

A P2P Bibliographic Content Management System: Realizing Decentralized Infrastructure for Social Softwares

Anwitaman Datta

School of Computer Engineering, Nanyang Technological University, Singapore
anwitaman@ntu.edu.sg

Abstract. We introduce a peer-to-peer bibliographic data & content management system which we call *PBDMS*, both as a social information system and useful tool to collaboratively manage bibliographic content, as well as a proof of concept that peer data management systems (PDMS) can support social softwares. In recent years, Web 2.0 social networks have revolutionized and dominated the internet usage, where the end users are active content contributors, besides being consumers as they had traditionally been. Likewise, in the past decade, peer-to-peer systems have brought about a paradigm shift in the internet infrastructure itself. Nevertheless, so far, social softwares use client-server centralized infrastructure. Users's privacy, control over individual's content, besides the traditional arguments of cost and scalability of a P2P systems are all critical incentives in realizing social softwares using P2P infrastructure, besides the interesting problems that arise while trying to build a decentralized infrastructure for social networking. We elaborate the solved challenges including a Sybil attack resilient DHT - *SocialCircle* - built using exclusively social links, highlight the features of the current PBDMS prototype, as well as outline some of the outstanding research issues and planned development.

Key words: Social Information Systems Software, P2P, Sybil Attack, Secure DHT, Digital Library

1 Introduction

This paper derives its ideas from diverse areas, and likewise contributes to several ideas as well. From purely *application perspective* it describes *PBDMS*, an application for users to maintain personal digital library and manage references by annotating the content with reviews or rating or tagging the content based on their personal discretion and need. Following the trend of numerous web based social libraries [3, 7, 22], *PBDMS* also allows users to share their resources like reviews with others in a selected manner or collaborate in groups to create such content. Each user maintains her own social contacts (buddy list), and can decide which of her local content should (or not) be shared with which specific buddies, or if to make it accessible to the larger community. Users can also explore their buddies' social network subject to access rights granted by these buddies.

By focusing specifically on sharing scientific papers and personal reviews of the same, *PBDMS* serves as a collaboration tool - where a group of researchers working on a project together can share their personal collection of research papers, or summaries of others' works as well as collaboratively build the knowledge base for their project, and is on the lines with systems like Bibster [20]. But unlike *PBDMS*, Bibster lacks social networking capabilities.

From *systems perspective*, by emphasizing on a peer-to-peer infrastructure for social networking and collaboration, *PBDMS* breaks new grounds - demonstrating the feasibility of such applications in a decentralized setting. The challenges of building such an infrastructure are many [14] depending partly on the granularity of desired reliability and functionalities. We solve some of those - particularly by designing an underlying peer-to-peer infrastructure *SocialCircle* - which in turn derives inputs from the social network layer to realize stable and secure routing, indexing and storage functionalities.

While there are many outstanding issues both from distributed systems perspective as well as from the perspective of end users who arguably cares only on the maturity and usability of the software and not its inner workings, the current implementation of *PBDMS*¹ nevertheless is a decent and working

¹ For source code and downloading executable (but no decent user manual yet), please visit <http://www3.ntu.edu.sg/home/anwitaman/researchPBDMS.html>

proof of concept and built using interesting innovations as well as by atypical use of existing ideas. Thus the innovations themselves are not the only highlights. An equally important contribution of this paper is the way existing ideas are all integrated together to build a peer-to-peer infrastructure based social networking applications.

Like in any traditional file sharing P2P systems, PBDMS allows searching resources in the network. The resources to look for are diverse, including user generated content or meta-information about the users themselves, such as their latest IP address or finding new possible friends. The search can be carried out by using Gnutella style partial flooding techniques, as well as by browsing a person's social connections, as well as by using the underlying DHT, depending on the applicability of a mechanism for each kind of search. Unlike traditional P2P systems and similar to online social networks, PBDMS has a strong sense of user identity which is address independent, and allows users to establish social relations with other users, browse their libraries, subscribe to items or users and get notifications on updates in a manner similar to RSS feeds, as well as push information to specific users and leave offline messages. It is also a demonstration of an online social networking application which can be used at workplace [31].

2 Privacy in the age of social networks

The landscape of Internet usage has changed dramatically in recent years, both in the way the computers connected to the network interact as well as the way the end-users using these computers interact - with the Internet and with each other. On the *networking plane*, infused by the (somewhat infamous) success of P2P file-swapping softwares, the last decade has witnessed an increased emphasis of using resources available at the edge to perform tasks which would otherwise have heavily burdened any centralized infrastructure. Thus to say, there is an increased proliferation of peer-to-peer mechanisms to either replace, or more often supplement, the client-server paradigm.

At the *application plane*, with the advent of Web 2.0 and social networks, we witness end users participating not only as passive consumers of content provided by the web-sites (client/server), but also as a contributor creating content collaboratively with fellow users. Thus at a logical level, many of these Web 2.0 applications are inherently peer-to-peer in nature. Nevertheless, somewhat ironically, all current Web 2.0 applications rely on an underlying infrastructure based on the traditional client-server model.

When the user interactions are peer-to-peer in nature, and while there is such a proliferation of unrelated P2P systems and applications, it is natural to ask if and how to realize a peer-to-peer underlying networking infrastructure for Web 2.0 applications. Perhaps in this irony lies the opportunity for P2P to redeem itself. The almost only well known popular P2P applications besides Skype [32] are file-sharing [1, 4] and video streaming [30]. Similar to social networks [31], most peer-to-peer applications add little value to workplace.

Like many other technologies, file-sharing is susceptible to illegal use. The technology of P2P file-sharing, and even more generally the whole P2P paradigm has thus often been demonized. The question has often been, what is a legitimate P2P application? We believe that P2P infrastructure for Web 2.0 applications, particularly social networks, is one such crucial application where end users can benefit from using a P2P infrastructure. The match could not have been any better or more natural than when both the underlying network resources and infrastructure, as well as the content is provided and consumed by end-users.

Of course at this juncture it is legitimate to ask, why use a peer-to-peer infrastructure for supporting social networks, when the good old client-server architecture works fine. One can give the traditional arguments that P2P scales well, since a growing user base naturally brings in more infrastructural resources. This definitely can be a good incentive for people with good ideas but little money to support and expand overnight if their service's popularity increases. Also, if popularity declines over time, there is less exposure. However, given the success of numerous upstart online social networking sites, which have managed to scale well to not just millions but even hundreds of millions of users, the traditional scalability argument alone does not justify the hassles of a P2P infrastructure.

Even as social networking sites claim to grow their user base at great speeds, the paranoid among us have long been wary, and as people gradually get to understand the implications better [29, 19] and

get more pragmatic, privacy and ownership of personal data are becoming major concerns. Particularly privacy and protection from massive data-mining and “big-brotherly” treatment of the users by the social networking service providers. This is expected eventually to lead to a significant population of users, who while they would like to enjoy the benefits and fun of social networks, may also want to restrict access to their personal data not only from fellow users who happen to be strangers, but also from any “big brother”. This disaffected population is expected to be the early adopters of social networks which rely on a peer-to-peer infrastructure and encryption. In the long run, we believe that if comparable quality of service can be achieved, a significantly large, if not all users will indeed be inclined to use it. As an anecdote, one may consider email users who use encryption (like PGP). It may be a small fraction of all the people who use emails, but nevertheless, their need is genuine, and the population of such users is non-negligible. Besides privacy and other related security concerns, the P2P approach also provides content creators to execute greater control over their content, as well as avoid censorship either by the website owner, or censorship of the hosting website by a third party.

While some sites follow up with corrective measures because of users’ outcry, e.g. [13], and one may also argue about legislative solutions to protect users’ privacy, there is no guarantee that in the future the users’ data will not be misused. Likewise, relying on service providers’ infrastructure makes the users vulnerable to censorship as well as leaves them at the mercy of the service providers’ whims. For instance, if a service provider decides to charge users for a service which is currently free, then the user is forced to either pay up or else leave the network, and lose all the social contacts and information she has put in the system. Finally, using a P2P infrastructure ensures that the user retains the ownership of all data concerning her, both about her as well as created by her.

And the granularity of such information is diverse. For example, no one else including the service provider needs to know who all an user has contact with. It’s a breach of the user’s privacy in two ways. First of all, the service provider knows that the user has such a contact. Moreover even if the service allows the user to hide such a contact from her other contacts, the service provider additionally knows that the user wants to hide such a contact from the others. In a peer-to-peer infrastructure, no one besides the user gets to know such information.²

The primary objective of using P2P infrastructure for social applications is thus to have a system which makes it technologically harder (ideally, impossible) to violate the users’ privacy and large scale data mining or censorship, even while the users continue to enjoy the advantages of social networking.

A P2P approach seems promising to be the right technology to achieve both privacy and freedom of speech. For this reason, user-provided content and participatory media creation suit themselves better to a peer-to-peer rather than a client-server model. Another incentive for users to embrace such a model is to evade any constraints put by the service provider in the present or future (e.g. for the amount of storage space, or subscription fees, or service shut down).

There are numerous discussions³ and studies about the merits, limitations as well as challenges [14, 35] of realizing such a peer-to-peer infrastructure for social networking applications.

The decentralized systems design of PBDMS has been inspired as much by this motivation to build a peer-to-peer social networking application, as well by the motivation to build a social networking application which is specifically useful for collaboration at workplace, particularly academic collaboration in this instance.

Finally we will like to note the absolute need of an open source project for this kind of initiative, to allow public scrutiny that there is no trapdoor for collecting users’ data.

3 PBDMS overview

In recent years numerous Peer Data Management Systems (PDMS) have been proposed [36, 24], aimed at developing a generic decentralized data management component using end user resources, playing an analogous role for data management as TCP/IP does for the underlying networking stack. We speculate PDMS will form the underlying substrate for decentralized social softwares and information systems.

² Network traffic analysis of users’ internet usage is an orthogonal issue.

³ e.g., A whole W3C workshop dedicated to such discussions <http://www.w3.org/2008/09/msnws/>

Social networks are explicit or implicit intersecting cliques of users. Even as we try to map the social network on a P2P overlay network, it is also crucial to follow a design which respects the principle of network data independence [21], using a logical layer to separate the social links from another logical layer (PDMS) which takes care of data management issues. The PDMS is itself in turn separated from a physical layer consisting of a structured overlay network - providing functionalities like indexing, storage, routing, load-balancing, address-independent communication among others.

However, since nodes communicate directly only with its social contacts, a fundamental premise on which most structured overlays are built - that any peer can communicate with any other peer, is potentially broken if communication is done using only social links. Later in section 4.2 we explain how we overcome this issue by building a new class of DHT.

3.1 Basic reference management and social networking functions supported in PBDMS

Similar to peer-to-peer applications like BibSter and Edutella [20, 28], PBDMS allows reference management including maintaining local library of documents and universal meta information such as (in the case of scientific publications) venue and year of publication, etc. Additionally, similar to web based services like BibSonomy [22], CiteULike [3], CiteSeerx [2] and LibraryThing [7] PBDMS also allows creation, sharing and collaborative maintenance of content which can be uniquely contributed by end users. Such content include users' reviews of specific articles or books, as well as collaborative tagging or ratings. A detailed comparison and distinction of the various systems can be found in table 1 in the related works section 5.

As an example of sharing, people collaborating in a project can share their opinions and knowledge about or build a shared library comprising of relevant related works. Thus, users can set up multiple groups with possibly common members, and decide to share and manipulate specific objects together. Alternatively someone can share his review of an article with everyone who may be interested, in lines with public reviews; other users can in turn rate these reviews - all enriching academic collaboration and enhancing shared knowledge specifically of collaborators as well as possibly the whole community.

For obvious reasons, individual users should be able to determine the individuals with whom they want to share each such user generated content. Users can define granularity of sharing restricted to specific immediate friends or the whole network. Users can search, browse other users' public profile and shared library, tag or rate these digital resources collaboratively, subscribe to be updated about changes to specific items or changes made by any particular user, as well as send (or leave offline) messages. Users can also decide which of their social contacts should be visible to which of their other contacts.

Unlike traditional peer-to-peer systems which do not need or have strong sense of identity, PBDMS needs and supports social contacts, which requires rediscovery of these contacts even if their physical address change over time. This is achieved by using an underlying DHT called SocialCircle which is in itself built using only social contacts and is thus resistant to Sybil attacks. We use the DHT as a self-referential directory [12]. We will explain the mechanisms below in section 4.

Beside these core reference management and social and collaboration functionalities, PBDMS has several simple but useful features including export (import) of the relevant content to (from) bibtex format, as well as generating PDF report of the reviews. Compatibility with web based systems like BibSonomy or web based social networking sites to import friends are not technologically challenging and can be easily integrated, but is absent in the current implementation, as is NAT traversal.

3.2 PBDMS deployment and evaluation

There are many ways such systems can be evaluated. Experiments with such systems often include some synthetic workloads, etc. However, such measurements are much about the quality of implementation and the choice of specific scenarios chosen, and less about the concepts. What we do need in the longer run is fine tuning the current implementation, but the current prototype is more a demonstrator of feasibility. We have tried it out in small scale deployments among colleagues and students within the local network using upto 15 nodes and diameter of 4 in the social network, and the system works fine functionally, actually pretty decently and is stable. That has been our principal objective so far.

Usual suspects of performance evaluation such as whether the system makes the best use of bandwidth or storage space, as well as others like how reliable the system is, or how well the underlying DHT works (which has been experimented in simulations of much larger real and synthetic social networks) still needs further attention, but are not the principal focus of this work. The former are implementation details and will be influenced by artefacts, while the later like the reliability of underlying storage layer, etc. depend on the total number of users in the system - and like most peer-to-peer systems can work only if there is at least a critical population of users. The fact that many other peer-to-peer systems do work is insurance enough that PBDMS too can.

3.3 Joining PBDMS social network

Like in any peer-to-peer network, a peer in PBDMS needs to find some node to join the system first. To facilitate finding some random node in the system, a logically centralized directory service can be used. PBDMS supports access through OpenDHT [8] which has recently suffered from a lot of down time, as well as OpenLookUp [9]. An user can also type in an IP address and port number to connect to another person, if she knows someone in the system. Like in online social networks, users can search for names and need to at some point find and be accepted as friends by some other people in the system. This may also be through extrinsic mechanisms where the newly joining node's real world friend can tell her about the public key she needs to find in the system to connect to him. Likewise, once users form social connections with some users, they can browse friend of friend network (subject to access rights) to discover new friends. Easy messaging within the system, as well as possible use of extrinsic communication mechanisms facilitate users to communicate and check out potential friends before adding them, and storing each others' public key. Notice that this way of adding new friends is not much different from web based online social networks like Orkut [10]. How peers rejoin the system in later sessions is discussed below in Section 4.1.

4 Identity crisis in a large-scale decentralized systems

In large-scale decentralized systems, there are two distinct identity related issues that may arise.

4.1 Coming back online: What's logging in, in a decentralized system?

Firstly, because of node mobility and dynamic IP address assignments, nodes' physical/IP address change over time. Many traditional peer-to-peer networks and applications like file sharing and streaming often do not care about the specific peers, as long as they are connected to some peers. However other applications can benefit or even need to relocate an existing peer. This is particularly the case while deploying social networking applications on a peer-to-peer infrastructure.

A simple solution to this issue can be to use a logically centralized directory service storing the up-to-date peer-to-address mappings. However, given that many of the peer-to-peer networks themselves work as decentralized directory services, one can also imagine using the peer-to-peer network itself as a self-referential self-contained directory service to store meta-information about the participants, including their current physical address. This basic idea has been proposed independently (and varying in details) in several academic as well as commercial peer-to-peer networks, including P-Grid [12] structured overlay, Microsoft's Peer Name Resolution Protocol [23] and Skype [32].

The basic idea is to use a self-referential directory [12] based on a DHT formed by the peers themselves. We assume that a peer's public key P_{pub} is known to its social contacts from the previous interactions. Whenever a peer returns online it inserts its latest address signed with its' private key in the DHT corresponding to a DHT key derived from a globally known hashing of its public key. Likewise, any peer looking for a specific contact can search for the contact's public key, and discover its latest physical address. Moreover, a continuous query for the same can be left at the responsible DHT node, so that when a peer comes back online and reinserts its latest address, other peers interested in this peer can be directly notified without them having to query again. This is essential to support presence, since peers otherwise won't know when to query back.

Likewise the same key is used by the user to publish anything else, and the corresponding DHT node keeps track of subscriptions made by other users for this particular user. Similarly, the DHT is also used to store offline messages for the node, which it can retrieve when it comes back online. People sending offline messages need to encrypt it with the target's public key to preserve privacy.

Thus to summarize, when a peer comes online, it reinserts its latest address corresponding to its public key and signed with its private key, and also issues queries to locate the latest address of all nodes in its buddy list list, as well as retrieve back any offline messages for itself.

Discovering the last address inserted by a buddy is however not sufficient, since the buddy may have in the meanwhile gone offline, and some other peer may be using the same address. Since buddies know each others' public keys, its easy to verify each other's identity once an address is found.

Notice that persistence of the necessary information - such as the mapping, or subscription information - in the underlying DHT, as well as other performance issues like load-balancing, need to be taken care of at the DHT layer. This is essential to achieve network data independence [21]. The current implementation is yet to be evaluated and stress tested to ensure that this underlying layer is stable. However, as mentioned earlier, such evaluation and fine-tuning is somewhat orthogonal to the scope of the current presentation. We next describe the design of the underlying DHT, which, unlike traditional DHTs, do not require a fully connected underlying graph, and instead embeds the DHT on the social graph.

4.2 A Sybil proof DHT: SocialCircle

Another aspect of identity in decentralized settings is that users can create bogus identifiers. A major security threat in such systems is if a resource rich adversary creates many bogus identifiers - popularly known as Sybil attack [16] - then it can disrupt the functionalities of the system (denial-of-service), as well as more actively hurt the other genuine users. A practical approach to thwart Sybil attacks in decentralized systems is to exploit social relationships which exist between real people and harder to fake in large numbers. SybilGuard [39] is one such approach which proposes generic techniques to isolate (most of the) spurious identifiers and relationships from real people and their relations and protect distributed systems from Sybil attacks. However, it does not specifically say how these ideas can be used to build any specific decentralized applications. Particularly, it does not say how to use the ideas to secure a distributed hash table structured overlay.

Structured overlays, e.g., Distributed Hash Tables (DHTs) provide essential indexing and resource discovering in distributed information systems. Typically, structured overlays are based on enhanced rings, meshes, hypercubes, etc., leveraging on the topological properties of such geometric structures. The ring topology is arguably the simplest and most popular structure used in various overlays. In a ring based overlay network like Chord [33] nodes are assigned to distinct points over a circular key-space, and the ring invariant is said to hold if each node correctly knows the currently online node which succeeds it (and the one which precedes it) in the ring. The ring is both a blessing and a curse. On the one hand, an intact ring is sufficient to guarantee correct routing. Hence, historically, all existing structured overlays over circular key space have considered it necessary de facto.

Previous attempts have used social network links to bolster DHTs, e.g., Sprout [26], preferring social links whenever possible, but nevertheless requiring links to random nodes also. Such an approach still relies on using the untrusted links most of the time, but was arguably as good as it could get under the older paradigm of DHT designs, where a completely connected underlying graph and ring invariance were considered necessary.

In the recent years several radical DHT designs have been proposed, for example VRR [15] proposed for ad-hoc environments and Fuzzynet [17] designed specifically to avoid ring maintenance. Neither of these two rely on sanctity of a ring or fully connected underlying graph. We design the SocialCircle DHT by adapting and hybridizing ideas from these two DHTs. Inlined in the description of SocialCircle below, we also point out which of the features are derived from which of VRR or Fuzzynet respectively.

Virtual ring routing (VRR) is a DHT style overlay layer approach used to define the underlying network's routing mechanism. It is implemented directly on top of the link layer and provides both traditional point-to-point network routing and DHT routing to the node responsible for a hashed key, without either flooding the network or using location dependent addresses. While traditional DHTs take

for granted point-to-point communication between any pair of participating nodes, VRR extends the idea, using only link layer connectivity. Essentially this means that the VRR scheme relaxes the traditional DHT assumption of a completely connected underlying graph. Each node in VRR has an unique address and location independent fixed identifier, organized in a virtual ring, emulating Chord style network. Each node keeps a list of $r/2$ closest clockwise and counter-clockwise neighbors for the node on the virtual ring. Such a set of neighbors is called the node’s virtual neighbor set (*uset*).

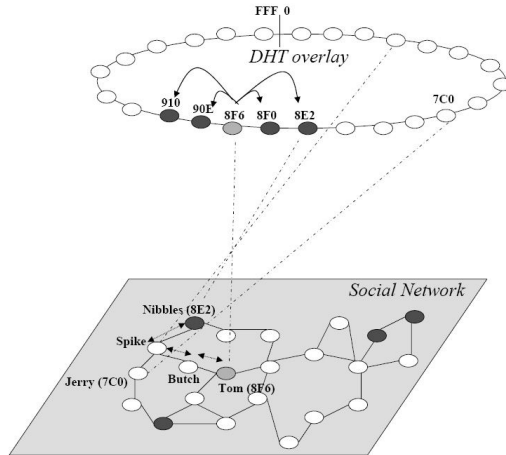


Fig. 1. Sybil attack resistant *SocialCircle* DHT exploiting social connections. This example of DHT over Tom & Jerry’s social graph is adapted from the virtual ring figure in [15] for routing in ad-hoc networks.

Typically, members in a node’s *uset* won’t be directly accessible to it through the link layer. Thus each node also maintains a second set called the physical neighbor set (*pset*), comprising nodes physically reachable to it through the link layer. In *SocialCircle*, we exploit this idea, and replace VRR’s *pset* with the set of friends a node has - its social set *sset*.

Thus, instead of exploiting the physical layer connectivity as VRR does, in *SocialCircle* we try to build the overlay over the *social plane* exploiting people’s social connections. In figure 1 the lower plane shows the social graph, while the upper plane shows the *SocialCircle* DHT. Techniques like SybilGuard [39] ensure that only legitimate links can be established at the social plane, thus providing protection from Sybil attacks.⁴ Adaption of VRR to exploit social links rather than physical neighbors provides a good abstraction, enabling us to realize a Sybil attack resistant DHT, where end-to-end routing can be achieved following a web or trust of friends-of-friends.

Finally, each peer maintains a routing table, which comprises of routes to its *uset* neighbors using its *sset*. These routes can be established and maintained using different strategies typically inspired by mobile ad-hoc routing protocols. Like in VRR, nodes in *SocialCircle* also keep track of the routes that pass through them. The advantage of using the DHT abstraction to do the routing over social graph is same as the use of DHTs instead of using flooding based search in a typical peer-to-peer system. The DHT abstraction ensures efficiency and certainty of routing to the appropriate target.

Thus, in the example from figure 1, *Tom* with logical identifier *8F6* on the *SocialCircle* has *8F0*, *8E2*, *90E* and *910* in its *uset*. *Spike* has *Jerry*, *Nibbles* and *Butch* in its *sset* since they are his direct social connections.

Tom needs to maintain routes to all its *uset* nodes, and thus, for *8E2*, he will have a route through his *sset* entry *Butch*, who will route through his *sset* entry *Spike*.

⁴ SybilGuard has not been implemented in our current *SocialCircle* implementation.

So when *Tom* needs to route a message to *7C0*, then it will try to forward the message closest to the target on the SocialCircle, which happens to be *8E2*. While the message is being routed to *8E2* following the *sset* nodes at each peer, *Spike* will observe that the ultimate destination is *7C0*, for which it may already have a route passing through it, and will thus forward the message to *Jerry*, instead of sending it to the intermediate destination *Nibbles*. *Jerry* processes the routing request, and forwards it to the final destination *Quacker*, who happens to have the identifier *7C0* on SocialCircle.

VRR works on such an opportunistic manner, where the route is forwarded along the virtual ring, but discovers shortcuts, so that the search is still efficient. SocialCircle preserves the same benefits by routing over the social links. Each hop on the social link involves IP level routing, which may need several hops, just like any logical overlay hop of traditional DHTs.

While the routing in SocialCircle follows the ideas from VRR, we use Fuzzynet’s data-management ideas [17] for storing and retrieving key-value pairs in SocialCircle. In contrast to traditional DHTs where data is necessarily stored over the consecutive nodes on the ring identifier space, Fuzzynet routes a “write” request on the DHT to arrive as close to the target as it can, and then gossips using the nodes’ links to store the data replicated at multiple nodes in the neighborhood, but not necessarily at the consecutive nodes. Lookup is done likewise. Such a non-deterministic (hence the name “fuzzy”) placement basically achieves the benefits of structured overlays - efficient search and of unstructured overlays - resilience, and has proven to be resilient against churn in comparison to traditional DHTs by orders of magnitude. Fuzzynet had less than 0.1% failures with experiments based on Skype peers’ availability, in comparison to up to 2-3% failures observed in traditional DHTs for similar level of churn.

We will also like to point out that there are two ways data and objects are stored by the applications themselves - (i) using the SocialCircle DHT’s placement scheme, which for example is used to store peers’ latest physical address or for indexing content in the network; (ii) using any other application/user level logic independent of the DHT, for example nodes may wish to use only their real world friends to back up data [37]. PBDMS uses both these approaches depending on the purpose, suitability and user discretion.

We have tried various artificial and real social connection data (for example from crawled online social network friend lists as well as PGP’s web of trust data) to evaluate SocialCircle, and the experiments suggest that the basic functions of routing, search and storage work well. A thorough evaluation of SocialCircle merits separate publication, and is under preparation.

5 Related works

The work presented here bears similarities with numerous and diverse domains. In Table 1 we compare PBDMS with numerous existing closely and loosely related systems.

There has been numerous DHT designs published in the last decade. However SocialCircle differs from most of them in that it uses only social connections to establish the DHT network, and is heavily motivated by recent innovations of DHT designs which do not require completely connected underlying graphs to establish a DHT [18, 15], and relies on techniques like SybilGuard[40] to provide security against denial of service attacks. The closest work to our approach was a different approach where Sprout DHT [26] was realized with a mix of social links as well as links to other nodes. In such an approach, most of the links and messages are however still arbitrary as in most other DHTs, and thus it is still vulnerable to sybil attacks.

Peer-to-peer paradigm has been proposed to distribute workload of citeseer [34] or for archival storage of digital documents [25]. These systems do not have any social component in their realization, and are classic p2p examples of using end node resources to distribute a task. A step closer to our approach are systems like ePost P2P mail system [27] and Friendstore [37] backup system, which both store user specific content - emails and backed up data respectively - by pooling resources in a peer-to-peer manner from the participating users. However, such data is accessed by only the owner, and there are no collaboration or social interactions in these systems. However, these systems, particularly ePost is a good proof of concept that a peer-to-peer infrastructure can be used for asynchronous communication among users. This is similar to the offline messages supported in PBDMS.

Importing buddies from traditional social networks is not technically challenging, and can be integrated in PBDMS in a fashion similar to [37], but has not been implemented yet. Likewise, from usage

System	Application	P2P or Web based	Identity persistence	Address independence	Persistent storage	DHT or Unstructured	User generated	Search/Browse sharing	Collaboration	Recommend	Cross compatibility
PBDMS	Ref. Mgmt.	P2P (social)	Strong	Yes	Partial	Both	Yes	Yes	Yes	Not yet	Partial
Bibster [20]	Ref. Mgmt.	P2P (random)	No	x	x	Unstructured	x	Yes	x	x	x
LibraryThing [7]	Ref. Mgmt.	Web based	Strong	Yes	Yes	x	Yes	Yes	Yes	Yes	x
BibSonomy [22]	Ref. Mgmt.	Web based	Strong	Yes	Yes	x	Yes	Yes	Yes	Yes	JabRef
CiteULike [3]	Ref. Mgmt.	Web based	Strong	Yes	Yes	x	Yes	Yes	Yes	Yes	x
CiteSeerx [2]	Ref. Mgmt.	Web based	Strong	Yes	Yes	x	Yes	Yes	Yes	Yes	x
Edutella [28]	Ref. Mgmt.	P2P/ Federation	No	x	x	Unstructured	Partly	Yes	x	x	No
Skype [32]	VoIP	P2P (both)	Strong	Yes	x	Unstructured	x	Partial	Partial	x	x
eMule [4]	File sharing	P2P (random)	Weak	x	No	Either	No	Yes	No	No	x
BitTorrent [1]	File sharing	P2P (random)	Weak	x	No	Tracker	No	Yes	No	No	x
Tribler [30]	Video	P2P (both)	Weak	Not yet	Partial	Unstructured	Partial	Yes	No	Taste buddies	?
YouTube [11]	Video	Web based	Strong	Yes	Yes	x	Yes	Yes	Yes	Yes	x
Orkut [10]	Social networking	Web based	Strong	Yes	Yes	x	Yes	Yes	Yes	x	x
Chord DHT [33]	Lookup	P2P (arbitrary)	No	x	x	x	x	x	x	x	x
Sprout DHT [26]	Lookup	P2P (both)	No	x	x	x	x	x	x	x	x
JabRef [6]	Yes	Local	x	x	x	x	Yes	Not yet	Not yet	x	BibSonomy
Friendstore [37]	Backup	P2P (social)	Strong	Multiple machines	Yes	x	x	x	x	x	Facebook
ePost [27]	Mail/Storage	P2P (arbitrary)	Strong	Yes	Yes	DHT	x	x	x	x	?
OverCite [34]	Distributed Server & Crawler	Web based	x	x	Yes	DHT	x	x	x	x	x
LOCKSS [25]	Archival Storage	Distributed	x	x	x	x	x	x	x	x	x
Google Scholar	Citation analysis	Web based	No	Yes	x	x	x	x	x	Yes	x

Table 1. Overview of and comparison of PBDMS with related works. By *user generated content*, we mean information which can uniquely be generated by the end user herself. Thus, universally known meta-information about a paper, even if filled in by the end user, does not qualify. However a comment or a rating or review is unique contribution by the user, and is treated as user generated content. If users' identity across multiple sessions persists and is used, only then address independence matters, otherwise it is not relevant. Some systems use the IP address to identify peers across session, in which case we say it has a weak context of identity. We use 'x' for attributes that are not applicable, while '?' for attributes we are unsure about. We mention 'not yet' for features that can be easily integrated using existing know-how, but is not yet implemented in the specific system. There are two aspects of web compatibility, namely with web based social networking sites, and with resources available online (e.g., .pdf documents). We have also included almost unrelated applications like traditional P2P file sharing such as eMule or BitTorrent as well as Web 2.0 video sharing sites like YouTube to distinguish and reinforce the notion of social networks in our work. For systems which are P2P in terms of users' participation, we also mention the kind of connections the peers have in the P2P network, which can be (i) random or arbitrary, (ii) social/real life connections or (iii) mix of both. If one compares PBDMS's features with a social networking website like Orkut or LibraryThing's features, it can be seen that there is a strong match for the aspects important for social networking, such as: user generated data, search, browse, collaborate, as well as kind of links each user has, viz social.

perspective, compatibility with web based systems is interesting and could be implemented in PBDMS. These are implementation issues and don't pose any research challenge.

There are numerous reference management softwares⁵. In functionalities, PBDMS differs from them primarily in the social nature of its utilization, by supporting sharing and searching of references as well as personal comments or other meta-information related to these references, and bears similarity to web based social libraries [3, 7, 22]. Our main contribution is to demonstrate the feasibility of such a social application using a P2P infrastructure, and PBDMS in its current form is not necessarily the best stand alone reference management software out there. Our next step is to take a more mature reference management software like JabRef [6] and integrate the social networking features of PBDMS to it, to enhance real usability of the PBDMS.

GPeerReview [5] focuses specifically on sharing reviews publicly in the form of endorsements to form a web of trust based reputation of scientific publications. PBDMS implicitly supports such function.

Tribler [30] is a very interesting project, which bridges the gap between peer-to-peer video streaming and Web 2.0 applications like YouTube. Tribler uses social context in various ways including allowing users of similar tastes to form ad-hoc communities of *taste buddies* in order to enhance the chances of discovering content of common interests, as well as allowing users to cooperate and coordinate with friendly peers in sharing bandwidth which is used as currency in the system in order to enhance the performance of the download process itself. The notion of taste buddies is promising in the context of PBDMS also, and we hope to integrate the same as part of future work. Still none of these notions of social collaboration require a strong coupling among peers. In contrast, the social network PBDMS builds has a much stronger notion of identity of individual users and social bonds between users, which requires persistence of this identifier across multiple sessions in an address independent way, and is achieved using techniques from our previous work [12]. Other approaches from Microsoft [23] and Skype [32] also exist. The Tribler website⁶ points out that similar mechanism using unstructured search is currently under investigation.

6 Future work and conclusion

Given the strong sense of identity in both the application layer at PBDMS and the underlying P2P infrastructure SocialCircle, other social mechanisms like reputation can be used to enforce or judge contribution of peers in the P2P infrastructure resources, as well as the quality of the content contributed by the users.

Thus at the networking layer, malicious behaviors like free loading, etc. can be thwarted, making the system robust. Likewise at the application layer, users have incentive to establish credibility by providing good content, helping build sustainable communities and knowledge base, and collaboratively filtering spam or spurious content. Such mechanisms are part of our future work.

The features for actual reference management in PBDMS were built from scratch and are limited. As a proof of concept it is fine, nevertheless, we want an application which can be used and so have started implementing and reintegrating these ideas with a mature open source reference management software JabRef [6].

Supporting other granularity of access control is one of the principal thrust of our work within the context of PeerSoN project.⁷ Another interesting aspects to study include data management aspects like support for richer queries, which will require both better data structures for storage - as well as more powerful and dedicated query language for querying social network data and mechanisms to process the same in a distributed manner. We are also currently exploring mechanisms to recommend team suitable to carry out a specific task, where suitability is determined by users' interests, competence as well as cohesion of the team as an ensemble, determined by the constituting members' social relationship. This is done in the context of mTeam project.⁸ Social networking and collaborative applications over a peer-to-

⁵ http://en.wikipedia.org/wiki/Comparison_of_reference_management_software

⁶ <http://www.tribler.org/trac/wiki/SocialOverlay>

⁷ www.peerson.net

⁸ <http://www3.ntu.edu.sg/home/anwitaman/researchMTeam.html>

peer infrastructure also exposes new challenges to maintain data consistency. We have separately carried out some theoretical work validated with only simulations on data consistency maintenance [38] in P2P networks. We need to implement and integrate the same, and potentially need to redesign the algorithms depending on the experience and lessons we learn when we try to implement them.

The current work demonstrates the feasibility of deploying social networking applications on a decentralized infrastructure. We argued that such a design is both natural as well as essential to meet privacy and data ownership needs of individuals in the era of online social networks. In order to realize a secure and trusted P2P infrastructure we propose a novel DHT design which relies on only social connections. We have implemented a working prototype of a social networking application running on such a peer-to-peer infrastructure. The application facilitates sharing and collaboration in managing bibliographic reference, and is expected to serve as an useful tool at workplace. We have come a long way from the traditional realm of peer-to-peer systems, and yet, we have just scratched the surface, and lot many challenges remain to be surmounted [14] to support social networking applications at a service quality comparable to current web hosted centralized ones, while delivering the promises and freedom of using a decentralized infrastructure.

Acknowledgments

I will like to acknowledge that part of the research presented in this paper has been supported by A*Star SERC Grant No: 0721340055.

I will also like to thank some of my undergraduate students whose contributions over the last two years have helped build the current PBDMS system, particularly Do Hoang Hai and Goh Chee Hong.

References

1. BitTorrent. <http://www.bittorrent.com/>.
2. CiteSeerx: Scientific Literature Digital Library and Search Engine. <http://citeseerx.ist.psu.edu/>.
3. CiteULike: Everyone's library. <http://www.citeulike.org/>.
4. eMule project homepage. <http://www.emule-project.net>.
5. GPeerReview. <http://code.google.com/p/gpeerreview/>.
6. JabRef reference manager. <http://jabref.sourceforge.net/>.
7. LibraryThing: Catalog your books online. <http://www.librarything.com/>.
8. OpenDHT. <http://www.opendht.org/>.
9. OpenLookup. <http://any.openlookup.net:5851/>.
10. Orkut. <http://www.orkut.com/>.
11. YouTube. <http://www.youtube.com/>.
12. K. Aberer, A. Datta, and M. Hauswirth. Efficient, self-contained handling of identity in peer-to-peer systems. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):858–869, July 2004.
13. Maria Aspan. Quitting Facebook Gets Easier, Feb. 2008. <http://www.nytimes.com/2008/02/13/technology/13face.html>.
14. Sonja Buchegger and Anwitaman Datta. A Case for P2P Infrastructure for Social Networks - Opportunities & Challenges. In *The Sixth International Conference on Wireless On-demand Network Systems and Services (IFIP/IEEE WONS 2009) special session on "Social Networks"*.
15. M. Caesar, M. Castro, E.B. Nightingale, G. O'Shea, and A. Rowstron. Virtual ring routing: network routing inspired by dhts. In *SIGCOMM, Proceedings*, 2006.
16. J.R. Douceur. The sybil attack. In *Peer-To-Peer Systems: First International Workshop, IPTPS, Revised Papers*. Springer, 2002.
17. S. Girdzijauskas, W. Galuba, V. Darlagiannis, A. Datta, and K Aberer. Fuzzynet: Zero-maintenance ringless overlay. Technical Report LSIR-REPORT-2008-006, EPFL, 2008.
18. Sarunas Girdzijauskas, Wojciech Galuba, Vasilios Darlagiannis, Anwitaman Datta, and Karl Aberer. Fuzzynet: Zero-maintenance Ringless Overlay. Technical report, 2008.
19. Jennifer Golbeck. Quechup: Another Social Network Enemy!, Sept. 2007. Oreillynet.com.
20. P. Haase, J. Broekstra, M.Ehrig, M. Menken, P.Mika, M. Plechawski, P. Pyszlak, B. Schnizler, R. Siebes, S. Staab, and C. Tempich. Bibster - A semantics-based bibliographic peer-to-peer system. In *ISWC 2004*.
21. J. M. Hellerstein. Toward network data independence. *SIGMOD Rec.*, 32(3), 2003.

22. Andreas Hotho, Robert Jschke, Christoph Schmitz, and Gerd Stumme. BibSonomy: A social bookmark and publication sharing system. In *Proceedings of the Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures*, 2006.
23. Christian Huitema and John L. Miller. Peer-to-peer name resolution protocol (PNRP) and multilevel cache for use therewith. United States Patent 7,065,587.
24. M. Karnstedt, K.-U. Sattler, M. Richtarsky, J. Muller, M. Hauswirth, R. Schmidt, and R. John. UniStore: Querying a DHT-based Universal Storage. In *ICDE 2007*.
25. Petros Maniatis, Mema Roussopoulos, TJ Giuli, David S. H. Rosenthal, Mary Baker, and Yanto Muliadi. Lockss: A peer-to-peer digital preservation system. *ACM Transactions on Computer Systems (TOCS)*, 2005.
26. S. Marti, P. Ganesan, and H. Garcia-Molina. Dht routing using social links. In *The 3rd International Workshop on Peer-to-Peer Systems*. Springer, 2004.
27. Alan Mislove, Ansley Post, Andreas Haeberlen, and Peter Druschel. Experiences in building and operating epost, a reliable peer-to-peer application. In *EuroSys '06: Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006*, 2006.
28. W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch. Edutella: A p2p networking infrastructure based on rdf. In *WWW 2002*.
29. Juan Carlos Perez. Facebook's Beacon More Intrusive Than Previously Thought, Nov 2007. <http://www.pcworld.com/article/id,140182-c,onlineprivacy/article.html>.
30. J. A. Pouwelse, P. Garbacki, J.Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, O. M. Reinders, M. R. van Steen, and H. J. Sips1a. Tribler: a social-based peer-to-peer system. *Concurrency and Computation: Practice and Experience*.
31. Maggie Shiels. Firms 'miss' social site success , July 2008. <http://news.bbc.co.uk/2/hi/technology/7501073.stm>.
32. Skype.com. Skype P2P telephony explained, 2004. <http://www.skype.com/intl/en/download/explained.html>.
33. I. Stoica, R. Morris, D. Liben-Nowell, DR Karger, MF Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *Networking, IEEE/ACM Transactions on*, 11(1):17–32, 2003.
34. Jeremy Stribling, Jinyang Li, Isaac G. Council, M. Frans Kaashoek, and Robert Morris. OverCite: A distributed, cooperative CiteSeer. In *3rd Symposium on Networked System Design and Implementation (NSDI'06)*, 2006.
35. Thorsten Strufe, Refik Molva, and Leucio Antonio Cutillo. Privacy Preserving Social Networking Through Decentralization. In *The Sixth International Conference on Wireless On-demand Network Systems and Services (IFIP/IEEE WONS 2009) special session on "Social Networks"*.
36. Igor Tatarinov, Zachary Ives, Jayant Madhavan, Alon Halevy, Dan Suciu, Nilesh Dalvi, Xin (Luna) Dong, Yana Kadiyska, Gerome Miklau, and Peter Mork. The piazza peer data management project. *SIGMOD Rec.*, 32(3), 2003.
37. Dinh Nguyen Tran, Frank Chiang, and Jinyang Li. Friendstore: cooperative online backup using trusted nodes. In *SocialNets '08: Proceedings of the 1st workshop on Social network systems*, 2008.
38. Zhijun Wang, Anwitaman Datta, Sajal K. Das, and Mohan Kumar. Cmv: File consistency maintenance through virtual servers in peer-to-peer systems. *Journal of Parallel and Distributed Computing*, accepted.
39. H. Yu, M. Kaminsky, P.B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. In *SIGCOMM, Proceedings*, pages 267–278. ACM New York, NY, USA, 2006.
40. Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. Sybilguard: defending against sybil attacks via social networks. In *SIGCOMM*, 2006.