P2P Systems Meet Mobile Computing: A Community-Oriented Software Infrastructure for Mobile Social Applications

Cristian Borcea
Department of Computer Science
New Jersey Institute of Technology
Newark, NJ, USA
borcea@cs.njit.edu

Adriana Iamnitchi
Department of Computer Science and Engineering
University of South Florida
Tampa, FL, USA
anda@cse.usf.edu

Abstract

The widespread adoption of powerful mobile devices creates an unprecedented potential for innovative mobile applications that can enhance users' social interactions. The current centralized mobile system and service architectures do not allow large-scale dynamic interactions between mobile devices, as required by these applications. This paper proposes Mobius, a decentralized solution that supports mobile social applications via a two-tier software infrastructure. In Mobius, a socially-aware peer-to-peer tier provides community-oriented data and persistent services for the mobile tier that runs the applications.

1. Introduction

The rich resources and the large-scale adoption of today's mobile phones create an unprecedented potential for innovative social applications. For instance, smart phones could "understand" the social context and silence themselves automatically during work or classroom meetings; they could monitor the vital signs of a person and call the nearest friend in case of an emergency; or they act like live video transmitters and broadcast to a dynamically inferred community of people interested in some content. More complex applications could take advantage of emergent community geo-social patterns, which could be inferred from a history of social interactions and places people visit. Such applications could: recommend newly identified groups or places to enhance their users' geo-social experience; personalize, in coordination with other mobile devices, an evacuation route from a building or a city in case of an emergency; and safely propagate software updates or curb malicious software dissemination.

So far, we have seen only simple social applications on mobile devices, such as mobile versions of social networking sites (e.g., Facebook or MySpace) and signaling matching interests between people using spontaneous one-hop ad hoc communication [1]–[4]. Developing more complex mobile social applications is currently limited by two major factors. First, similar to software development by hardware companies in the early years of computing, mobile services are provided only by the cellular network operators in collaboration with the phone manufacturers, thus limiting competition and variety. Second, the current system architectures for service provisioning are centralized and will hardly scale to millions of customers who would make data-intensive demands. Additionally, their centralized nature could discourage users from sharing geo-social information with a potential "big brother".

We propose Mobius, a self-organizing, self-adaptive, community-oriented two-tier software infrastructure that allows users to contribute their computers to provide persistent services for mobile applications in a distributed fashion. Mobius (i) provides support for computation offloading to increase the battery life of mobile devices; (ii) allows users to write their own mobile services; and (iii) collects and exploits social knowledge for system adaptation and support for mobile social applications.

2. Overview

Mobius consists of two tiers: (1) a Mobile (wireless) human-centric tier, which runs mobile applications and collects geo-social context information, and (2) a Peer-to-Peer (wired) system tier, which runs services in support of mobile applications and adapts to the geo-social context to enable energy-efficient, scalable, secure, and reliable mobile applications. Similar to traditional P2P networks, users will contribute their resources, both wireless mobile devices and wired PCs, to create this system. As a necessary departure from the dependence of mobile applications on cellular network operators, it allows mobile applications on phones (i.e., Mobile tier) to interact with user-deployed services in the P2P tier. This independence is achieved using community-supported services running on a collection of user-owned resources in a P2P manner.

Mobile-only solutions are not sufficient even if the devices stay connected to the Internet because the mobile devices can be turned off or run out of battery. The P2P tier provides

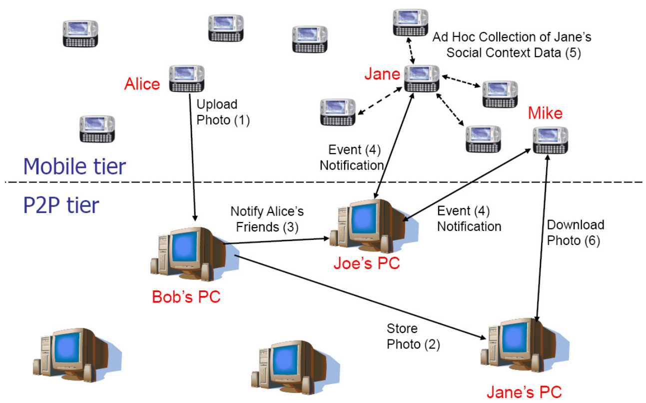

Figure 1. Application scenario: A community multimedia-sharing service

persistent services and more powerful resources (power, computation, applications and software services).

Mobius does not preclude existing or new services offered by network providers or third parties. To illustrate the novel features of Mobius, with a special emphasis on geo-social management and adaptation, let us assume the scenario in Figure 1:

User-deployed mobile services. Bob creates a new social multimedia-sharing service that enables mobile users to upload multimedia content from their phones and share it with their friends in real-time. The friends to receive the content can be specified according to the type and strength of social ties (e.g., friends, co-workers with strong social ties, ideological allies, or everybody personally known). Bob installs this service on his PC, which is part of the P2P tier. The service description and potential security/privacy policies are registered with the P2P tier. Then, Bob announces the service to his friends.

Data management. Alice decides to share a photo using Bob's service. She uses a mobile application, which can interact with multiple multimedia-sharing services. The application tags the photo with the name of the place and potentially its physical location read from the built-in GPS. When Bob's service receives the request to upload Alice's photo, it invokes a data management primitive in the P2P tier to store the photo. This primitive submits a data replication request that will place data on storage close to the destination communities specified by Alice.

Community interactions. Once the data is stored, Bob's service contacts an event notification service to alert Alice's specified social contacts about the photo. The smart phones of these contacts receive a "new content available" event and dispatch it to the corresponding application. The application then decides based on the current social context of the user whether to alert her immediately or postpone this action. To decide, the application can use ad hoc communication with nearby devices and the location of its device to infer the current social context. Furthermore, the mobile application can contact a service in the P2P tier to retrieve social information about the nearby people. If the social context is appropriate, the application alerts the user with an audio signal; otherwise, it just registers the event for later display. The transfer of the photo will only happen with the explicit user's consent.

Service management. While much of the user-produced media today is of limited interest outside the close social group, there are situations where, for example, phone-taken photos have potentially huge impact on public awareness[1]. Alice may decide to make the geo-tagged photo available to users independent of her social community. The geo-tag will be used in key-based searches by other users potentially interested in Alice's photo. The P2P tier can coordinate the replication of data to alleviate hot spots and improve data availability in response to significant popularity.

Application offloading. Finally, a user might decide to run a photo classification application on her smart phone. This application performs image processing to extract and compare image features. Since this is a computationally/energy intensive application, the mobile application can decide to offload the computation on a node in the P2P network. The offloading decision is negotiated with the P2P tier, which can grant or deny the request based on various policies and incentives that can include social knowledge.

3. Socially-Aware P2P Tier

The P2P tier is a *multi-purpose* P2P infrastructure aimed to provide support to the mobile users and their social applications. The innovative aspect of our approach lays in designing an infrastructure that *collects and exploits social knowledge* about and from the mobile tier above. Mobius incorporates social information deep into the self-organizing P2P structure to allow for adaptive mechanisms that better react to the social patterns of the mobile users.

The following objectives direct the design of the P2P tier: (1) Provide core services to mobile users, such as reliable data management, service discovery, search capabilities, and computing (e.g., image processing, path finding in the physical world); (2) Provide support for dynamic instantiation and management of user-defined services; (3) Maintain social state respecting user-specified privacy rules; (4) Allow support for dynamic node participation and heterogeneity via self-configuration.

Overall architecture. The P2P tier, as depicted in Figure 2, has several layers. The *Geo-Social P2P Services* layer supports the traditional P2P maintenance components, such

1. The recent political protests from Burma were photographed using cellular phones [5], and the photos were published in newspapers around the world

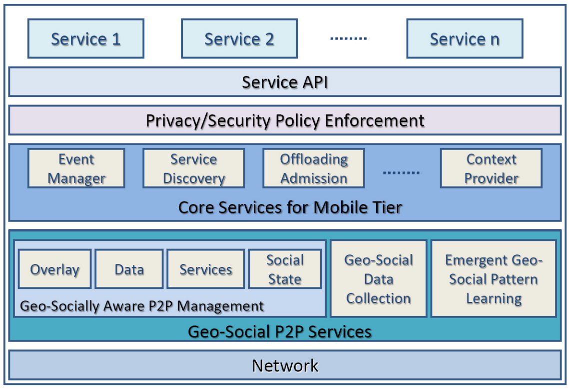

Figure 2. P2P tier architecture.

as overlay and data management. However, these components are now informed by geo-social data and emergent social patterns, information maintained by the Social State Management component. We consider using an unstructured P2P topology for flexible mapping of social context onto the lower layer networking infrastructure [6]. In particular, using an unstructured P2P topology allows peers to choose their neighbors based on trust inferred from the social network, and not imposed by the node ID space, as in the DHT-based overlays.

Since the P2P tier runs both core services and user-level services to support mobile applications, a Service Management component is necessary. This component decides, based on social knowledge, where to instantiate community services and where to replicate overloaded services.

The *Core Services for Mobile Tier* layer includes services for supporting mobile applications such as Service Discovery, Event Manager, Offloading Admission Manager, and Social Context Provider. While the first two services are well understood, the Offloading Admission Manager and Context Provider are novel to our architecture. The Offloading Admission Manager coordinates the offloading of applications from the mobile tier to the less resource-constrained P2P tier. It decides whether to grant application offloading requests, where to place the offloaded applications, and it coordinates the transmission of the results back to the mobile tier. The Context Provider service exports geo-social data and emergent patterns to the software levels above, subject to privacy and security policies enforced by the layer above. The Context Provider allows therefore the development of new, secure socially-aware user services supported through a service API.

In the following we sketch the research challenges for providing a socially-aware P2P infrastructure.

Collecting social data and inferring social context. Inferring social information from the use of technology is possible due to two main factors. First, the highly popular online social networking applications, such as MySpace or Facebook, can provide rich information about the social fabric of the Internet users. Second, the mobile devices equipped with communication and localization capabilities can help refine reported social relationships.

Online social networking applications reportedly include hundreds of million of users from around the globe [7]. According to the Facebook website [8], more than 68 million users are active in the network, spending on average 20 minutes per day on the more than 18,000 applications now available. Users participate in these social networks by declaring their relationships with other users of the network. While many of these networks are built for socializing, some are specialized on topics such as political debate (Essembly [9]) or professional networking (LinkedIn [10]). When specialized, the relationships carry the semantics specific to the topic of interest: e.g., Essembly users semantically distinguish between friends, allies, and enemies [11].

Using only information from online social sites, however, and mapping trust onto declared relationships can be misleading: due to the social incentives hidden in declaring relationships, such declarations are over-inflated. As a consequence, a declared "friendship" relationship may not carry the amount of trust typically expected between friends. Augmenting these declarations with information from mobile devices (collected and processed by the *Geo-Social Data module*) will build a more accurate representation of the social world. We note that inferring social information from mobile devices alone is limited by the difficulty to differentiate co-location from social interaction, especially in densely populated environments, such as an urban campus [12].

Inferring communities. Social state management. Social network analysis has been intensely studied recently [13]–[15] and topological properties such as power-law and small world have been found in networks ranging from high school friendships [13] to citation networks [16]. Moreover, Girvan and Newman [17] showed that tightly-knit groups (such as a research group in a specific field) can be identified by network analysis.

However, due to privacy rules, full knowledge about the social graph may not be available in Mobius. Collecting social data from online social applications can be done only with the user's consent. In a "big brother" scenario, global social data is centralized. At the other extreme, each user maintains her own host that stores geo-social information about all her relations. A real scenario will be somewhere in between, with hosts maintaining overlapping social state, but none having a global view of the social network. Social state will thus be partitioned on multiple peers. In such a case, several issues are raised: (i) how to collect and group data in order to infer the social group of a particular user; (ii) how to map social data onto different peers; (iii) how to name and store inferred communities for later use.

One way to map social state onto peers is based on social ties for better privacy (since friends already know each other's information) and better locality (resources of a community may be grouped together). Ideally, community-specific state and services are mapped onto community-owned peers. Since people are typically part of multiple social groups (family, co-workers, friends), it is likely that services from different communities may be co-located on some common set of hosts. We stress that social knowledge informs the choice of neighbors of a peer, but lack of social relationships between two nodes will not prevent them from collaborating in the overlay. Mapping social state onto hosts can also be considered based on geographical locality.

Socially-aware service and data management. Social ties can be used to infer and provide incentives for participations, such as hosting services or storing user-generated content. Many issues are raised by this idea, however: How many services need to be supported by a community? How to deal with overlapping communities? Alternatively, can we use relationship transitivity to provision for a low-resource community?

Large or rapidly growing social communities may also pose scalability challenges: services will need to be replicated dynamically as a response to growing popularity. This requires monitoring resources in order to automatically trigger service replication and employing techniques for service placement based mostly on slowly-changing attributes. For example, mobile users change their location frequently, but perhaps less so the geographical area (city or state); social relationships are typically slowly changing. Candidates to host services will be selected based on social ties (providing incentives and trust) and resource capabilities (including network characteristics and geographical location). Service replication might require replicating service data as well. In particular cases, however, the service data might be already replicated on the peer where the new service is instantiated. This behavior is due to the use of socially-aware decisions for both data placement and service placement.

The data management component is used for durable storage, content delivery, and retrieval (key-based searches). In addition to data placement and replication, there is also the question of durability. For example, the multimedia to be delivered by Bob's service may not guarantee durability above some limited period. Data that migrate closer to the mobile user is not to stay there for long, yet what is a useful time interval? Is a community repository necessary for providing useful durability?

4. Mobile Tier

Figure 3 presents the components of our common mobile platform, which exports an API for context-aware mobile applications, collects geo-social context information, and collaborates with the P2P tier for geo-social adaptation. There will be two communication planes between the Mobile tier and the P2P tier: the application plane, between user-level applications (on mobile devices) and services (on PCs), and the system plane, between various modules of the mobile platform and core services or modules in the P2P tier. Mobile devices collect and share with the P2P tier three types of information: their location, their resources, and their neighboring devices/users.

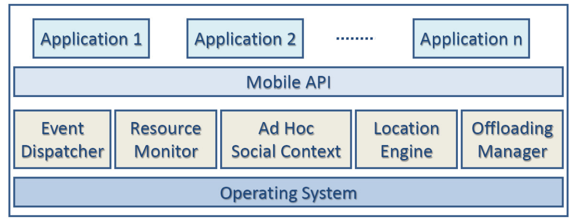

Figure 3. Mobile Node Architecture

Location collection and sharing. Most mobile applications need their users' location for basic context-awareness. Ideally, a location system should be accurate, scalable, cost effective, easily deployable, work both indoors and outdoors, and allow users to control the sharing of their location. No currently available system has all these features, and in fact, some of the features are contradictory. Users would understandably be concerned if a certain hardware/software infrastructure would track them to determine their real-time location. Therefore, the Location Engine module will determine the location locally using GPS or software solutions such as Intel Placelab [18], which work both indoors and outdoors. In this way, it allows users to control when and how often location updates are sent to the P2P tier. Since computing and sending location could consume significant power, social knowledge can be used to decrease the frequency of the updates or even to turn off the engine for certain periods of time (e.g., attending weekly meetings or classes).

Application offloading. The Resource Monitoring module keeps track of the available local computing and communication resources, with a special emphasis on battery power and memory. This module is used by local context-aware applications to adapt their functionality. Furthermore, it can be used by the Offloading Manager to provide support for application or component offloading [19], [20] when the local resources are insufficient or the battery costs are too high. The offloading process is done in collaboration with the Offloading Admission service in the P2P tier, which decides whether to allow the offloading and on what host, based on geo-social data (e.g., user's PC or user's friend's PC) and privacy preferences. Trade-offs between computation and communication with respect to energy consumption must be taken into account in order to ensure that offloading provides the expected benefits.

Ad hoc collaboration. Mobile devices communicate over the Internet (cellular or WiFi) with the P2P tier; even the communication between mobile devices takes place indirectly through this tier. The only exception when they communicate directly is the Ad Hoc Social Context module, which uses one-hop ad hoc communication (WiFi or Bluetooth) to gather social information. This module can use a protocol that simply learns the MAC addresses of neighboring devices or can learn user identities associated with those devices. In the former case, stronger privacy is ensured as only the P2P tier would know the association between MAC addresses and user names, but less functionality is available on the mobile. For instance, learning the current social context will always have to be done by the P2P tier. In the latter case, mobile devices can infer this context locally, especially if they store other social information as well; however, maintaining privacy could be more complex. Gathering local social context information can be done periodically or on-demand, with a trade-off between information accuracy and power consumption.

Asynchronous communication. Mobile applications communicate synchronously (i.e., request/reply) with services. However, they often need to be contacted when certain social, temporal, or geographical conditions are met. For example, an application might want to be notified in real-time of the co-location of two users, the presence of a user at a given place, or a new person or place recommendation based on newly identified geo-social affinities. To achieve this asynchronous communication goal, applications (or user-level services on their behalf) register events with the Event Manager service in the P2P tier. Event notifications sent by this service are received by the Event Dispatcher module on the mobile device, which subsequently delivers them to the corresponding applications.

5. Summary

This paper proposes two main ideas for discussion. First, it proposes to use a peer-to-peer infrastructure for supporting mobile social applications. This is a fundamental departure from the vertically integrated solutions of content distribution networks for which P2P systems became popular, as it requires service instantiation and management in addition to data management. Second, it proposes to use geo-social knowledge as the basis for self-organizing, self-adaptive resource and data management techniques in a P2P infrastructure. Additionally, social knowledge can be used for privacy protection and providing incentives for resource contribution. Incorporating social knowledge in system design can open new directions of research and bring significant potential performance improvements over techniques that use only system metrics.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grants No. CNS-0831753 and CNS-0831785. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] "Lovegety: Love Japanese style," http://www.wired.com/news/culture/0,1284,12899,00.html.

[2] "Nokia sensor," http://www.nokia.com/sensor/.

[3] N. Eagle and A. Pentland, "Social serendipity: mobilizing social software," *Pervasive Computing, IEEE*, vol. 4, no. 2, pp. 28–34, 2005.

[4] M. Terry, E. D. Mynatt, K. Ryall, and D. Leigh, "Social Net: Using Patterns of Physical Proximity Over Time to Infer Shared Interests," in *Proc. Human Factors in Computing Systems (CHI 2002)*, 2002, pp. 816–817.

[5] "Bbc burma protests: Your pictures (2007)," http://news.bbc.co.uk/2/hi/in_pictures/7012813.stm.

[6] S. Marti, P. Ganesan, and H. Garcia-Molina, "DHT routing using social links," in *3rd International Workshop on Peer-to-Peer Systems (IPTPS 2004)*, 2004. [Online]. Available: citeseer.ist.psu.edu/article/marti04dht.html

[7] "Réseaux sociaux: des audiences différentes selon les continents," *Le Monde*, January 14, 2008, http://www.lemonde.fr/web/infog/0,47-0@2-651865,54-999097,0.html.

[8] "Facebook(2004)," http://www.facebook.com.

[9] "Essembly(2006)," http://www.essembly.com.

[10] "Linkein(2002)," http://www.linkedin.com/.

[11] T. Hogg, D. Wilkinson, G. Szabo, and M. Brzozowski, "Multiple relationship types in online communities and social networks," in *AAAI Spring Symposium on Social Information Processing*, 2008.

[12] "Machine perception and learning of complex social systems," http://reality.media.mit.edu/.

[13] M. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[14] A.-L. Barabási, *Linked: The New Science of Networks*. Perseus Publishing, 2002.

[15] S. Dorogovtsev and J. Mendes, "Evolution of networks," *Advances in Physics*, vol. 51, no. 4, pp. 1079–1187, 2002.

[16] M. Newman, "The structure of scientific collaboration networks," *Proc. Natl. Acad. Sci. USA*, vol. 98, pp. 404–409, 2001.

[17] M. Girvan and M. Newman, "Community structure in social and biological networks," *Proc. Natl. Acad. Sci. USA*, vol. 99, pp. 8271–8276, 2002.

[18] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Placelab: Device positioning using radio beacons in the wild," in *Proceedings of the Third International Conference on Pervasive Computing (2005)*, May 2005, pp. 116–133.

[19] A. Messer, I. Greenberg, P. Bernadat, D. Milojicic, D. Chen, T. Giuli, and X. Gu, "Towards a Distributed Platform for Resource-Constrained Devices," in *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS 2002)*, July 2002.

[20] R. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H. Yang, "The Case for Cyber Foraging," in *Proceedings of the 10th ACM SIGOPS European workshop: beyond the PC (EW10)*. ACM Press, 2002, pp. 87–92.