SOUP: An Online Social Network By The People, For The People

David Koll
University of Göttingen
Göttingen, Germany
koll@cs.uni-goettingen.de

Jun Li
University of Oregon
Eugene, USA
lijun@cs.uoregon.edu

Xiaoming Fu
University of Göttingen
Göttingen, Germany
fu@cs.uni-goettingen.de

ABSTRACT

With increasing frequency, users raise concerns about data privacy and protection in centralized Online Social Networks (OSNs), in which providers have the unprecedented privilege to access and exploit every user’s private data at will. To mitigate these concerns, researchers have suggested to decentralize OSNs and thereby enable users to control and manage access to their data themselves. However, previously proposed decentralization approaches suffer from several drawbacks. To tackle their deficiencies, we introduce the Self-Organized Universe of People (SOUP). In this demonstration, we present a prototype of SOUP and share our experiences from a real-world deployment.

1. WHY DO WE NEED SOUP?

As online social network (OSN) providers deal with tremendous amounts of user information, they can obtain deep insights into their users’ personal communication, interests, opinions, and social relationships, which raises severe privacy and security concerns. Facebook, for example, has passed the one billion user mark and controls data of roughly one sixth of the world’s population. Still, the hunt for user data is not over, as showcased by the recent acquisitions of WhatsApp and Instagram in multi-billion dollar agreements.

While it remains to be seen whether OSN providers would give up a major source of income and grant comprehensive security and privacy means to their users, decentralized OSNs (DOSNs) pave a promising way for better user data security and privacy. Instead of relying on a central data repository, a DOSN can allow users to regain control over their data. For instance, they can encrypt their data and enforce access control to it according to their own need, and then distribute the data to specifically chosen storage sites.

However, recently proposed DOSNs introduce new shortcomings, ranging from limited success in providing high data availability to insufficient robustness, high management overhead, discrimination of users, a dependency on certain nodes, lack of data encryption, and deployability issues [3]. To address these drawbacks, we introduce a different DOSN approach called Self-Organized Universe of People (SOUP). In short, we provide the following contributions:

1) To achieve high data availability, we propose a new, generic approach to storing user data in a DOSN. In addition to enabling every user to store their data at their own machine, our mechanism selects other OSN participants as mirrors for the user’s data to make the data highly available, even if the owner herself is offline. SOUP does not rely on permanently available storage, although it can make an opportunistic use of such resources as they become available by quickly spreading relevant information among users.

2) To provide a robust OSN, SOUP ensures that regardless of participants’ social relations or online probabilities, data for all participants is highly available. Whereas previous works often limit a user in her mirror selection, we allow each user to select as many mirrors as required to make her data available. At the same time, SOUP ensures that there exist no more replicas than required to limit the overhead.

3) To achieve reliability and resiliency, SOUP is designed to be adaptive to the dynamics often seen in a DOSN. It can quickly respond to changes in the system and continue to provide high performance by favoring more recent observations in the network. Moreover, its operation is not significantly affected by malicious OSN users, as it can tolerate up to half of the links in the OSN being compromised.

4) To grant data privacy, SOUP offers effective mechanisms to encrypt the data and ensures that only eligible users can access (parts of) the data based on Attribute Based Encryption (ABE) [2].

A key component of SOUP that is critical to substantiate these contributions is how we select nodes as mirrors while addressing the critical challenges revealed by previous works. The selection process exploits recommendations and mirror feedback among friends to find the best suited mirrors for each user. We describe the procedure in detail in [3].

2. SOUP IN A NUTSHELL

Nodes (or users) in SOUP form a structured overlay, as shown in Figure 1. The overlay acts as an information directory and is based on a distributed hash table (DHT). The DHT enables efficient publish and lookup operations in a decentralized fashion, making a centralized information repository unnecessary. After joining the DHT via a bootstrapping node (e.g., $y$ in Fig. 1), every SOUP user can publish her directory entry at the node responsible for her ID in the DHT key space (e.g., $v$’s entry is published at $s$—Step 1 in Fig. 1), and any other node retrieve the entry (e.g., $u$ can look up $v$’s ID—Step 2). An entry contains a user’s
username, her unique SOUP ID, the interfaces via which she can currently be contacted, and the SOUP IDs of all the mirrors of her data.

Note, that a user only publishes a pointer (i.e., SOUP ID) to mirror nodes in the DHT (e.g., w publishes the mirrors at y—Step 3), whereas her data—protected by encryption and access control—is stored among nodes themselves. Storing the data in the DHT would result in a high management overhead in the presence of OSN typical churn rates [1]. Instead, every user maintains its own data, and selects a small set of other nodes as mirrors to store a replica of its data, in order to keep its data available even when it is offline. Data replication is necessary, as SOUP does not rely on a central repository and users are not permanently online.

If a user wants to communicate with another user, she looks up the partner’s entry and extracts the addressable interfaces to create a direct communication channel. The communication partners (u and v in Fig. 1) then exchange signed SOUP objects, which contain the OSN data (Step 4).

SOUP is also designed to be friendly to mobile nodes. As mobile nodes often experience high churn and long response times [1], SOUP exempts mobile nodes from the DHT and relays their DHT requests through relay nodes, which not only increases the stability of the DHT, but also saves resources at the mobile nodes.

On top of a SOUP node, applications will exploit the aforementioned features. SOUP provides a generic API to applications, and multiple applications can run concurrently on top of SOUP and share the same user data.

3. DEPLOYMENT EXPERIENCE

For both desktop and mobile Android devices we implemented (1) a SOUP middleware implementing the functionality shown in Figure 1 and (2) a demo application that uses SOUP and provides the user with a GUI. We show the interplay of the components in Figure 2. The demo application can create arbitrary application content, which is then encapsulated into SOUP objects before being transmitted to its destination, and thereby remains transparent to the middleware. Upon receiving application data, the reverse process is executed. As the underlying DHT, we use FreePastry.

We have deployed our prototype in a real-world DOSN consisting of 31 devices of which four were Android devices. Figure 3 shows the most bandwidth-intensive 20 minutes we observed for any user during multiple days of data collection. Messaging activities or simple profile requests are hardly distinguishable from an idle link. More intense features such as skipping through a photo album of a friend do not consume a regular user's bandwidth as well, as she takes her time to view the pictures. Note, that the traffic a user generates by consuming content is the same as in centralized OSNs, as the user needs to download the data in those systems as well. Only when producing content or acting as a mirror, SOUP generates additional traffic. As a consequence, in Figure 3, the link is most utilized at the creation of a photo album, as the data has to be distributed to the mirrors as well.

4. REFERENCES


1http://www.freepastry.org
2http://soup.informatik.uni-goettingen.de/