MobiClique: Middleware for Mobile Social Networking

Anna-Kaisa Pietiläinen
Thomson
annakaisa.pietilainen@thomson.net

Earl Oliver
University of Waterloo
eaoliver@uwaterloo.ca

Jason LeBrun
University of California, Davis
jblebrun@ece.ucdavis.edu

George Varghese
Univ. of California, San Diego
varghese@cs.ucsd.edu

Christophe Diot
Thomson
christophe.diot@thomson.net

ABSTRACT

We consider a mobile ad hoc network setting where Bluetooth enabled mobile devices communicate directly with other devices as they meet opportunistically. We design and implement a novel mobile social networking middleware named *MobiClique*. MobiClique forms and exploits ad hoc social networks to disseminate content using a store-carry-forward technique. Our approach distinguishes itself from other mobile social software by removing the need for a central server to conduct exchanges, by leveraging existing social networks to bootstrap the system, and by taking advantage of the social network overlay to disseminate content. We also propose an open API to encourage third-party application development. We discuss the system architecture and three example applications. We show experimentally that MobiClique successfully builds and maintains an ad hoc social network leveraging contact opportunities between friends and people sharing interest(s) for content exchanges. Our experience also provides insight into some of the key challenges and short-comings that researchers face when designing and deploying similar systems.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*store and forward networks*

General Terms

Design, Experimentation

Keywords

Opportunistic communications, Mobile social networks, Bluetooth

1. INTRODUCTION

Applications in the virtual world such as online social networks and instant messaging have done much to remove the tyranny of geography. Beyond friendship and exchanges between two parties (which we refer to as *dyadic* communication), virtual groups have proliferated creating communities centered around interests varying from gaming to editing Wikipedia pages. Despite the increased power and reach of virtual communities, we postulate that the power of physical communities based on physical contact and closeness will continue to be an essential part of human relationships. Further, physical communities have a different set of capabilities that are complemented by, but not subsumed by, virtual communities. Rather than viewing virtual communities and physical communities as *competing* entities, we think of them as *complementary*. We design *MobiClique* as a way to leverage the virtual and physical worlds so that users can move between them in a way that enhances both.

An interesting answer to the problem above was initially provided by the MIT Serendipity project [4]. Serendipity was inspired by the growing and ubiquitous use of personal mobile devices. It proposed a simple situated introduction system that relied on Bluetooth device discovery for locating nearby users and a central server for matching the user profiles. Today, most of the advanced mobile terminals come equipped with multiple wireless radio interfaces including Bluetooth, 802.11, and cellular radio, which allow mobile devices not only to discover but also to communicate with other devices in their neighborhood. However, these communication opportunities remain largely unexplored.

In this paper we build on the ideas of Serendipity and other similar systems [12, 10] to design and implement MobiClique, a mobile social software that allows people to maintain and extend their online social networks through opportunistic encounters in real-life. In contrast to previous solutions, MobiClique does not depend on a central server or infrastructure connectivity; it relies exclusively on opportunistic connections between neighboring devices. MobiClique leverages existing online social networking services (OSNs) that maintain user connections in a *virtual world* to enable decentralized ad hoc interactions. We bootstrap MobiClique using the existing virtual world user profiles and enable temporary connections based on proximity and social compatibility in the *physical world*. When mobile users meet opportunistically, and if the two user profiles share some pre-defined relationship such as friendship or interest, the users are alerted and can choose to have an *exchange*. An exchange can be anything from a simple friendship introduction to content dissemination. Later, these ad hoc interactions can be transformed, if the user wishes to, into

WOSN'09, August 17, 2009, Barcelona, Spain.

"friend" links and other objects (e.g. status updates) in the virtual world. Moreover, MobiClique goes beyond dyadic device to device exchanges as each contact can increase the size of the social graph on a device upon discovery of previously unknown devices and communities. This offers an interesting framework for collaborative forwarding using opportunistic contacts as proposed in [15]. Finally, we propose a mobile social ad hoc networking API that allows developers to build new and unforeseen applications on top of the MobiClique middleware and the social networking overlay it provides.

In order to test this new type of ad hoc social applications, it is important to rely on real-life experiments: first, because environmental conditions are generally complex and difficult to model or simulate, and second, because the behavior of users, which is a key feature of the application, has to be observed in situ. However, running such experiments poses unique challenges: finding a favorable social environment, providing sufficient applications to keep people interested in the experiment, while keeping them as simple as possible, avoiding time wasting boot-strapping by leveraging existing social networks and dealing with the limitations of available mobile devices including limited power and communication efficiency. We show how we have encountered and addressed these challenges during two experiments held in two major networking conferences. In our trials, 28 and 22 participants respectively, use MobiClique to successfully build and maintain ad hoc social networks, and to exchange messages between each other.

The paper is organized as follows. We present MobiClique in Section 2. We describe the initial prototype implementation and three mobile social networking applications we implement using MobiClique. In Section 3 we discuss two real-life user trials we conduct in networking conferences with MobiClique enabled smartphones. We survey related work in Section 4, and conclude in Section 5.

2. MOBICLIQUE DESIGN

2.1 Social Network Data Portability

Online social networks, such as MySpace, Facebook, and LinkedIn, are designed to create and maintain users' social profiles and the virtual social network formed between users. A typical *social profile* consists of a unique user identifier, varying pieces of personal information, a list of friends (i.e. links to other users of the service) and a list of social groups (or networks) consisting of users sharing some common interest. In the initial design of MobiClique a social profile consists of a unique user identifier, a user name, a short description (can be used as a status message), a list of friends and a list of interest groups. We choose this simplified format mainly for fast prototyping. However, it can be easily replaced by an existing OSN profile format or a future standard. There already exist several open initiatives to standardize the user and social profile representations and access including FoaF[1], SocialGraph API[2], and OpenSocial[3].

We use the Facebook API[4] to bootstrap the MobiClique social profile. The Facebook API provides access to its underlying social graph data for web and stand-alone desktop applications. The API methods include functions to retrieve the full user profile, and the list of friends, networks, groups, and events that the user is attending.

Currently MobiClique only retrives information from Facebook. The methods to post information back to Facebook are limited. One can create new events, upload pictures, and change the user's status through the API; however, addition of new friends or groups for example is not yet supported. We believe that in the future, the OSN APIs will open up and allow third-party applications to update any kind information. In MobiClique the potential updates include status and location updates, new contacts and contact statistics, and other content exchanged over opportunistic communications.

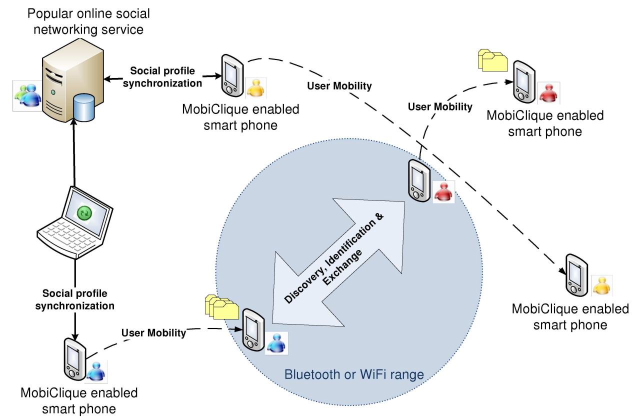

Figure 1: MobiClique system overview.

2.2 System Architecture

The MobiClique system overview is illustrated in Figure 1. The social profile of each MobiClique node is initialized from and kept synchronized with an existing OSN service (illustrated on the left side of the figure). While the MobiClique system mainly operates in ad hoc mode as the devices come within the range of each other, we assume that the devices occasionally have access to the Internet (possibly through a laptop with an Internet connection) to synchronize the social information between the device and an OSN service.

In ad hoc mode, each MobiClique node executes a periodic loop that consists of three steps: (1) neighborhood discovery, (2) user identification (and authentication), and (3) data exchange. The neighborhood discovery method depends on the radio technology being used: commonly available options with today's mobile device hardware include Bluetooth device discovery or broadcast beacons on a well-known WiFi SSID. We illustrate the ad hoc mode by the circle in the center of Figure 1 where two devices move in the vicinity of each other and engage in interaction. Of course the neighborhood can, and usually will, contain more than two devices; the system must therefore manage multiple simultaneous connections.

Upon discovery of a new device in the neighborhood, MobiClique enters the identification phase where the devices open a communication link between each other to exchange the user identity information. Upon a first encounter the

[1]http://www.foaf-project.org/
[2]http://code.google.com/apis/socialgraph/
[3]http://www.opensocial.org/
[4]http://developer.facebook.com

```
[de]register_application(name)
register_event_interest(event)
[get|set]_social_profile(profile)
send_message(destination, ttl-timestamp,
             ttl-hops, message)
```

Figure 2: MobiClique API.

devices exchange their full social profiles during the identification phase. MobiClique stores the profiles persistently along with other contact statistics to avoid unnecessary profile updates and to make subsequent decisions, e.g., to forward messages between nodes (see below for details). During subsequent contacts the full profiles are exchanged only if the profile has changed since the last encounter (i.e., user changed his nickname or status, or added new friends and so forth). Otherwise the nodes only exchange their user identifiers. In the initial design we do not include any security mechanisms. However, the user identification and authentication could be based on public keys as proposed in the Host Identity Protocol (HIP) architecture [8], for example.

Once the identification is successfully completed, the last step of the interaction is the data exchange phase, whereby devices exchange application level messages between each other. These messages are stored persistently on the devices within the limits of storage space and forwarded to other devices as contact opportunities arise. These opportunistic exchanges combined with human mobility create a temporal communications network as in Pocket Switched Network (PSN) [15] where messages travel from device to device over multiple hops without any infrastructure connectivity. In MobiClique messages can be addressed directly to other MobiClique users or groups of users. More specifically, applications can send messages to the friends of the user or to interest groups the user has joined. The forwarding is performed using two simple rules: (1) unicast messages, i.e., messages for a specific user, are sent either upon meeting the destination directly or forwarded through friends of the destination, and (2) group messages are flooded within the corresponding interest group so that each member of the group will participate to the forwarding until everybody has received the data. These rules ensure that no node receives content unless it belongs to the target group of interest or is a friend of the destination. This provides already a minimum guarantee on privacy and it helps also as an incentive mechanism. However, addressing these two issues more deeply is out of the scope of our first prototype design.

Messages are removed from the system after an application defined time-to-live (TTL) has expired. MobiClique TTL is a combination of an absolute timestamp and a maximum number of hops. Messages are deleted either when the TTL has passed or the message has taken a maximum number of hops in the network.

2.3 Mobile Social Networking API

The MobiClique API is detailed in Figure 2. Applications wanting to use the MobiClique communication service must first register with the middleware through the API. The API provides methods to manage the local node social profile to change personal information and to add and remove friends and interests. The system informs applications on various events using asynchronous notifications. Possible events include contact events (contact start and contact end) and incoming message notifications. Each contact event contains the encountered node's social profile to enable application level functions based on contact type and contact statistics. Finally, the API supports opportunistic messaging between devices through a destination-oriented messaging abstraction. A message is an autonomous piece of data consisting of a (optional) file attachment and textual data formatted as a list of name-value pairs. The messaging method takes also as parameter the destination (either a friend or a interest group id) and the TTL as defined in the previous section.

2.4 Prototype Implementation

We implement MobiClique using both C++ and C# to run on Windows Mobile platform. Our implementation builds on the code and functionality of the reference implementation of the Haggle[5] network architecture [11], and on our experiences on opportunistic communications [13]. Haggle is an architecture for opportunistic, data-oriented networking. The core system is event-driven and consists of a single event queue and a set of managers that create and act on various system events such as new neighbor discovery or incoming data from the local applications or the network. We extend the Haggle reference implementation with the node social profile management and social graph maintenance. We also implement our own social graph based forwarding algorithm.

For the initial experiments we enable only the Bluetooth connectivity due to its energy efficiency. In laboratory benchmarks, we observe that constantly powering the WiFi interface drains the battery of our experimental devices in couple of hours while Bluetooth provides approximately 8 to 10 hours of battery life [13]. The Bluetooth device discovery is performed every 2 minutes (+/- small random delay to avoid synchronized discoveries) for the duration of 10.24 seconds, which is the recommended minimum duration by the Bluetooth standard [1]. For data communications, we use device-to-device RFCOMM links and only three simultaneous connections are allowed at a time. This is less than the maximum allowed number of devices in a Bluetooth piconet (i.e., 7). We chose this limit to reduce interference.

The prototype profile initialization is implemented using a simple Facebook desktop application. The application connects to Facebook, prompts for user login, retrieves user profile information (user identifier, user name, list of groups and networks and list of friends) and stores this data in a text file. This file is copied to the user device (using Microsoft's ActiveSync synchronization software). Upon the first launch of the MobiClique application on the device, the system reads the file and the user is presented with a profile setup dialog that allows her to edit the personal information and select the friends and interest groups to be imported to the ad hoc social profile.

2.5 Applications

The prototype includes the following three user applications that we implement using the proposed API:

Mobile Social Networking. The application displays to the user the current set of neighboring devices with their social profiles and provides an interface to add or remove friends and change interests. The user can enable distinct alerts to be notified when friends, friends of friends, or mem-

[5]www.haggleproject.net

	CoNEXT'07	CoNEXT'08
Duration	3,5 days	3,5 days
Participants	28	22
Active time	56.91h	56.70h
Inactive time	30.72h (49.80%)	22.19h (30.07%)
Bluetooth contacts	15918	11352

Table 1: Characteristics of the collected data sets.

bers of the interest groups are in the neighborhood (i.e. within the range of Bluetooth).

Asynchronous Messaging. This application can be used to send messages between a pair of users over the social network substrate. It is an ad hoc analogue to existing Internet based instant messaging applications. A user can at anytime send a message that consists of a textual message body and an optional file attachment to a friend in her MobiClique friend list.

Epidemic Newsgroups. This application is an opportunistic equivalent of the traditional Usenet messaging service. It enables discussions among multiple disconnected participants sharing some specific interest. As with asynchronous messaging, a message consists of text and an optional file.

3. USER TRIALS

3.1 Experimental Setup and Limitations

We conduct two field trials with 28 and 22 participants respectively using MobiClique enabled smartphones in two major networking conferences. The objective of our experiments is to validate our design choices; and to collect information on contact opportunities and real user experience about mobile social applications. We conduct our experiments in conference settings primarily due to the convenience of distributing and retrieving the mobile devices. The mobility patterns of conference members are also geographically constrained, which creates opportunities for interaction. Moreover, conference attendees typically have pre-existing social relationships, which we expect MobiClique to be able to exploit and enrich.

The main experimental device used in our experiments is HTC s620 Windows Mobile Smartphone[6]. The device has a 200MHz TI processor, 64MB of RAM, 128MB of ROM and a MicroSD slot (we equip the devices with a 1 GB MicroSD card). The radio interfaces include a quad-band GSM/EDGE cellular radio, Bluetooth v1.2 and 802.11b/g.

Table 1 summarizes the experiments characteristics and collected traces. The *active time* is defined as an average time between the first and the last action recorded on each participating device. The *inactive time* is the time that the device was not collecting data (i.e., either the software was not running or the device was powered off). We note that the percentage of inactivity is high in both experiments, primarily due to battery depletion. While this is part of the normal use of mobile devices, the experimental software also adds to the energy consumption due to frequent Bluetooth operations and SD card I/O; devices typically run for approximately 8 to 10 hours. As most of the participants do not put their SIM cards on the device, the motivation to keep the device charged and running is less than with their personal devices. We observe a small decrease in the inactivity between the experiments, which we mainly attribute to a more stable implementation of MobiClique in the second experiment.

[6]www.htc.com/product/03-product_s620.htm

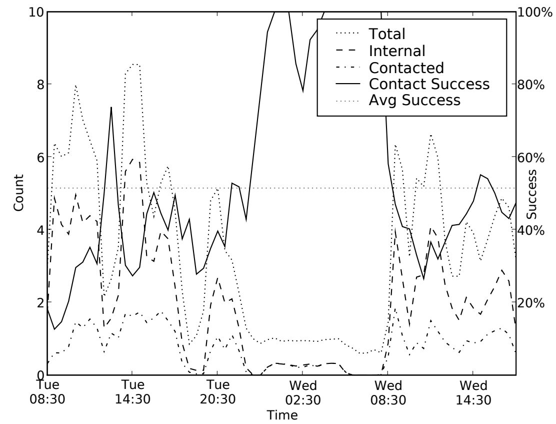

Figure 3: Bluetooth device discovery and connection success rate per scan (CoNEXT'07).

The second major limitation with the current system is the performance of Bluetooth in a dense ad hoc network. Bluetooth operations such as device inquiry, service discovery and RFCOMM connection setup are vulnerable to interference from other devices competing of the same radio resource. Moreover, when performing a device discovery, a device can not answer device inquiries from other devices. In addition, user mobility impacts the device discovery and connection success rates. These create challenging conditions for applications relying only on opportunistic Bluetooth contacts as we miss many contact opportunities. We illustrate this in Figure 3 where we plot the average number of devices discovered per each neighborhood discovery and the Bluetooth RFCOMM connection success rate as MobiClique tries to connect to each discovered device in order to exchange social profiles and data. The average Bluetooth connection success rate is only 50% meaning we manage to identify only half of the neighboring MobiClique devices. At the CoNEXT'08 experiment we miss even more contacts as we use Bluetooth service discovery to identify the MobiClique devices prior the RFCOMM connection establishment. The service discovery manages to correctly discover MobiClique nodes only 56% of the time, after which the actual connection setup and social profile exchange succeeds similarly only around 50% of the time. Despite these limitations, we still manage to obtain interesting results in terms of social network evolution and opportunistic communications that are discussed next.

3.2 Evolution of an Ad-Hoc Social Network

In this initial analysis, we aim to illustrate the effectiveness of MobiClique in building and maintaining an ad hoc social network and communities and assess the potential of social network based PSN. We plot the initial and final friendship graphs from the experiments in Figure 4. The initial graph is defined during the social profile bootstrapping. In both experiments the initial friendship graph has

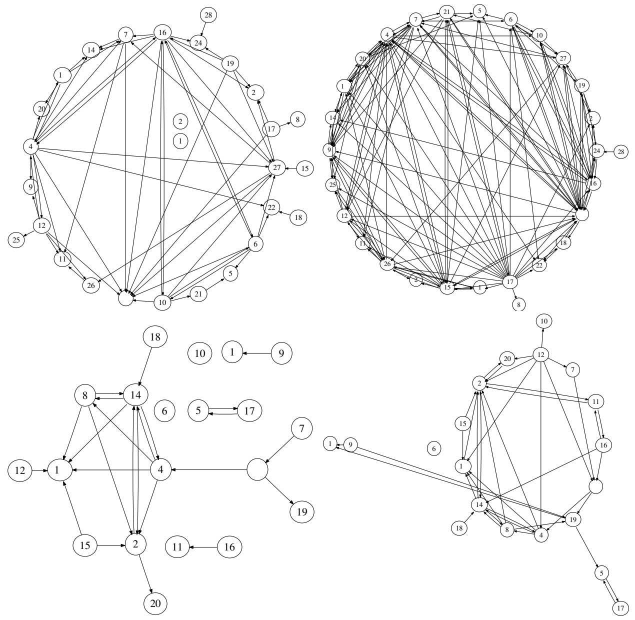

Figure 4: The initial (left) and final (right) friendship graphs of CoNEXT'07 (top) and CoNEXT'08 (bottom) experiments.

a mixture of connected and disconnected nodes. The average degree in the beginning is 5.11 at CoNEXT'07 and 1.90 at CoNEXT'08. During the conference MobiClique creates opportunities to meet new friends based on shared interests and common friends. The resulting final friendship graph is connected (except for a single node at CoNEXT'08) and the average friend degree is 9.18 at CoNEXT'07 and 2.90 at CoNEXT'08. The CoNEXT'08 experiment is less successful in building the network due to performance limitations of Bluetooth. Similarly, during the experiments users can discover and join new interest groups. At CoNEXT'07 the average interest degree evolves from 4.43 to 6.32 (out of 31 possible) and at CoNEXT'08 from 6.90 to 10.55 (unlimited number of interests). The experimental results show that MobiClique can contribute to the development of social networks even in a context of a well-defined community.

Next, we examine the interaction opportunities created by MobiClique. Figure 5 shows the evolution of the average number of contact opportunities over the experiment. We plot separately the contacts between any participating node, between nodes sharing an interest and between friends. We define a *contact* as the time between the first device discovery until the device is no more in the neighborhood during two consecutive scans. We use this definition to overcome some of the problems of Bluetooth discovery. We analyse the node identifiers and social links offline based on the known MAC addresses of the participating devices. The internal contacts evolution follows a similar trend in both experiments. The slow initial contact rate of CoNEXT'07 is due to a small software problem that was fixed during the day. The final average number of contacts per device is 568.50 at CoNEXT'07 and 540.57 at CoNEXT'08. At CoNEXT'07 49% of all the contacts occur between people sharing some

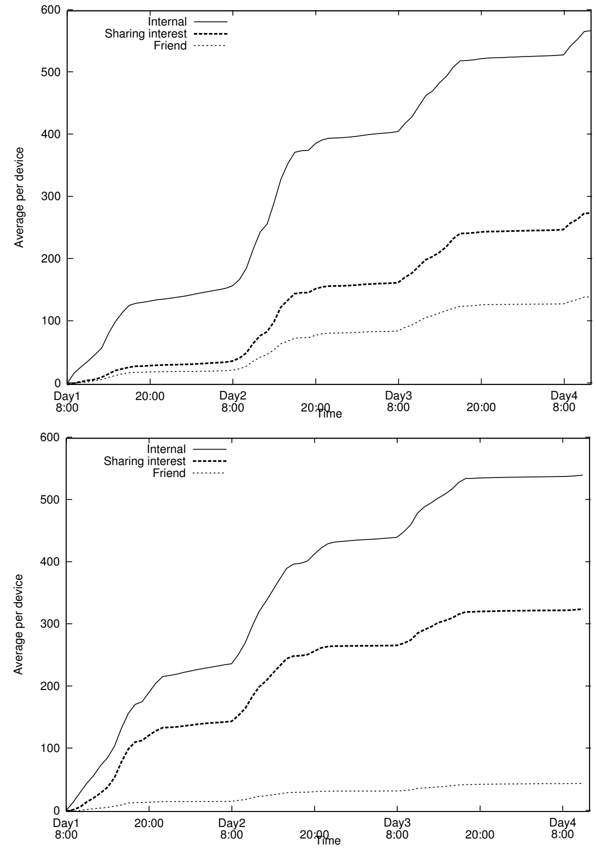

Figure 5: The sum of contact opportunities on average per participant at CoNEXT'07 (top) and CoNEXT'08 (bottom).

interest and 25% between friends. At CoNEXT'08 the respective rates are 60% and only 8% for friends.

At CoNEXT'07 we record 306 user generated messages and at CoNEXT'08 64. The lower user activity at CoNEXT'08 is again explained by the low efficiency of the selected radio interface. However, we conclude that in both experiments MobiClique is successful in building and maintaining an ad hoc social network overlay that is exploited by the participants to exchange messages between each other and within interest groups over multiple hops. We leave the detailed analysis of the messaging performance as future work.

4. RELATED WORK

Social networking has emerged as a hot research topic. MIT's Serendipity project [4] is one of the first projects that explored the mobile aspects of social networking. There have been several socially motivated technologies since Serendipity that are based on Bluetooth proximity detection [12, 10, 7]. The basic idea of Bluetooth based contact discovery is extended further in [6], where the proximity data is stored on a central server and can be later visualized through a Facebook application. MobiClique goes beyond simple profile matching as it uses proximity to exchange messages, hence potentially creating new types of communities and applications.

Most of the work in mobile social communications has been commercial and centered around sending location and status updates from mobile devices towards centralized (and proprietary) activity aggregation services (and then possibly again back to the mobile devices as notifications). Examples include Dodgeball[7], Jaiku[8], Twitter[9], and aka-aki[10]. In contrast to these, MobiClique functions mainly in ad hoc mode and leverages existing OSNs.

MobiClique also differs from all of the above cited systems in that it provides a generic framework to implement various mobile social applications. Other social network middleware architectures have been proposed including MobiSoc [2] for mobile devices and Roadspeak [14] for vehicular networks but they rely on constant availability of centralized servers.

Social networking in the context of opportunistic communications has been studied mainly to build efficient forwarding algorithms. Several recent studies [9, 5, 3] propose to use various properties of the social graph such as node centrality and community structures to make efficient forwarding decisions. MobiClique is an ideal environment to deploy and test the performance of such opportunistic forwarding algorithms which leverage the social profiles and networks of the users, as it monitor at the same time the mobility and the social behavior of users. Furthermore, the traces collected by MobiClique can be used for simulating opportunistic networks as in [9].

5. CONCLUSIONS

MobiClique is a mobile social networking middleware that leverages existing social networks and opportunistic contacts between smartphones to form ad hoc communities for social networking and social graph based opportunistic communications. We have designed and implemented MobiClique on Windows Mobile smartphones and validated the system through two user trials in two networking conferences. Despite some experimental hazards, we have shown that MobiClique is able to create mobile social networks and contact opportunities and that Pocket Switched Networks built on a such overlay can provide an interesting novel networking service. We hope that MobiClique will foster the vision of going beyond ad hoc local dyadic interaction to build new type of communities based on physical proximity, and that these communities will provide the basis for a new generation of mobile social networking applications. MobiClique and the experimental data will be made publicly available for downloading to encourage the adoption of this new social communication approach and to allow others to experiment with mobile ad hoc social networking.

6. ACKNOWLEDGEMENTS

We would like to thank all the experimentalists for participation. We would also like to gratefully acknowledge Augustin Chaintreau, Stratis Ioannidis, Abderahmen Mtibaa, Martin May and Jon Crowcroft for their useful remarks during the project and comments on earlier drafts of the paper.

[7]http://www.dodgeball.com
[8]http://jaiku.com
[9]http://twitter.com
[10]http://www.aka-aki.com

7. REFERENCES

[1] Bluetooth Special Interest Group (SIG). *Specification of the Bluetooth System. Core Package version: 1.2*, November 2003.

[2] C. Borcea, A. Gupta, A. Kalra, Q. Jones, and L. Iftode. The mobisoc middleware for mobile social computing: challenges, design, and early experiences. In *MOBILWARE '08: Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications*, 2007.

[3] E. M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, 2007.

[4] N. Eagle and A. Pentland. Social serendipity: Mobilizing social software. *IEEE Pervasive Computing*, 4(2), 2005.

[5] P. Hui, J. Crowcroft, and E. Yoneki. BUBBLE Rap: Social Based Forwarding in Delay Tolerant Networks. In *Proceedings of MobiHoc*, 2008.

[6] V. Kostakos and E. O. Neill. Cityware: Urban computing to bridge online and real-world social networ ks. In M. Foth, editor, *Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City*. 2008.

[7] A. Miklas, K. Gollu, K. Chan, S. Saroiu, K. Gummadi, and E. de Lara. Exploiting social interactions in mobile systems. In *Proceedings of UbiComp 2007*, 2007.

[8] R. Moskowitz and P. Nikander. Host Identity Protocol (HIP) Architecture. RFC 4423 (Informational), May 2006.

[9] A. Mtibaa, A. Chaintreau, J. LeBrun, E. Oliver, A.-K. Pietiläinen, and C. Diot. Are you moved by your social networks application? In *WOSN'08: Proceedings of ACM SIGCOMM Workshop on Online Social Network*, August 2008.

[10] T. Nicolai, E. Yoneki, N. Behrens, and H. Kenn. Exploring social context with the wireless rope. In *Proceedings of the OTM Workshop MONET*, 2006.

[11] E. Nordström, P. Gunningberg, and C. Rohner. A search-based network architecture for mobile devices. Technical Report 2009-003, Uppsala University, January 2009.

[12] P. Persson and Y. Jung. Nokia sensor: from research to product. In *DUX '05: Proceedings of the 2005 conference on Designing for User eXperience*, 2005.

[13] A.-K. Pietiläinen and C. Diot. Experimenting with opportunistic communications. In *MobiArch'09: The 4th ACM International Workshop on Mobility in the Evolving Internet Architecture*, 2009.

[14] S. Smaldone, L. Han, P. Shankar, and L. Iftode. Roadspeak: enabling voice chat on roadways using vehicular social networks. In *SocialNets '08: Proceedings of the 1st workshop on Social network systems*, 2008.

[15] J. Su, J. Scott, P. Hui, J. Crowcroft, C. Diot, A. Goel, E. de Lara, M. H. Lim, and E. Upton. Haggle: Seamless networking for mobile applications. In *Proceedings of UbiComp*, 2007.