My3: A highly-available P2P-based online social network

Rammohan Narendula, Thanasis G. Papaioannou, and Karl Aberer
School of Computer and Communication Sciences, EPFL, Switzerland
Email: firstname.lastname@epfl.ch

***Abstract*—As the online social networks (OSNs), such as Facebook, witness explosive growth, the privacy challenges gain critical consideration from governmental and law agencies due to concentration of vast amount of personal information within a single administrative domain. In this paper, we demonstrate a privacy-aware decentralized OSN called *My3*[1][2], where users can exercise full access control on their data. Our system exploits trust relationships in the social network for providing the necessary decentralized storage infrastructure. By taking users' geographical locations and online time statistics into account, it also addresses availability and storage performance issues.**

I. Introduction

Online social networks have recorded an unprecedented growth recently. For example, Facebook has more than 500 million active users with 50% of them visiting the portal daily. As a result, these OSNs have become store houses of vast amount of personal data in the form of messages, photos, links, and profile information. Moreover, most of the current social networks operate on infrastructure administered by a single authority (a.k.a, *big-brother*), which may perform mining of personal data hosted inside user profiles for business purposes, e.g., targeted advertisements. In a typical OSN, the users have no control or awareness on how their personal sensitive data is used by the service provider. Despite these, the exponential growth of the OSNs provides negligible motivation for the OSN operators to address the privacy concerns of the users. Often, such practices warrant intervention of governmental agencies [1]. In order to address the increasing privacy concerns of OSN users, big-brother-free alternatives (e.g. [2], [3]) have been proposed. Replacing the big-brother with a community of users, enables OSN users to have complete control on their profile content. In this paper, we demonstrate such a system, referred to as *My3*. A preliminary description of the system was first presented by the authors in [2]. As we demonstrate here, My3 makes a user's profile accessible to all of his friends in the social network (and only to them) even when he, himself, is offline, by means of appropriate replication schemes. Profile updates are propagated among replicas, so that eventual consistency is met. As opposed to completely centralized approaches (e.g. Facebook) or decentralized approaches that employ always-on personalized servers [3] for realizing online social networks, My3 represents a third (hence "3" in the name) alternative, which builds a decentralized OSN by replication schemes that exploit overlaps in online times of trusted friends.

[1]*Mythri* ("My3") is a Sanskrit word which means *friendship*.

[2]This work was supported by the Swiss Nano-Tera OpenSense project (Nano-Tera ref. 839_401).

II. System Overview

The *My3* system exploits the trust relationships among friends in the social network to improve the availability of the system. A user trusts some of his friends both for storing his profile content and for enforcing access control on the access requests for his profile. We believe that, leveraging mutual trust relationships for this purpose, simplifies the system to a great extent. Each user in the social network is required to run the My3 client[3]. The set of trusted nodes for a user u where his profile stored, is referred to as his *trusted proxy set (TPS)*. The TPS members for a user are properly selected with respect to the availability and performance goals based on the following observations: every user in an OSN has friends scattered over a limited set of geographical locations (e.g. his home town, working location, home country, location of previous institute etc.), and each user's online times are predictable to a large extent (e.g. online in his office hours, completely offline on weekends). The system exploits these facts in the TPS computation so that any friend of user u can access u's profile during the online time of that friend.

To this end, we assume that each user u in the social network is characterized by two parameters: his *geographical location* and his *online time period*. The geographical location determines the time zone of the user. The online time period represents the usual time that the user is online in the social network. This is the time window in which the user contributes his resources (i.e. bandwidth, storage, and processing power) for the social network operation and serves data access requests. We argue that two users can contact each other and thus exchange data if and only if their online time intervals overlap.

Distributed Hash Table: The *My3* system employs a distributed hash table (DHT) hosted at the resources contributed by the users. This DHT is used for storing meta information, e.g. the current IP address of a user's client. A user u and his TPS mapping is stored in the DHT in the form of *(key,value)* pair with *key* being the identifier of the user and *value* being the members of the TPS. This user-to-TPS mapping in the

[3]Client, node, and user all refer to the same throughout the text.

DHT is useful for contacting the nodes where the profile of a particular user is stored.

Online Time Graph: The online time graph for the user u contains all the user's friends as vertices. Edges can exist only between a friend and a trusted friend or two trusted friends if and only if there is an overlap in their online times. An online time graph is said to be *valid* if it is connected (then every friend of the user is connected to at least one of his trusted users) and the sub-graph induced by the trusted node set is also connected (then updates can propagate among trusted nodes only). Note that this graph is expected to be connected for a reasonable number of trusted friends; otherwise, an untrusted node should be included in the online time graph and encryption mechanisms should be used for communicating through this node.

Storage Layer: Here, we briefly describe the construction of the set TPS for a user u. It must be noted that each node in the TPS maintains one replica of the user's profile. One of the TPS members is assigned as a *mount point* for each friend of a user u. The user u's profile is said to be *mounted* on this trusted node w.r.t the corresponding friend. The friends of a user in the social network access the user's profile through these mount points. Note that, by definition, the mount point is available at some point in time during the friend's online time frame, so that he can access u's profile. However, a single-mount-point-per-user allows to access the profile replica only when that mount point is online. To increase the availability, we can use all the nodes in TPS as mount points. In this case, the above mount point would be the *primary mount point* and the remaining would be the *secondary* ones. We exploit location information of the friends of a user, in order to place data as close as possible to the nodes. This is quantified by a metric *access cost* between two locations, which can be defined in terms of communication latency between two nodes in these locations.

The set TPS and the mount points are computed from the *online time graph* based on one of the following objectives:

- *Minimize the number of replicas (MNR):* The MNR approach aims to minimize the storage and replica management overhead. It models the TPS construction as the problem of computing *minimum connected dominating set (MCDS)* on the online time graph of a user, with the additional constraint that only trusted nodes must be part of the set. A specific node from this set, with minimum access cost, is chosen as the mount point for a specific friend of the user.
- *Minimize the access cost (MAC):* The MAC approach minimizes the access cost incurred in accessing a user's profile. Hence, for every friend u, it assigns the nearest (i.e., with minimum access cost) trusted node connected to u in the online time graph, as the mount point. The set TPS contains all such mount points.

III. Demonstration

In this demo, we illustrate the following aspects of the My3 system, in addition to the usual activities users perform on an OSN, such as adding new friends, sharing a link, commenting on links, etc:

1) Storage Layer (TPS Construction): A user feeds his social graph and trusted friend set to the system, which invokes the storage layer computation module to construct the TPS and mount points for the friends in the social graph.
2) Profile Accessibility: A friend in the user's social graph accesses the user's profile through the mount point.
3) Update Propagation: An update done by a friend on a user's profile stored at a mount point, is propagated to the other mount points.
4) Eventual Consistency: Several concurrent updates done on a user's profile by friends through different mount points are propagated and the profile replicas stored at all the mount points will be *eventually consistent*.

Fig. 1. Visualization of the My3 system

Figure 1 shows the visualization of the My3 system in action. The left side window shows an example social network with 15 users with the edges representing the friendship relations. Users in the same time zone are marked with the same color. The right side window shows the corresponding online time graph for user 10. The trusted nodes are shown with a dark border, while the diamond shape node represents the TPS members. In this example, the MNR algorithm returned only a single TPS member as the mount point for all of user 10's friends. The slider helps to visualize the current status of the system at different time instances. Additional status information on various updates propagation can also be seen, which is not described for brevity reasons.

IV. Summary

In this demo, we presented My3: a P2P-based online social network. My3 offers accessibility to a user profile even when the user is offline, by means of two replication schemes with different design goals. The system exploits users' availability and trust relationships for computing the replication points.

References

[1] *http://mashable.com/2011/06/08/facebook-eu-probe/*.
[2] N. Rammohan, T. G. Papaioannou, and K. Aberer, "Privacy-aware and highly-available osn profiles," *19th IEEE International Workshops on Enabling Technologies Infrastructures for Collaborative Enterprises*, 2010.
[3] *https://joindiaspora.com/*.