (e.g., in a structured or unstructured overlay), the projection graph model can be applied for studying and improving the routing in a socially-informed application.

Other systems directly reflect the topology of the social graph of their users. Turtle [4] uses trust relationships between users to build overlays for private communication and preserve the anonymity of its users. F2F [19] uses social incentives to find reliable storage nodes in a P2P storage system. Sprout [20] enhances the routing tables of a Chord DHT with additional trusted social links of online friends, to improve query results and reduce delays. Tribler [6] allows users that are socially connected and participate in the same BitTorrent swarms to favor each other in content discovery, recommendation and downloading of files.

In other works such as [21], peers are organized into social P2P networks based on similar preferences, interests or knowledge of their users, to improve search by utilizing peers trusted or relevant to the search. Similarly, in [22] a social-based overlay for unstructured P2P networks is outlined, that enables peers to find and establish ties with other peers if their owners have common interest in specific types of content, thus improving search and reducing overlay construction overhead. In [23], P2P social networks self-organize based on the concept of distributed neuron-like agents and search stimulus between peers, to facilitate improved resource sharing and search. In such systems, the peers form edges over similar preferences of their owners or search requests, reflecting the P2P edges in the projection graph model. Thus, they implicitly use this model to organize the peers into P2P social networks.

Relevant to our work is the notion of the *group reduced graph*, as sketched in [24], where groups of users are replaced by a single "super" vertex whose neighborhood is the union of the neighborhoods of all group members. This technique allows for easier construction of group centrality measures, but a basic challenge, as stated in [24], is to justify the removal of internal social links in a group. However, in our instance of the reduced model, as will become more clear in Section V, this is not a problem: members of a group can store their data on the same peer which has complete knowledge of the social subgraph formed by the union of individual users' social data.

Studies such as [24] and [25] analytically discuss the betweenness centrality of a group of nodes by computing shortest paths between nodes outside the group, that pass through at least one node in the group [24] or all the nodes in the group [25]. Similarly, we study the betweenness centrality of peers representing groups of users. However, we assume that *all* users are mapped on peers (groups) and compute the peer betweenness centrality based on shortest paths between users mapped on different peers only.

IV. Social Network Centrality Measures

This section formally defines the degree centrality, node betweenness centrality and edge betweenness centrality for a social graph and its corresponding projection graph. For each measure, we study the connection between the social and projection graph and formulate research questions that we answer experimentally. In the following, we assume that multiple users can be mapped on the same peer and a user can be mapped only on one peer.

A. Degree Centrality

The degree centrality $C_D(n)$ of a node n in a graph is the number of edges that n has with other nodes. The degree centrality of a user u mapped on peer P_i can be expressed as the edges that u has with users on different peers than P_i, and the edges that u has with users mapped on the same peer P_i:

$$C_D(u) = \sum_{\substack{v \in P_V(j), \\ P_j \neq P_i \in V_P}} e(u,v) + \sum_{u \neq v \in P_V(i)} e(u,v), \forall u \in P_V(i) \tag{1}$$

We can express the degree centrality of a peer P_i as a function of the sum of the degree centralities of the users mapped on P_i, the sum of edges between users mapped on P_i and the sum of edges between users of peers P_i and $P_j, \forall P_j \neq P_i \in V_P$:

$$C_D(P_i) = \sum_{u \in P_V(i)} C_D(u) - \sum_{u \neq v \in P_V(i)} e(u,v) - \sum_{P_j \neq P_i \in V_P} \Big(\sum_{\substack{u \in P_V(i) \\ v \in P_V(j)}} e(u,v) - 1 \Big) \tag{2}$$

Equation (2) allows us to analytically calculate the exact degree centrality of a peer if the peer can access its users' social connections and infer its P2P edges with other peers. However, it is generally difficult to determine the exact degree centrality of a peer when the peer is granted access to view only a user's degree centrality score but not the user's connections. Thus, a research question is:

Question 1: *How well can a peer estimate its degree centrality in the projection graph, based only on the degree centrality score of its users in the social graph, and what parameters affect the accuracy of this estimation?*

B. Node Betweenness Centrality

Betweenness centrality $C_{NB}(u)$ of a user $u \in V_G$ is the sum of fractions of shortest paths between users s and t that pass through user u, denoted by $\sigma(s,t|u)$, over all the shortest paths between the two users, $\sigma(s,t)$:

$$C_{NB}(u) = \sum_{s \neq t \in V_G} \frac{\sigma(s,t|u)}{\sigma(s,t)} \tag{3}$$

Betweenness centrality $C_{NB}(P_i)$ of a peer $P_i \in V_P$ is the sum of fractions of *weighted* shortest paths between peers P_j and P_k that pass through P_i, denoted by $\lambda(P_j,P_k|P_i)$, over all the *weighted* shortest paths between the two peers, $\lambda(P_j,P_k)$:

$$C_{NB}(P_i) = \sum_{P_j \neq P_k \in V_P} \frac{\lambda(P_j,P_k|P_i)}{\lambda(P_j,P_k)} \tag{4}$$

The shortest paths between users can be divided into four categories, as illustrated in Figure 2. The first category reflects the shortest paths between two users s and t that pass through u and each user is mapped on a different peer. The second category (and its omitted symmetrical for $u \in P_V(i)$, $t \in P_V(i)$, $s \in P_V(k)$) reflects the shortest paths between s and t, when one of them is mapped on the same peer as u. The third category reflects the shortest paths between s and t when they are mapped on the same peer P_j, but different from u. The forth category reflects the case that all three users are mapped on the same peer P_i.

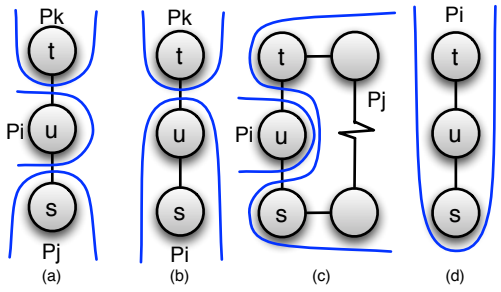

Fig. 2. The four categories of shortest paths between two users s and t through u, when users are mapped on peers.

Thus, we can express the betweenness centrality of user $u \in P_V(i)$ as follows:

$$C_{NB}(u) = \sum_{P_j \neq P_k \in V_P} \Bigg(\sum_{\substack{s \in P_V(j) \\ t \in P_V(k)}} \frac{\sigma(s,t|u)}{\sigma(s,t)} + \sum_{\substack{s \in P_V(i) \\ t \in P_V(k)}} \frac{\sigma(s,t|u)}{\sigma(s,t)} + \sum_{\substack{s \in P_V(j) \\ t \in P_V(j)}} \frac{\sigma(s,t|u)}{\sigma(s,t)} + \sum_{\substack{s \in P_V(i) \\ t \in P_V(i)}} \frac{\sigma(s,t|u)}{\sigma(s,t)} \Bigg) \tag{5}$$

We observe that only the first term of (5) reflects shortest paths that contribute to P_i's betweenness centrality. In fact, considering all possible pairs of users s and t, $\forall P_j \neq P_k \in V_P$, this term is essentially the portion of peer betweenness centrality of P_i in the projection graph contributed by user u.

As demonstrated in (5), it is difficult to analytically determine the peer betweenness centrality with respect to the centrality of its users due to the various types of shortest paths in which users participate. Also, the peer might not be granted access to traverse the P2P topology and calculate its exact betweenness centrality in the projection graph, for example due to user access policies on other peers or unavailability of peers. Assuming a peer is granted access to its users' betweenness centrality scores, a research question is:

Question 2: *How well can a peer estimate its node betweenness centrality in the projection graph, based only on the node betweenness centrality score of its users in the social graph, and what parameters affect the accuracy of this estimation?*

C. Edge Betweenness Centrality

Betweenness centrality $C_{EB}(e)$ of an edge $e \in E_G$ is the sum of fractions of shortest paths between two users s and t that contain e, denoted by $\sigma(s,t|e)$, over all the shortest paths between the two users, $\sigma(s,t)$, or more succinctly:

$$C_{EB}(e) = \sum_{s \neq t \in V_G} \frac{\sigma(s,t|e)}{\sigma(s,t)} \tag{6}$$

Betweenness centrality $C_{EB}(E)$ of a P2P edge $E \in E_P$ is the sum of fractions of *weighted* shortest paths between two peers P_j and P_k that contain E, denoted by $\lambda(P_j,P_k|E)$, over all *weighted* shortest paths between the two peers, $\lambda(P_j,P_k)$:

$$C_{EB}(E) = \sum_{P_j \neq P_k \in V_P} \frac{\lambda(P_j,P_k|E)}{\lambda(P_j,P_k)} \tag{7}$$

As with the node betweenness, the shortest paths between users that contain edge e can be divided into five categories, as illustrated in Figure 3. We omit the illustration of the symmetrical cases for (a), (c) and (d).

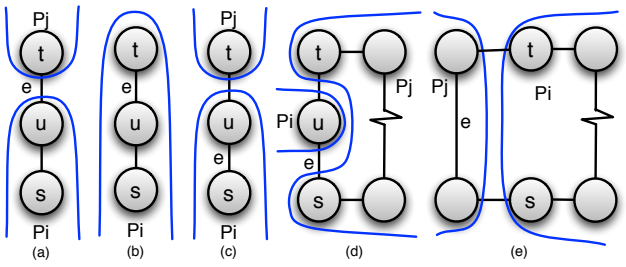

Fig. 3. The five categories of shortest paths between two users s and t through e, when users are mapped on peers.

Based on the intuition of Figure 3, we can express the betweenness centrality of an edge e as follows:

$$C_{EB}(e) = \sum_{P_i \neq P_j \in V_P} \Bigg(\sum_{\substack{s \in P_V(i) \\ t \in P_V(j) \\ e \in P_E(i,j)}} \frac{\sigma(s,t|e)}{\sigma(s,t)} + \sum_{\substack{s \in P_V(i) \\ t \in P_V(i) \\ e \in P_E(i,i)}} \frac{\sigma(s,t|e)}{\sigma(s,t)} + \sum_{\substack{s \in P_V(i) \\ t \in P_V(j) \\ e \in P_E(i,i)}} \frac{\sigma(s,t|e)}{\sigma(s,t)} + \sum_{\substack{s \in P_V(j) \\ t \in P_V(j) \\ e \in P_E(i,j)}} \frac{\sigma(s,t|e)}{\sigma(s,t)} + \sum_{\substack{s \in P_V(i) \\ t \in P_V(i) \\ e \in P_E(j,j)}} \frac{\sigma(s,t|e)}{\sigma(s,t)} \Bigg) \tag{8}$$

We observe that only the first and forth terms of (8) reflect shortest paths that contribute to the betweenness centrality of a P2P edge connecting peers P_i and P_j. In fact, when we consider all possible pairs of users s and t, these two terms are essentially the portion of the edge betweenness centrality between two peers contributed by a social edge e. Using similar argumentation with the node betweenness centrality, a research question is:

Question 3: *How well can we estimate the edge betweenness centrality of a P2P edge, based only on the edge betweenness centrality scores of its social edges, and what parameters affect the accuracy of this estimation?*

V. Projection Graph Topology

In order to answer the research questions from Section IV, we used five real networks to extract projection graph topologies with varying number of users per peer. Table I presents a summary of these networks, which cover diverse application domains, such as file sharing (gnutella) (from [26]), email communications of company employees (enron), trust on consumer reviews (epinions) and friendships in a news website (slashdot) (from [27]) and have sizes between $10K$ and $100K$ nodes. Even though the gnutella networks are not social networks like the other three, we use them because they exhibit social properties [26] and we have two instances of different sizes, enabling us to study the variation of the social network metrics with the network size. We consider all networks undirected and unweighted and used only the largest connected component (LCC) from each graph to ensure reachability between all pairs of users and peers.

TABLE I
Summary information of the real networks used

Network (abbreviation)	Number of Users LCC (original)	Number of Edges LCC (original)
P2P-Gnutella04 (gnutella04)	10,876 (10,876)	39,994 (39,994)
Email-Enron (enron)	33,696 (36,692)	180,811 (183,831)
P2P-Gnutella31 (gnutella31)	62,561 (62,586)	147,878 (147,892)
Soc-Epinions1 (epinions)	75,877 (75,879)	405,739 (405,740)
Soc-Slashdot0922 (slashdot)	82,168 (82,168)	504,230 (504,230)

We use the observation that people naturally form social communities with strong social bonds that translate into incentives for sharing within the community. We thus assume that there exists a community peer (provided by a member of the social community) that stores the social data of all community members. To study the properties of projection graphs of peers, we first clustered users into social communities and then mapped each community to a peer. Social edges of users connecting different peers were transformed into weighted P2P edges. Communities were identified by utilizing a modified algorithm of the Louvain method [28] for fast community detection in large networks. This method first splits users into very small communities, and then iteratively reassigns users to other communities and merges them in order to improve the overall modularity score. The modularity of a partition in a graph measures the density of the links inside communities as compared to links between communities. The Louvain method detects communities very fast, even for graph sizes in the order of millions of users, with a very wide range of community sizes, i.e., a lot of small groups of 2–10 users, as well as very large groups in the order of 1000s users. The largest community usually represents the *core* of the network whereas the smallest ones, loosely connected to the core, reflect the *whiskers* of the network [29].

Since we consider that a community is mapped on a peer contributed by a user, it would be unrealistic to map a very large community on one peer. Thus, we consider the communities exceeding a *max-size* as individual subgraphs and recursively apply the Louvain method on these subgraphs. We call this technique "Recursive-Louvain" and tested it with values for the *max-size*=10, 100, 500 and 1000. We used *max-size*=100 as it offered a local minimum for the standard deviation of size of communities. The value of *max-size*=100 supports the findings in [29] according to which the best communities with respect to *conductance* are relatively small, with sizes up to 100 users per community. We compare in Table II the summary statistics of the formed communities with the Louvain and Recursive-Louvain methods for *max-size*=100. Using the Recursive-Louvain method we successfully split most of the larger communities into smaller ones (4 to 50 times smaller) and improve the overall standard deviation of the size of communities formed from each network (some cases 6 to 30 times smaller).

TABLE II
Summary statistics for communities identified with Louvain (L) and Recursive-Louvain (RL) methods

Social Network	Number of Comm. L / RL	Avg. Users per Comm. L / RL	Standard Deviation L / RL	Min/Max L (RL Max)
gnutella04	2384/3013	4.0/3.6	23.0/3.5	2/1299 (89)
enron	2434/4303	11.9/7.6	139.1/15.7	2/4845 (1204)
gnutella31	13425/14385	4.4/4.3	3.0/2.8	2/3594 (97)
epinions	8481/16404	7.1/4.6	196.4/7.9	2/15770 (484)
slashdot	6879/18846	9.5/4.3	225.2/6.9	2/17012 (358)

As presented in Section IV, we want to estimate the three social network measures with only local information on peers and P2P edges. The parameter we consider that affects the most this estimation is the number of users mapped on peers. Thus, in our experiments we vary the average size of communities mapped on peers. Using the Recursive-Louvain method we identified a set of communities with fairly small average size (about 4–5 users), which were used as a baseline for our experimentation with increasing average size of communities. To produce communities with increasing number of users, we incrementally merged the smallest, socially-connected communities, until we reached the desired average number of users per community (and thus peer) in the range of $10, 20, ..., 1000$ users/peer. This merging process finds support from [29] where it is suggested that small communities can be combined into meaningful larger ones.

Figure 4 presents the rank distribution of the size of communities formed by this process, for the five networks

studied and for various average community sizes, ranging from 5 to 1000 users/peer. The average community size for all networks for the range of 5–100 users/peer exhibits a $Zipf$ distribution with two main exponents. The first one describes the size of community among the top 10 ranked sizes. The second one describes the rest of the sizes ranking lower. The $Zipf$ distribution applicable in this range of community sizes shows that the communities formed maintain a social structure of power-law nature observed in communities [30]. When the average community size is increased above 100–200 users/peer, the $Zipf$ distribution is no longer applicable, as the communities become more uniform.

VI. Experimental Results

As demonstrated by the expressions in Section IV, it is not easy to answer analytically the questions stated because of the various terms intercorrelated in the calculation of each centrality metric. Our experiments aim to answer these questions by examining how each of the centrality metrics for a peer depends on the number of users mapped on the peer and their cumulative centrality metric. To this end, we study the three metrics on the five real graphs and their extracted topologies, as explained in Section V.

A. Dependency on Number of Users per Peer

Figure 5 presents the rank distribution of the peer degree in the projection graph for different average size of community, for each of the five networks examined. The user degree rankings of the networks (points marked as "1") follow a $Zipf$ distribution demonstrating a power-law nature (especially the larger networks *epinions* and *slashdot*). Similar to the community size rankings, the networks exhibit a strong 2-exponent Zipf distribution when the size of communities increases from 5 to about 20–50 users per peer, meaning that the topologies inherit social structure from the social graph distributed on the peers. Beyond a community size of about 100 users, the topology becomes significantly uniform: most peers exhibit a similar degree, thus degree rankings show similar frequency. This effect intensifies as the average community size increases to 1000 users per peer.

Figure 6 compares the average degree centrality (DCP), node betweenness centrality (NBCP) and edge betweenness centrality (EBCP) for peers, with the respective cumulative centrality metric for users mapped on peers. Figures 6(a) and 6(b) show that the degree and node betweenness centrality of peers increase with respect to the average size of community. This means that adding more users to a community of a peer directly affects the centrality of the corresponding peer according to these metrics. We can explain this as follows: by increasing the size of the communities, we reduce their number. In effect, more users mapped on a peer translates to inter-peer social edges with new peers, thus more connectivity for the peer as well as opportunity to participate in more shortest paths. Figure 6(c) shows that the cumulative edge betweenness centrality of social edges between peers does not change for a range of size of communities. This is because when increasing the community size from 1 to about 50 users per peer, many social edges are mapped *within* the peers and not *between* peers. Within this range, the weighted edge betweenness centrality of the P2P edges decreases: the number of peers is reduced, but the edges connecting users are mostly within peers, thus the P2P edges lose importance (in terms of edge betweenness).

As the number of users mapped on the same peer increases, the degree and node betweenness centrality of peers reach a maximum point. For the degree centrality this can be seen at the point where the slope is steepest (e.g., the *gnutella04* topology reaches a maximum degree centrality at about 55 users per peer, whereas the *slashdot* topology at about 90 users per peer). From equation (2), when the average size of community per peer $\overline{|V_P(i)|}$ increases, the second and third terms increase as well but at a different rate than the first term, thus the difference of them becomes least at this maximum point. The network is optimally divided in communities mapped on peers which exhibit highest average degree and node betweenness centrality. For the edge betweenness, this is a turning point: between 50 and 100 users per peer, more social edges are mapped on P2P edges, in effect reversing the decline.

As the community size increases further, the peer degree centrality decreases rapidly to very small values (also verified by the flat distribution of peer degrees). Also, the opportunity for importance in the form of node betweenness is distributed uniformly across all peers since they start forming a very small, tightly connected clique. For the smallest network *gnutella04*, this drop takes effect quickly at about 60 users per peer, whereas for larger networks, like *epinions* and *slashdot*, at about 200–500 users per peer. At the same time, even though the betweenness centrality of P2P edges increases, the opportunity for importance is distributed evenly across very few P2P edges. Eventually, by increasing even further the community size, the peer degree reaches 0, since at that point all the users are mapped on one peer and this peer has no inter-peer edges. It is important to note that depending on the application domain, the network properties of the topology may vary, even for seemingly small networks such as the *enron* email graph in comparison to *slashdot* or *epinions* graphs.

B. Estimation of Centrality Measures

In Section IV we ask how well we can estimate the degree, node and edge betweenness centrality of peers and P2P edges when considering only local information, i.e., the cumulative scores of users (social edges) mapped on peers (P2P edges). Here we investigate within what range of number of users per peer this estimation maintains high accuracy. Figure 7 presents for each metric the correlation of the scores of peers and cumulative scores of users per peer, with respect to the average number of users per peer. Specifically, we calculate each correlation based on the tuple $\{A, B\}$ of scores per peer (edge): (A) the cumulative centrality of users (edges) mapped on a peer P_i (P2P edge E), and (B) the centrality of the corresponding peer P_i (P2P edge E) in the resulting topology. The correlation is calculated by taking into account the tuples

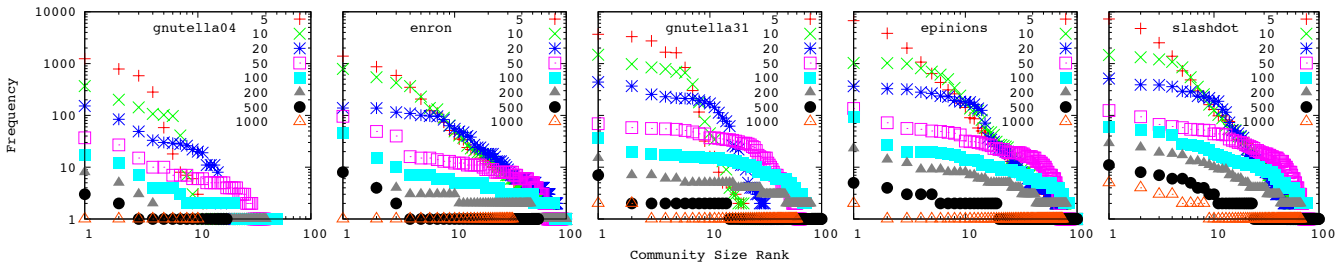

Fig. 4. Distribution of the community size rank vs frequency observed in the different average size of communities and different real networks.

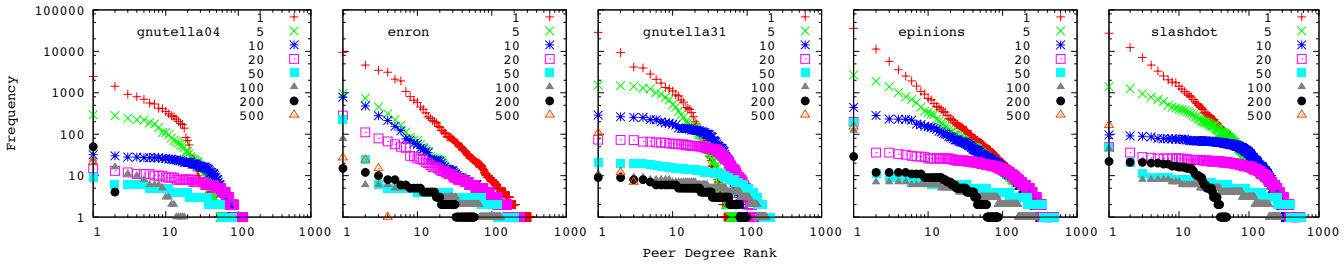

Fig. 5. Distribution of the peer degree rank vs frequency observed in the different average size of communities and different real networks.

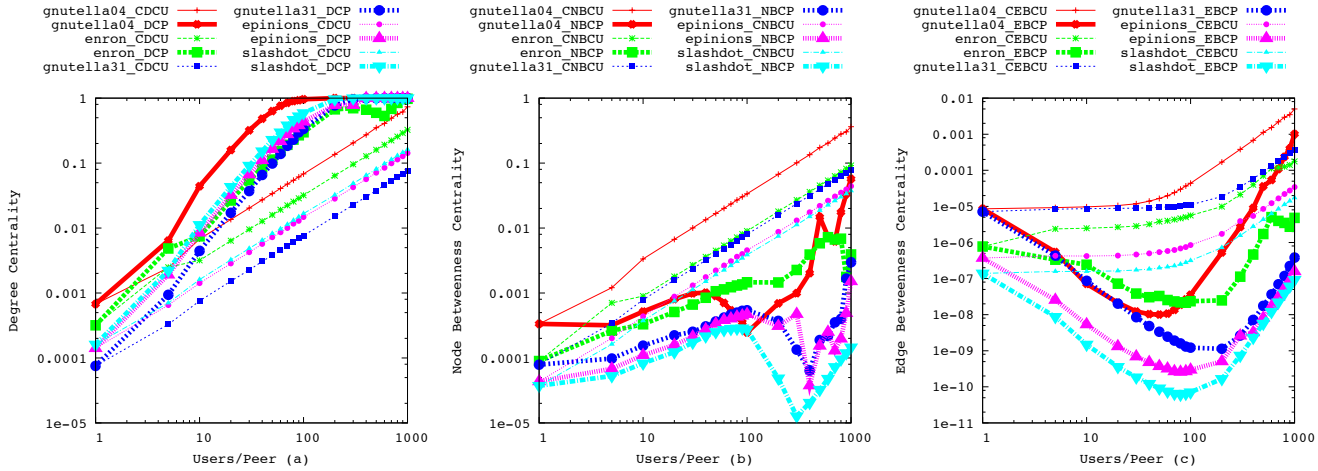

Fig. 6. Comparison of cumulative normalized scores of users (thin lines) vs average normalized scores of peers (thick lines) for Degree, Node Betweenness and Edge Betweenness Centrality.

across all peers (P2P edges) in the network, given a particular ratio of users per peer.

We observe that, for most of the networks, the correlation of the degree and node betweenness centrality remains fairly steady and high overall (> 0.8) for communities of size less than 100 users. From that point on, the correlation decreases rapidly. We can explain this drop by observing the reverse relationship between each respective pair of metrics (Figure 6). This trend is generally consistent across all sizes and types of real graphs, but some networks present an outlier behavior. For degree centrality, *gnutella31* maintains a high correlation up to 300 users/peer, and *enron* maintains high correlation up to 800 users/peer, despite being the 2nd smallest network. For node betweenness centrality, the two *gnutella* networks demonstrate some extended high correlation even up to 300 users/peer. The edge betweenness centrality drops significantly with the increase in community size, demonstrating that it is more sensitive to this parameter than the degree or node betweenness centrality.

C. Estimating Top-N% Scoring Peers

In very large social networks, calculating the exact node betweenness centrality of users can be intractable. Instead, we could use approximate measures such as κ–path centrality [31], to identify a small percentage of users that exhibit high betweenness in the social graph. Thus, we study the following question:

Question 4: *Given a set of users with relatively high node betweenness centrality, can we identify peers that exhibit similar high centrality in the projection graph?*

We investigate this question with two methods. In the first

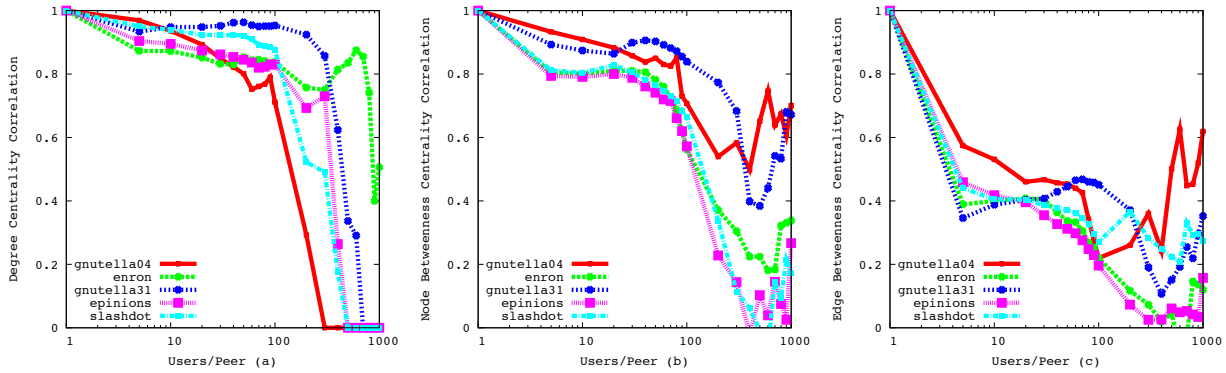

Fig. 7. Correlation of cumulative normalized centrality scores of users vs normalized centrality scores of peers for Degree, Node Betweenness and Edge Betweenness Centrality.

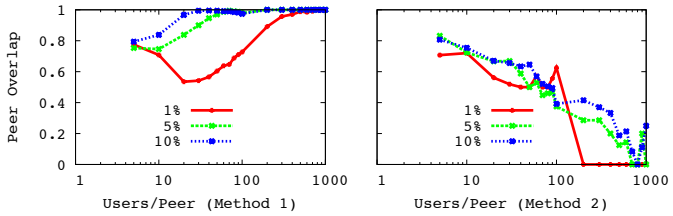

Fig. 8. Percent overlap of peers for node betweenness centrality in slashdot.

method, we pick users in top-$N\%$ betweenness centrality, and create a unique set U of their peers (set U, size k=$|U|$). Then, we pick k peers with the highest betweenness centrality (set P, size k=$|P|$). Finally, we compare the sets U and P to find the overlap of same peers. In the second method, we pick communities in top-$N\%$ cumulative score of betweenness centrality, and create a set C of their peers (set C, size q=$|C|$). Then, we pick peers in top-$N\%$ betweenness centrality (set P, size q=$|P|$). Finally, we compare the sets C and P to find the overlap of same peers.

Figure 8 shows for the network *slashdot* the percent overlap of peers for the node betweenness centrality using the two methods described. The results for the other networks are similar and we do not include them for brevity. Using the first method and top-5% of users, we can achieve 80–90% accuracy in identifying important peers for communities which are within the range of 5–100 users per peer. Above this range, the topology becomes too densely connected and the node betweenness centrality of peers is uniformly distributed across a small number of peers. Also, users of high importance are usually socially close to each other [32] and since the first method identifies only those users, there is a higher probability these users are mapped together on just a handful of peers. The overlap shows we can accurately predict the corresponding top peers. The second method achieves high accuracy but for a tighter range of users per peer. In comparison to the first method, the second method degrades in performance when increasing the community size. This means the cumulative score of users is more sensitive to the size of grouping of users per peer than the individual scores of users.

D. Summary of Lessons

Our experiments revealed the following:

Lesson 1: The increase of the average community size has an immediate effect on the organization of the topology and thus on the social network measures of each peer. From our experiments we identify a turning point where the degree and node betweenness centralities of peers reach a maximum. Before this point is reached, the projection graph resembles more closely the social graph it maps. Thus, the correlation of social network metrics between users and peers is highest and the two measures for peers can be estimated with good accuracy by the cumulative scores of users. When this point is reached, the P2P network exhibits an optimal structuring and the opportunity for peers to influence information flows through them is maximal. After this point, the topology loses any social properties, becomes a highly connected clique and the peers acquire equal opportunity to participate in any social graph traversal.

Lesson 2: Users mapped on a peer reflect their importance in the social graph onto their peer in the P2P topology in two ways: either directly by connecting their peer with other peers (degree centrality), or indirectly by situating their peer on multiple shortest paths between other peers (betweenness centrality). For small and medium size communities, we observe high correlation between users and peers for both of these centrality metrics. Thus, the centrality scores of users acquired from local information available to peers are good predictors of the importance a peer will have in the network. In effect, this means a peer can estimate with high accuracy its importance in the projection graph without the need to traverse the P2P network, which might be difficult due to network size, peer churn and user data access policies on other peers.

Lesson 3: There are high betweenness social edges in the projection graph that control significant information flow between different parts of the network. However, the importance of a social edge reflected on the P2P network depends on the way the social edges are mapped in the topology. From our experiments we observe that when more social edges are mapped within peers instead of between peers, the importance of social edges between peers becomes less significant and the

estimation of edge betweenness centrality of P2P edges is less accurate.

Lesson 4: Top betweenness centrality peers affect system performance and security. However, deterministically identifying high betweenness peers in projection graphs is infeasible not only because of the networks' large scales but also because of their dynamic nature caused by high peer churn. Our experiments show we can identify with 80–90% accuracy the peers with high betweenness centrality, by knowing the *top* 5% betweenness centrality users in the social graph.

VII. Summary and Discussions

In this work we proposed the *projection graph*, a model for studying the network properties of a P2P system that hosts the social graph of its users in a distributed fashion. We represented analytically the relations between the social graph and the projection graph in terms of degree, node and edge betweenness centrality. Because the analytical expressions are heavily dependent on the topology of the two graphs, we studied experimentally the correlation between their centrality metrics. Our experiments showed that within a range of 50-150 users mapped on a peer, there is an optimal organization of the projection graph with respect to degree and node betweenness centrality. Furthermore, there is a high correlation between the properties of users and the peers that store their data, which degrades rapidly when the number of users per peer passes this optimal organization threshold. This correlation allows us to estimate with high accuracy the centrality of peers based on the centrality scores of users and their edges. The applicability of our findings is discussed below.

Social search: Targeting important degree peers in the projection graph can improve the performance of a social search. We have shown that the projection graphs exhibit power-law properties within a range of community size. By applying techniques such as [17], targeting important degree nodes in a power-law topology improves significantly the performance of a search. Our correlation results demonstrate that peers who store the social data of high degree users will also have high degree centrality in the projection graph and thus will be highly involved in the traversal of the social graph. By targeting these high degree peers during a search, we could decrease search time and increase breadth of search and diversity of results. This is particularly applicable to works such as [5], [21] and [22], where social search can be enhanced by estimating the peer degree centrality in the projection graph.

Monitoring and trace collection: Social applications can collect usage statistics or better provision high node betweenness peers. These peers are situated on many shortest paths between other peers (and their users) and a high portion of the workload from social applications will pass through them. The system could infer where these "hot-spot peers" will appear just based on the centrality metrics of the users mapped on those peers, with no need to analyze the dynamic P2P overlay or the projection graph. Identifying such hot-spot peers can be used for monitoring much of the socially-routed peer-to-peer traffic; for placing caches or replicas; and for avoiding bottlenecks by remapping high betweenness users onto a better provisioned peer.

Data dissemination: P2P applications can make use of high betweenness edges to monitor information flow between different parts of the network, place caches of data on the edges' end peers for faster dissemination, create caches of search results and indexes of data location for speedup of access and search. This result could be used by systems such as Tribler [6] to advance its mechanisms for caching metadata of user activity, content location and data placement for downloads between several communities of users.

P2P churn: The performance of a P2P application is inherently depended on its tolerance to churn. In the presence of high peer churn, users could change their storage peers for better data availability. Since social networks are less dynamic than P2P networks, the user social connections are typically stable. Thus, peers can estimate accurately their importance in the topology, based only on locally stored information, i.e., the centrality of their users, which, given the slow dynamic of social networks, can be computed infrequently.

P2P socially-aware network overlay: Overlay communication overhead can be reduced if peers infer and enhance their routing tables with important P2P edges in the projection graph. These P2P network paths can be used frequently within the context of a particular application traversing the social graph and can be explicitly defined and used in the construction of the P2P overlay. This idea could be embedded in systems that already build on a socially-informed design, such as Turtle [4] and Sprout [20], but instead of using *single* social edges between users, they could exploit the P2P edges between peers which represent *multiple* edges between groups of users. These edges can be used for faster and more secure peer discovery with reduced network hops.

Vulnerability to peer attacks: Central users are connected to each other more directly than average [32], which means they are more likely to be part of the same community. Consequently, they could be mapped on the same peer, thus increasing significantly the peer's centrality. These high centrality peers control much of the information flowing through the P2P topology and could, on one hand, be more vulnerable, and on the other hand, more effective if they participate in a malicious attack. Such peers can be targeted for quarantine in the early stages of a malware outburst or used to disseminate faster and more efficiently security software patches and emergency announcements.

Size of meaningful social communities: Our empirical study led, via a different methodology, to a previous result known as the Dunbar number. Dunbar [33] predicted that on average we can manage a meaningful social circle of about 150 friends, no matter how sociable we are. Also, people tend to self-organize in groups of around 150; above this number, social cohesion begins to deteriorate as groups become larger. In [29] the best communities with respect to conductance were found to be relatively small with sizes only up to about 100 users, thus agreeing with Dunbar's number. Above this size, the quality of communities is questionable as they start

blending more with the core of the graph. In this study, we show that for a particular range of the average number of users per peer, the projection graphs follow closely the properties of the social graph distributed on the P2P network. We find that independently of the network tested, within a range of about 50–150 users per community, the organization of the projection graph topology reaches an optimal point. This range agrees with Dunbar's number and leads us to believe that there is a high correlation between the way people organize within the social graph into social groups or communities and how the properties of such social organization can be embedded in the emerging projection graph topology.

We conclude this discussion with the following future work research questions: What is the best mapping of communities onto peers, and according to what social network metrics and application domains? Even more importantly, could a system self-organize to produce such optimal placement of communities? Finally, given a mapping of communities onto peers based on an existing social P2P system, what are the design characteristics of socially-aware applications that mine the social graph, while making use of the socially-informed peer-to-peer topology?

ACKNOWLEDGMENT

This research was supported by the National Science Foundation under Grants No. CNS 0952420 and CNS 0831785. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

[1] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: defending against sybil attacks via social networks," in *SIGCOMM'06 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*, Pisa, Italy, 2006, pp. 267–278.

[2] P. Maniatis, M. Roussopoulos, T. J. Giuli, D. S. H. Rosenthal, and M. Baker, "The LOCKSS peer-to-peer digital preservation system," *ACM Trans. Comput. Syst.*, vol. 23, no. 1, 2005.

[3] D. N. Tran, F. Chiang, and J. Li, "Friendstore: cooperative online backup using trusted nodes," in *1st Workshop on Social Network Systems*, Glasgow, Scotland, 2008, pp. 37–42.

[4] B. C. Popescu, B. Crispo, and A. S. Tanenbaum, "Safe and private data sharing with Turtle: Friends team-up and beat the system," in *12th Int. Workshop on Security Protocols*, Cambridge, UK, 2004, pp. 213–220.

[5] K. P. Gummadi, A. Mislove, and P. Druschel, "Exploiting social networks for internet search," in *5th Workshop on Hot Topics in Networks*, Irvine, CA, USA, 2006, pp. 79–84.

[6] J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. van Steen, and H. Sips, "Tribler: A social-based peer-to-peer system," *Concurrency and Computation: Practice and Experience*, vol. 20, pp. 127–138, 2008.

[7] A. Mislove, A. Post, P. Druschel, and K. P. Gummadi, "Ostra: leveraging trust to thwart unwanted communication," in *5th Symposium on Networked Systems Design and Implementation*, San Francisco, CA, USA, 2008, pp. 15–30.

[8] S. B. Mokhtar, L. McNamara, and L. Capra, "A middleware service for pervasive social networking," in *1st Int. Workshop on Middleware for Pervasive Mobile and Embedded Computing*, Urbana Champaign, IL, USA, 2009, pp. 1–6.

[9] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, "MobiClique: Middleware for mobile social networking," in *2nd Workshop on Online Social Networks*, Barcelona, Spain, 2009, pp. 49–54.

[10] E. Sarigol, O. Riva, and G. Alonso, "A tuple space for social networking on mobile phones," in *26th Int. Conf. on Data Engineering*, Long Beach, CA, USA, 2010.

[11] A. Toninelli, A. Pathak, A. Seyedi, R. Sepicys Cardoso, and V. Issarny, "Middleware support for mobile social ecosystems," in *2nd Int. Workshop on Middleware Engineering*, Seoul, Korea, 2010.

[12] S. Buchegger, D. Schiöberg, L. Vu, and A. Datta, "PeerSoN: P2P social networking: Early experiences and insights," in *2nd ACM Workshop on Social Network Systems*, Nuremberg, Germany, 2009.

[13] A. Shakimov, H. Lim, L. Cox, and R. Cáceres, "Privacy, cost and availability trafeoffs in decentralized OSNs," in *2nd Workshop in Online Social Networks*, Barcelona, Spain, 2009.

[14] L. Cutillo, R. Molva, and T. Strufe, "Safebook: a privacy preserving online social network leveraging on real-life trust," *IEEE Comm. Magazine*, Dec. 2009.

[15] K. Graffi, C. Gross, D. Stingl, D. Hartung, A. Kovacevic, and R. Steinmetz, "LifeSocial.KOM: A secure and P2P-based solution for online social networks," in *IEEE Consumer Communications and Networking Conference*, Las Vega, NV, USA, 2011.

[16] N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea, and A. Iamnitchi, "Prometheus: User-controlled P2P social data management for socially-aware applications," in *11th Int. Middleware Conference*, Bangalore, India, 2010.

[17] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, "Search in power-law networks," *Physical Review E*, vol. 64, no. 4, p. 046135, Sep 2001.

[18] J. Golbeck, "The dynamics of web-based social networks: Membership, relationships, and change," *First Monday*, vol. 12, no. 11, Nov 2007.

[19] J. Li and F. Dabek, "F2F: reliable storage in open networks," in *5th Int. Workshop on Peer-to-Peer Systems*, Santa Barbara, CA, USA, 2006.

[20] S. Marti, P. Ganesan, and H. Garcia-Molina, "Sprout: P2P routing with social networks," in *Current Trends in Database Technology, EDBT 2004 Workshops*, vol. 3268, 2004, pp. 425–435.

[21] S. J. Yang, J. Zhang, L. Lin, and J. J. Tsai, "Improving peer-to-peer search performance through intelligent social search," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10 312 – 10 324, 2009.

[22] C.-J. Lin, Y.-T. Chang, S.-C. Tsai, and C.-F. Chou, "Distributed social-based overlay adaptation for unstructured P2P networks," in *IEEE Global Internet Symposium*, Anchorage, AK, 2007.

[23] F. Wang and Y. Sun, "Self-organizing peer-to-peer social networks," *Computational Intelligence*, vol. 24, no. 3, 2008.

[24] M. G. Everett and S. P. Borgatti, "The centrality of groups and classes," *Journal on Mathematical Sociology*, vol. 23, no. 3, pp. 181–201, 1999.

[25] E. D. Kolaczyk, D. B. Chua, and M. Barthélemy, "Group betweenness and co-betweenness: Inter-related notions of coalition centrality," *Social Networks*, vol. 31, no. 3, pp. 190–203, 2009.

[26] M. Ripeanu, A. Iamnitchi, and I. Foster, "Mapping the Gnutella network," *Internet Computing, IEEE*, vol. 6, no. 1, pp. 50–57, 2002.

[27] J. Leskovec. (2011) Stanford large network dataset collection. [Online]. Available: http://snap.stanford.edu/data/

[28] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 10, 2008.

[29] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney, "Statistical properties of community structure in large social and information networks," in *17th Int. Conf. on World Wide Web*, Beijing, China, 2008.

[30] A. Aaron Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 066111, 2004.

[31] T. Alahakoon, R. Tripathi, N. Kourtellis, R. Simha, and A. Iamnitchi, "K-path centrality: A new centrality measure in social networks," in *4th ACM Workshop on Social Network Systems*, Salzburg, Austria, 2011.

[32] X. Shi, M. Bonner, L. Adamic, and A. C. Gilbert, "The very small world of the well-connected," in *19th Conf. on Hypertext and hypermedia*, Pittsburgh, PA, USA, 2008.

[33] R. Dunbar, *Grooming, gossip and the evolution of language*. Harvard University Press, 1998.