Towards secure user-centric networking: Service-oriented and decentralized social networks

Ingmar Baumgart, Fabian Hartmann
Institute of Telematics, Karlsruhe Institute of Technology (KIT), Germany
Email: {ingmar.baumgart, fabian.hartmann}@kit.edu

***Abstract*—Mobile devices like laptops or smartphones are getting more and more powerful, but still these devices are mainly used to access services, which are provided by centralized servers in the Internet. We argue that the full potential of such mobile devices could be unfold if these devices would provide services like instant messaging or file transfer themselves in a peer-to-peer manner. In this paper, we introduce SODESSON, a middleware which enables easy and secure access to services that get provided by devices belonging to the user himself and his friends or colleagues. This novel communication paradigm of user-centric networking leads to more efficient and secure communication, since the indirection introduced by servers is eliminated. Given that we focus on user-centric communication, we are able to exploit the trust relationships and communication pattern of a social graph to reach these goals.**

I. Introduction

Modern communication becomes more and more ubiquitous in our everyday lives. While stationary desktop computers are still used on fix locations like an office space or a living room, a lot of different communication gadgets go along with us as we move between different locations. Mobile devices like laptops, smartphones, music players, or handheld gaming consoles are capable of local ad-hoc communication or Internet access. We often even have multiple devices at our disposal at the same time, accessing different kinds of communication services. While Internet-based services are typically provided by a centralized server infrastructure, modern devices are often powerful enough to provide services themselves in a peer-to-peer manner.

A different aspect of our daily lives is the membership in Online Social Networks (OSNs), which we use to stay in contact with friends, family, and colleagues. A typical OSN as Facebook or Google+ lets us maintain these contacts in one single place, share personal data and send messages. It is often possible to provide different access rights to different contacts, depending on how familiar we feel with them. But since these OSNs are provided by centralized servers, there are several major drawbacks. This includes privacy issues, because all personal data is stored on the servers of the OSN operator. Furthermore permanent Internet connectivity is need to access these servers. Finally this approach excludes services, which can only be provided by a mobile device itself.

In this paper, we introduce a first draft of SODESSON (acronym for *Service-Oriented, Decentralized and Secure Social Networks*), a novel middleware which enables easy access to services that get provided by devices belonging to the user himself and his social environment. We assume that each user of our network possesses multiple mobile and/or stationary devices which have services running that he wants to share with users that are familiar to him. We use the term *user-centric networking* for this scenario. Since multiple devices of a single user might provide the same service (for example a chat service provided by an instant messaging application), the users should be able to focus on the specific service and the person who provides it, but users should not have to worry about manually accessing the correct device. With user-centric networking the *user addresses services provided by users*, instead of specific devices. The SODESSON middleware abstracts from these details by automatically choosing the device with the best availability. To a certain degree traditional applications like email or instant messaging are already user-centric, but depend on dedicated servers per application. In contrast our middleware is application-independent and completely decentralized.

The rest of the paper is structured as follows: In section II we give an overview about related work in the area of decentralized social networks and opportunistic networking. In section III we present a detailed motivation for our new communication paradigm. The architecture for our SODESSON middleware is explained in section IV followed by a description how we leverage the social context in section V. In section VI we propose an evaluation framework for user-centric networks followed by a discussion of open research issues in section VII.

II. Related work

Decentralized social networks in the classic OSN sense currently are in the spotlight, both by researchers and non-academic open source projects. The idea of self-governed data publication is based on an increasing public awareness of privacy protection. This has its origin in the discontent with strict terms of service of centralized OSNs like Facebook and fear of identity theft.

Diaspora [1] is a popular decentralized OSN based on independent servers (comparable to protocols like SMTP or XMPP), which has gained notable news coverage [2][3]. Some academic approaches are LifeSocial [4], Safebook [5] and PeerSoN [6], which are all based on DHTs. All these networks have a feature set in common that is similar to Facebook and offer a direct alternative for publishing profile data, photo albums and status updates.

However, SODESSON has a different direction than these approaches, as it focuses on an easy “just works” access to

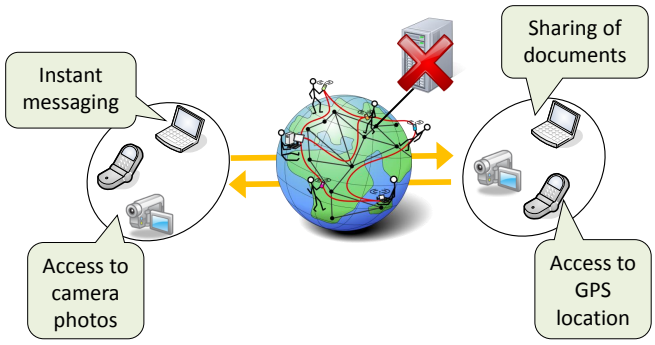

Fig. 1. SODESSON scenario: User-centric networking

services on mobile devices and does not take global connectivity for granted. The social context is leveraged mainly for efficiency and security aspects as shown in section V.

Hence, SODESSON is more comparable to designs like the Unmanaged Internet Architecture (UIA) [7] than typical OSNs. UIA focuses on the easy, user-friendly establishment of connections in heterogeneous environments between devices that have been introduced to each other before. It addresses several challenges and possible solutions for this task, but uses a device-oriented addressing scheme. Our SODESSON middleware, in contrast, provides a user-centric addressing scheme, hiding the complexity of service discovery and device selection from users and applications.

MyNet [8] is a project by Nokia which picks up the idea from UIA, adds improvements as the security concept of Passlets and evaluates it by users studies, focusing on an ease of use. Still, MyNet leaves the basic UIA architecture untouched.

Haggle [9] is a data dissemination framework based on opportunistic communication between mobile devices in a data-centric paradigm. The framework focuses on data search and retrieval in mobile ad-hoc networks and relieves applications from their infrastructural requirements. Haggle regards these networks as a resource pool for data storage and distribution and does not focus on addressing specific users. Leverage of the social context is only considered marginally.

III. Motivation

SODESSON's goal is to unfold the full potential of mobile devices by providing a service directly on a mobile device itself (see Fig. 1). Examples include the provision of Internet access via WLAN for other users (*user-provided networking* [10]), the remote access to photos of a camera, and the provision of a local weather or traffic report.

These services should, however, be made available to such users, for which a personal trust relationship exists, since providing a service leads to two challenges: Personal data may be disclosed as well as costs may occur to the device owner (e.g. traffic fees for Internet access). Social networks represent social relationships and therefore provide an excellent basis for the integration of trust relationships to secure this novel form of service provision. Integrating the social context leads to fundamental new research challenges as well as opportunities and might change the future social behavior in terms of collaboration and communication fundamentally. High-speed local communication via ad-hoc links or infrastructure networks and automated yet secure access are feasible between device that "know" each other on a social context basis. SODESSON includes contact and rights management as we know it from typical OSNs as Facebook to establish knowledge about the social context in the SODESSON-enabled devices.

Our user-centric networking approach leads to a new addressing scheme: Instead of addressing servers (by their IP addresses), we address services provided by trusted users of the social network. A major challenge of this new paradigm is the development of suitable communication protocols, e.g. to discover services in a distributed manner. Because services are provided directly between users' devices, the geographical proximity between devices can be exploited to keep network traffic local (e.g. direct communication via Bluetooth, WLAN, a local corporate network, or within an urban area). In contrast the traditional approach always involves an indirection over a centralized server. Therefore the direct approach is more robust and efficient as well as cost-effective, since the costly deployment and maintenance of such servers is eliminated. An additional benefit of the new approach is that services often can be accessed without the need for Internet connectivity. This may lead to new forms of communication particularly in areas without or with only limited Internet connectivity (reducing the "digital gap").

IV. Architecture

SODESSON's architecture follows the philosophy that user interaction is only required on a very high abstraction level. We assume that nowadays most users possess multiple devices (stationary and mobile) which can provide different types of services. For users who want to use a service, it is tedious additional work to remember which service is provided by which device or to find the best connectivity among multiple, possibly unavailable devices. Another issue with mobile devices is limited battery supply. If a service (or specific data) can be provided by multiple devices it is preferable to choose a device with plenty of spare resources (e.g. a desktop computer). These tasks can be encapsulated by a smart middleware, that relieves the users from keeping track of those devices.

For example, if Alice wants to send an instant message to Bob, she should not need to speculate which distinct device Bob is currently actively using to ensure that he reads her message as soon as possible. In this case, since we don't want to presume permanent Internet connectivity, we need a distributed approach for service discovery to find the device Bob is currently using.

For another example, if Alice wants to access Bob's latest shared photo album, she does not care which distinct device in Bob's possession actually provides the pictures. Multiple devices in Bob's possession might run the same service and store different pictures. Addressing each single device until the

correct pictures are found might be a lot of work for Alice. She should rely on SODESSON to do this automatically.

When we regard these two use cases, we can identify three major layers of abstraction on each SODESSON enabled device:

- An application layer, which contains several applications providing the services (e.g. instant messaging or file sharing). In contrast to a traditional device-centric addressing scheme, these services are addressed by a user-centric addressing scheme. Thus, the address consists of a device-independent combination of target user and target service (in the first example Alice would address *InstantMessaging@Bob*).
- A middleware which maintains the status of registered applications on the device and keeps track of what services are available. It also maintains the social graph, keeps track of familiar devices, deals with device mobility and enforces access control for provided services.
- Multiple network interfaces which can used by the middleware to communicate. The selection of a network interface depends on the requested service and the available devices (for example using a laptop for sending messages via Bluetooth to a smartphone, while exchanging large data volumes via Gigabit Ethernet).

The remainder of this section discusses the SODESSON middleware layer which sits below the applications and above the different network interfaces. As shown in Figure 2, the SODESSON middleware consists of the following components:

- **Contact manager:** This module is connected with the user interface and holds all information regarding the social context of the person the device belongs to: Here the user can manage his contact list, credentials for authentication and give basic rights to different users. Since a user typically has a very diverse set of social contacts with varying trust level, each contact can be assigned to one or more *groups*. Each group represents a different class of social contacts with a specific trust level, which can be used for easy access control to selected services. This trust relationship is asymmetric: If user A decides to add user B to one of his groups this does not implicate that user B adds A to any of his groups. Every user is identified by a random 160 bit *user id*. A user can be added in three ways: By entering his user id manually, by selecting a physically close user (user ids are broadcasted in local networks) or, if available, by using a global distributed lookup service like e.g. P2PNS [11].
- **Service manager:** This module brings together the knowledge about services that are currently available on this device, as well as the knowledge about connectivity towards other devices that are compatible with these services. The service manager's main task is to select a suitable device for the delivery of messages from the application layer which are addressed to a specific user and service. Applications use a publish/subscribe-based interface to deliver messages to the service manager.

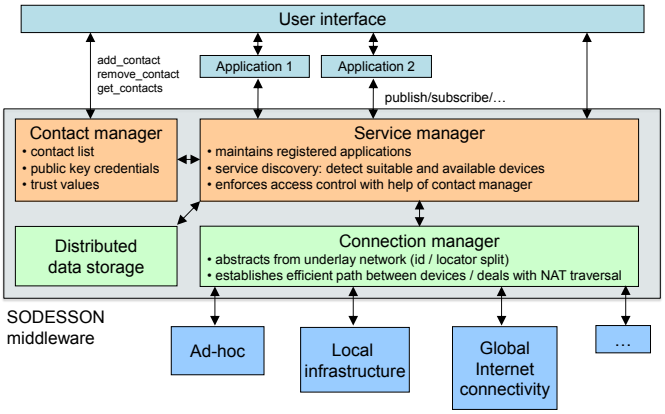

Fig. 2. SODESSON architecture

The device selection is based on metrics like fastest connection, most local connection, most frequently used device or most recently used device.

- **Connection manager:** Once the service manager has selected the best device based on the chosen metric for the given combination of user and service, it passes messages from the application layer down to the connection manager for delivery. This module abstracts from the different possible physicals connections provided by the device's interfaces, supports NAT traversal and provides an ID/locator split architecture to handle device mobility.
- **Distributed data storage:** Depending on the type of service (see Section IV-A) not the direct connection to a specific device may be needed, but instead access to a multi-device spanning data storage like a DHT. This data storage could be created Internet-wide, within a LAN or as a mixture of both: A relay node inside the LAN might accept requests by device that are only capable of local communication and forward these delegations into the Internet. Again it is up to the connection manager to choose the best interface here.

A. Types of services

Different applications have different demands regarding network and device resources, as well as availability. For example, a profile page as we know it from Facebook is a rather small set of data. Typically user wants his profile page to be available at all times, regardless of the connectivity status of his own devices. To achieve this in a distributed system the profile page could be stored in a (Internet-wide or local) DHT. On the other hand, sharing a large movie file between two users is not feasible in an Internet-wide DHT with high churn. In this case access to the movie file can only be provided by direct communication as long as a connection to the device is available.

By taking these use cases into account, we identified three types of services:

- **Direct**: Interactive communication between online devices (e.g. file transfer)

- **Persistent**: Distributed storage (e.g. access of a user profile if user is offline)
- **Hybrid**: Combination of direct and persistent (e.g. delay-tolerant delivery of IM message if user is offline)

The hybrid type is the most complex one, because the service manager has to decide in which cases an application message needs to be persisted. This is not only dependent on the pure connectivity of the devices, but also on the user presence status. For example, if a user participates in an IM session on his desktop PC and suddenly leaves for lunch, but leaves his PC running, it is still reachable in the network. However, the user might want to get subsequent messages on the smartphone he is carrying with him on his way to lunch.

V. Social Context Leverage

SODESSON focuses on social relationships with a close locality. Regarding this kind of context between the users brings advantages in two aspects: efficiency and security.

- **Efficiency**: While it is still possible to give access to geographically distant friends or relatives via the Internet, the framework shows its full potential in easy access via spontaneous wireless connections or high-speed local area networks. Even when using Internet access, connecting two devices in the same city is more efficient than detouring via a server on the other side of the world.
  Adding new devices to the personal pool is also easy since no dedicated share of the device is needed. The same concept of basic and service-specific access rights applies and can be propagated from existing devices in the pool. On the other hand, the new device can silently announce its participation for existing services, no human interaction is needed.
  Service discovery in general gets much easier and less costly. While it remains a big challenge in decentralized networks, a lot of complexity can be avoided by focusing on devices inside the social context only.
- **Security** Security is another aspect which profits from social relationships and locality. For each trusted contact a public key is stored. Personal meetings make it easy to sign each others public keys and thus create a web of trust which is congruent to the personal social network. The SODESSON middleware uses these keys for unified authentication, encryption and access control independent from the applications.
  Stored data (permanently available data from persistent services as well as ephemeral data from hybrid services) can be stored on devices of familiar and trusted users only, making autonomous data networks with fully connected trust feasible, comparable to Darknets.

VI. Simulation Environment

Since distributed systems tend to be very complex and are hard to analyze, we started to work on a suitable simulation environment, which allows us to evaluate user-centric networks. There are several peer-to-peer simulators like *PeerSim* [12] or *PlanetSim* [13], but they lack suitable models for e.g. social user behavior, mobile device classes or user mobility. Another

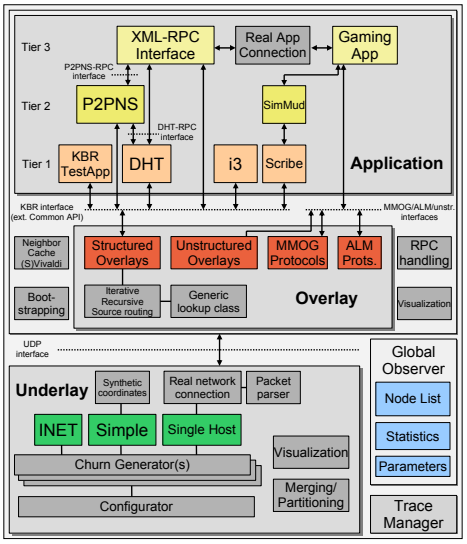

Fig. 3. OverSim's modular architecture

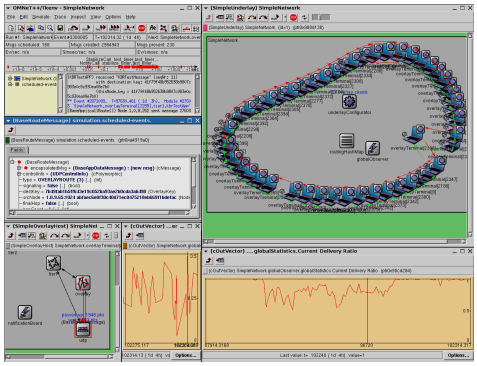

Fig. 4. Screenshot of OverSim's user interface

class of simulators is used for the evaluation of delay-tolerant networks. These simulators (like e.g. ONE [14]) provide very detailed movement models but lack support for infrastructure networks.

Therefore we started to extend our popular simulation framework *OverSim* [15] to a framework for the evaluation of user-centric networks.

A. OverSim

OverSim [15] is a flexible overlay simulation framework based on OMNeT++. OverSim comprehensively includes many structured and unstructured peer-to-peer protocols as well as several event distribution protocols.

All protocol implementations can be used for both simulation as well as real world networks. Additionally, OverSim provides several common functions for structured peer-to-peer networks to facilitate the implementation of additional protocols and to make them more comparable. OverSim utilizes the GUI that comes with of OMNeT++ to display overlay and underlay topologies and network packets in detail (see Fig. 4), thus allows for intuitive debugging.

OverSim's architecture shown in Figure 3 allows the mod-

ularized modeling of all components in a P2P network in easily exchangeable or extensible manner, thus facilitating code reuse. Several exchangeable underlay network models allow to simulate complex heterogeneous underlay networks as well as simplified networks for large-scale simulations (up to 100 000 nodes have been simulated successfully).

1) Underlay abstraction: The framework provides different underlay abstraction models differing in complexity and accuracy, being the *SimpleUnderlay*, support of the *INET Framework* as well as the *SingleHostUnderlay*.

The SimpleUnderlay is most suitable for the evaluation of user-centric networks and thus we give a short introduction on this model. It combines a low computational overhead with high accuracy, making it a good model for simulating large overlay networks. Nodes are placed into a n-dimensional Euclidean space, determining mutual delays based on their euclidean distance. Nodes' positions are chosen to match the measurements from the *CAIDA/Skitter* project. Additionally, each node is assigned to a logical access network characterized by bandwidth, access delay, jitter and packet loss parameters to allow the simulation of heterogeneous access networks. Mobility can be achieved by changing coordinates, access network characteristics and the IP address of a node. To model bandwidth effects, each node contains a logical sending queue. The SimpleUnderlay allows for simulation of underlay network partitioning and merging.

2) Churn modeling: OverSim provides several models for generating churn, including a lifetime-based churn model supporting different distribution functions (e. g. Weibull, Pareto or Exponential). It is possible to use more than one churn generator at the same time to simulate groups of nodes with different churn behaviors. For each churn generator, different node configurations and overlay parameters can be specified, allowing easy generation of heterogeneous devices, which is particularly needed for our user-centric networking scenario.

B. User-centric networking extensions

To enable the simulation of user-centric networks, OverSim needs to be extended by several new components.

First of all the representation of a *social graph* is needed. For each user its neighbors are individually classified in different groups (e.g. family member, friend, colleague). This is important, since the class of a contact has influence on the mobility, trust and application models.

Each user of this social graphs owns several *devices*. We classify these devices in three groups:

- **Fixed devices**: These are devices which are stationary, like desktop computers or a WLAN access point.
- **Mobile powerful devices**: These are mobile devices which are carried around by its owners and are rather powerful like a laptop computer (which e.g. either has a large battery or is often attached to external power sources).
- **Mobile limited devices**: These are mobile devices like smartphones or PDAs, which are short on resources like battery or computing power.

Since there are complex dependencies between social context, device type, mobility model and application traffic it is difficult to build a coherent model, which covers all of these aspects. A frequent approach to generate behavior models is to use traces from social experiments (like mobility models used in DTN networking). Unfortunately traces containing annotations for all relevant aspects (social graph with user groups, device types, mobility and application traffic) are not yet available.

Due to the lack of such traces we argue, that such models need to be generated based on the motivation why users move or communicate. For the evaluation of user-centric networks, we propose the following *SODESSON user model*.

In this model each user and each device device has a current *location*. Since people spend most of their time in very few places [16], we think the following four locations are sufficient:

- **Home**: Typically users stay in their homes at night and part of the weekend. When they are at home they mainly communication with their family members or friends. Often there is a fixed WLAN access point available.
- **Work**: During working hours many users stay in a single office location and mainly communicate with colleagues. Often they have a dedicated stationary desktop computer in their office.
- **Leisure**: In their spare time users often move to leisure locations (e.g. a bar or a cinema) to meet and to communicate with friends using mobile devices.
- **Travel**: The travel locations are symbolic locations and represent users, which are currently traveling. These users don't have access to any (personal) fixed devices and often are physically far away from their everyday locations. As a consequence communication with their friends or family members at home needs Internet connectivity and often comprises larger network latencies.

Our underlay network model is mainly based on OverSim's scalable *SimpleUnderlay* model, which already provides typical Internet latencies. For our scenario we need to randomly map every logical user location to a Skitter/CAIDA synthetic coordinate at the beginning of a simulation considering several constraints: We assume the *home*, *work* and *leisure* locations of a user are geographically close and therefor the euclidean distance in the synthetic coordinate needs to be close too. The second constraint is the sharing of locations: If a colleague (or family member) of a user *X* already has synthetic coordinates for its work (or respectively home) location assigned, this coordinates get reused for the locations of user *X*.

An important aspect, which needs to be modeled for user-centric networks, are connectivity domains (e.g. local ad hoc networks, local corporate network, global Internet connectivity) and limited connectivity due to NATs/firewalls. In a simple first step, we only differentiate between local ad hoc connectivity for all devices currently at the same location and global Internet connectivity for some devices at selected locations.

VII. Current research focus

The SODESSON architecture presented in this paper is still work in progress. Our current research focus is on the leverage of the social context for *overlay routing* and *distributed storage*.

Overlay routing provides the ID/locator split architecture and is needed to establish a communication path between mobile devices. As a first step we compared the performance of several well-known structured overlay protocols under typical churn with OverSim. The simulations results have shown, that Bamboo[17] and Kademlia [18] have the best performance vs. cost tradeoff and are good candidates, if there is global Internet connectivity.

But clearly a global structured overlay based on Bamboo or Kademlia isn't optimal for our scenario. What if devices frequently have to switch between Internet wide and local communication? And how can we use the trust from our social graph to improve routing, if some nodes are malicious?

The same questions apply to the distributed storage protocol. Classical replication strategies for distributed hash tables don't seem to be suitable in our scenario. Instead we need to exploit the information from our social graph and our mobility behavior to select suitable devices for replication. To increase the robustness of the system against attackers it seems promising to store data on trusted devices of our social contacts. But does this really improve availability? This again depends on the mobility behavior of our users.

VIII. Conclusion

SODESSON was motivated by the fact, that although mobile devices get more and more powerful, they are mainly used as dumb terminals to access services which are provided by central server in the Internet. This is even more absurd if we consider, that many applications deal with direct communication between users, like instant messaging, voice communication or file transfer.

In this position paper we argued, that the direct provisioning of services by the mobile devices themselves in a peer-to-peer approach would be more robust, more efficient and safer than the indirection introduced by using central servers in the Internet. Therefore we introduced the novel concept of *service oriented and decentralized social networks* as an facility towards user-centric networking. We presented the architecture of our SODESSON middleware which relieves the burden of service discovery, access control and mobility management from the applications.

A challenging task is the evaluation of such user-centric networks, since there are complex dependencies between social context, user mobility, application traffic and the type of involved devices. Therefor we proposed several extensions to our OverSim simulation framework with the goal to provide a realistic but scalable models to evaluate such networks.

Currently many research questions like suitable overlay routing protocols and replication strategies are still open. Nevertheless we think that user-centric networking is a very promising communication paradigm and we should start to discuss the challenges and opportunities as early as possible.

Acknowledgment

This research is supported by the Concept for the Future of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative.

References

[1] "Diaspora website," https://joindiaspora.com/, Jul. 2011.

[2] J. Dwyer, "Four Nerds and a Cry to Arms Against Facebook," *New York Times*, May 17 2010. [Online]. Available: http://www.nytimes.com/2010/05/12/nyregion/12about.html

[3] M. Shiels, "The anti-Facebook," *BBC News*, May 12 2010. [Online]. Available: http://www.bbc.co.uk/blogs/thereporters/maggieshiels/2010/05/the_antifacebook.html

[4] K. Graffi, S. Podrajanski, P. Mukherjee, A. Kovacevic, and R. Steinmetz, "A distributed platform for multimedia communities," in *IEEE International Symposium on Multimedia (ISM '08)*, IEEE. Berkley, USA: IEEE Computer Society Press, Dec 2008, p. 6.

[5] L. Cutillo, R. Molva, and T. Strufe, "Safebook: A privacy-preserving online social network leveraging on real-life trust," *Communications Magazine, IEEE*, vol. 47, no. 12, pp. 94 –101, dec. 2009.

[6] S. Buchegger, D. Schiöberg, L.-H. Vu, and A. Datta, "Peerson: P2p social networking: early experiences and insights," in *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*, ser. SNS '09. New York, NY, USA: ACM, 2009, pp. 46–52. [Online]. Available: http://doi.acm.org/10.1145/1578002.1578010

[7] B. Ford, "UIA: A Global Connectivity Architecture for Mobile Personal Devices," Ph.D. dissertation, Massachusetts Institute of Technology, Sep 2008.

[8] D. N. Kalofonos, Z. Antoniou, F. D. Reynolds, M. Van-Kleek, J. Strauss, and P. Wisner, "Mynet: A platform for secure p2p personal and social networking services," *Pervasive Computing and Communications, IEEE International Conference on*, vol. 0, pp. 135–146, 2008.

[9] E. Nordström, P. Gunningberg, and C. Rohner, "A search-based network architecture for mobile devices," Department of Information Technology, Uppsala University, Tech. Rep. 2009-003, Jan. 2009.

[10] R. Sofia and P. Mendes, "User-provided networks: consumer as provider," *Communications Magazine, IEEE*, vol. 46, no. 12, pp. 86 –91, december 2008.

[11] I. Baumgart, "P2PNS: A Secure Distributed Name Service for P2PSIP," in *Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2008), Hong Kong, China*, Mar. 2008.

[12] M. Jelasity, A. Montresor, G. P. Jesi, and S. Voulgaris, "The Peersim simulator," http://peersim.sf.net/, Dec. 2008.

[13] P. Garca, C. Pairot, R. Mondjar, J. Pujol, H. Tejedor, and R. Rallo, "Planetsim: A new overlay network simulation framework," in *Software Engineering and Middleware*, vol. Volume 3437/2005, 2005, pp. 123–136.

[14] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for dtn protocol evaluation," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, ser. Simutools '09. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, pp. 55:1–55:10. [Online]. Available: http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5674

[15] I. Baumgart, B. Heep, and S. Krause, "OverSim: A Flexible Overlay Network Simulation Framework," in *Proceedings of 10th IEEE Global Internet Symposium (GI '07) in conjunction with IEEE INFOCOM 2007*, Anchorage, AK, USA, May 6–12, 2007, pp. 79–84.

[16] M. C. Gonzlez, C. A. Hidalgo, and A.-L. Barabsi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008. [Online]. Available: http://dx.doi.org/10.1038/nature06958

[17] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, "Handling Churn in a DHT," in *ATEC '04: Proceedings of the annual conference on USENIX Annual Technical Conference*, Boston, MA, USA, Jun./Jul. 27–2, 2004, pp. 127–140.

[18] P. Maymounkov and D. Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in *Peer-to-Peer Systems: First International Workshop (IPTPS 2002). Revised Papers*, vol. 2429/2002, Cambridge, MA, USA, Mar. 7–8, 2002, pp. 53–65.