U-Hopper: User-centric Heteogeneous Opportunistic Middleware

Iacopo Carreras and David Tacconi
CREATE-NET
Via alla Cascata 56/C
Trento, Italy
name.surname@create-net.org

ABSTRACT

This demonstration presents U-Hopper, a user-centric heterogeneous opportunistic middleware specifically tailored to the diffusion of user centric information, such as contextual and entertainment data, in opportunistic environments. The proposed platform exploits proximity wireless interfaces available on most commercial mobile devices for disseminating data among mobile users. Such diffusion if driven by the specific interests of mobile users, combined with the intrinsic locality of data being generated in such pervasive environments. The prototype is developed over java-enabled smartphones and relies on Bluetooth connectivity for achieving proximity communications. In this paper, U-Hopper is described in all its functional components, together with the details of its software implementation.

Keywords

opportunistic communications, middleware platform, mobile phones, bluetooth, J2ME.

1. INTRODUCTION

Opportunistic networking [1] refers to the possibility of delivering data applying an epidemic-like forwarding mechanism, without the need for any dedicated infrastructure. Such communication paradigm received great attention in the last few years as an emerging technology for disseminating data in challenged environments, where due to environmental constraints it is not possible to build an alternative communication infrastructure, or in pervasive environments, where data exchanges are driven by the "social interactions" of mobile users [2]. In particular, the latter case is a direct consequence of the fact that mobile devices (e.g., smartphones or PDAs) are nowadays largely available among people and of the constantly increasing computing, communication and storage power of such devices. Several mobile phones are in fact equipped with Bluetooth, Wi-Fi and Wibree (in the near future), technologies that are directly accessible for programmers through freely available and easy-to-use APIs. Further, mobile phones are now capable of intensive processing operations and of storing large amounts of data in their internal memory.

However, although opportunistic networking has been deeply investigated from a theoretical point of view, only few real deployments have been proposed, with the main goal of understanding the networking performance of the implemented protocols [3] or the social aspects related to proximity communications [2].

Following these considerations, we have developed a *User-centric Heterogeneous Opportunistic Middleware* (U-Hopper), running on any java-enabled smartphone and leveraging Bluetooth connectivity for exchanging data. Such platform combines the user preferences with the requirements imposed by the pervasive services hosted on users' portable device in order to gather and disseminate data. Such process is fully distributed and self-organized, as it does not require any human supervision.

In the proposed demo, we will present a prototype of the U-Hopper platform. The demonstration will comprise three classes of devices. The first one is constituted by sensors embedded in the environment and constantly broadcasting localized information such as snapshots of the conference site or advertisements. Such information is gathered by the second class of devices, which consists of a limited number of smartphones acting as mobile nodes and running the U-Hopper platform. Through these devices the information will flow around by means of P2P data exchanges among mobile nodes running the U-Hopper middleware. Finally, information is displayed over a laptop, the third class of devices, which provides a richer user-interface through which the information captured by the U-Hopper platform is displayed.

2. SYSTEM OVERVIEW

The considered system architecture consists of two classes of nodes: *User-Nodes* (U-Nodes), which are resource-rich mobile devices (e.g., smartphones, PDAs) carried around by users during their daily activities, and *Tiny-Nodes* (T-Nodes), which are resource-constrained devices embedded in the environment and providing localized information [4]. A part from their technological differences, the 2 classes of devices play a different role in the network. T-Nodes act as *providers of information*, constantly broadcasting localized information such as advertisements, or snapshots of the surrounding environment. Conversely, U-Nodes act as *consumers of information*, reading T-Nodes in their communication range, and augmenting pervasive services with such

Bionetics'07 December 10-13, 2007, Budapest, Hungary

data. The data generated by T-Nodes is first stored in the U-Nodes internal memory, and then diffused by means of opportunistic peer-to-peer data exchanges. Users' mobility is therefore exploited in order to achieve system-wide communications. In this demo, we will present a U-Hopper, a User-centric Heterogeneous Opportunistic Middleware initially introduced in [5]. Such platform resides on U-Nodes and exploits any proximity communication interface (i.e., Bluetooth, Wi-Fi) in order to (i) gather localized information originating from T-Nodes embedded in the environment (ii) opportunistically disseminate the stored data to other U-Nodes. The information diffused includes data received from T-Nodes as well as any other information shared by the user (e.g., music, videos, etc.).

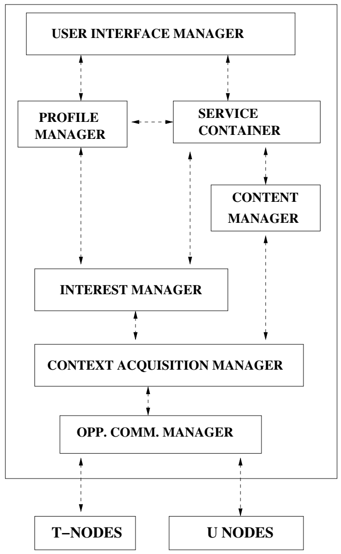

Figure 1: Block diagram and interconnections of the U-Hopper platform.

As depicted in Fig. 1, the U-Hopper middleware is composed by six distinct components.
The *User Interface Manager* (UIM) handles any human-to-machine interaction such as data insertion and visualization.
The *Service Manager* (SM) is the execution environment where pervasive services are running. This component allows services deployment, deprecation and update.
The *Interest Manager* takes into account (i) the user preferences and (ii) the requirements deriving from the pervasive services hosted by the SM, and produces a list of "interests", which are a high-level description of the information the user is interested in. Such interests regulate the way according to which information is exchanged between any 2 U-Nodes accordingly.
The *Content Manager* (CM) manages the persistent storage available on mobile devices. In particular, it provides context data insertion/deletion, update and search functionalities. In addition, the CM runs appropriate "data aging" algorithms that are needed in order to discard outdated information and preserve the available resources. Such techniques implement information filtering rules that trade off data locality (both in the time and space domains) for available resources (i.e., storage, communication, etc.) [6]. As an example, we can think at a special sale offer ending at 5 pm of the current day. Clearly, as soon as the offer is no longer valid, it is useless to store the corresponding information. The CM is in charge of detecting such situations and of determining when to remove data from the users' device permanent storage.
The *Context Acquisition Manager* (CAM) takes into account the information deriving from the Interests Manager, and applies data filters on the incoming and outgoing data flows.
The *Opportunistic Communication Manager* (OCM) monitors the availability of data sources in the surrounding environment, and seamlessly performs any networking operation needed for gathering the discovered data. This includes data originating from T-Nodes as well as from other U-Nodes. Please, refer to [5] for a more detailed description of U-Hopper system components.
Fig. 2 depicts the handshake regulating the data exchange between any 2 nodes meeting. The data exchange is triggered by a Node 1 receiving a beacon message used for discovering neighboring peers. In response, Node 1 sends its own interests (Interests 1 MSG), which are a description of the information Node 1 is interested in. Node 2 responds with the data stored in its own internal memory matching Node 1 interests (DATA 2 MSG), and subsequently, with its own interests (Interests 2 MSG). Finally, the data exchange is terminated with Node 1 sending any data matching Node 2 interests. The corresponding system components interaction flow, when generating the user interests, is presented in Fig. 3. When a beacon message is received by the OCM, a request for the user interests is invoked. Such request is then captured by the IM, which gathers the user profile, the service constraints and returns the user interests. It is worth remarking that the described actions are performed by the U-Hopper platform transparently to the user, thus increasing the system usability of the system, since it does not require any human intervention. On the counterpart, when the interests from an encountered node are received, the data stored in the internal memory is searched accordingly, and information matching the received interests is send back. The corresponding system components interaction flow is depicted in Fig. 4.

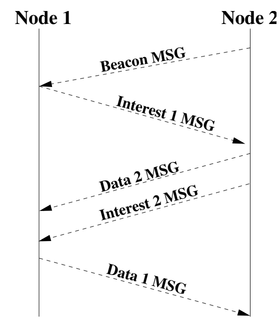

Figure 2: Data handshake between any 2 nodes meeting.

3. IMPLEMENTATION DETAILS

The main goal of the proposed demo is to show a prototype implementation of the described system over widely diffused software/hardware platforms. In order to embrace the largest number of "potentially available" mobile devices, we selected smartphones as the target platform over which we developed U-Hopper. In fact, smartphones are nowadays

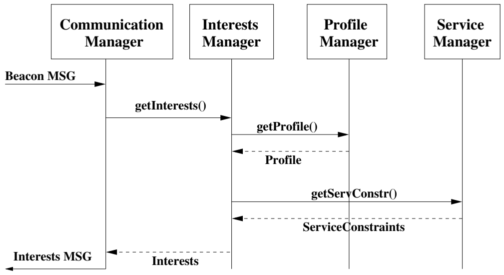

Figure 3: System components interaction flow, when generating the user interests.

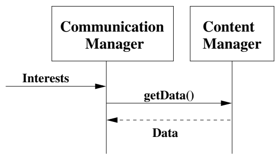

Figure 4: System components interaction flow when retrieving data, starting from user's interests.

typically carried around by users during their daily activities and at the same time, they have reached a sufficiently large computing and communication power to perform very complex operations. We also decided to use some laptops for broadcasting advertisement information or collecting application's statistics.

In order to leverage on a widely diffused and standardized computing environment, we choose to develop U-Hopper as a java Midlet running over J2ME (MIDP profile 2.0) [7], which is currently available on most of the smartphones shipped today. By implementing U-Hopper as a java component, we are guaranteed that the software will be portable over a large set of devices.

The fist technological issue to be solved in developing a middleware for opportunistic environment is given by the selection of the proper network interface. In fact, as detailed in Sec. 2, opportunistic communications are the primary mean by which information is diffused in the described environment. Currently, Bluetooth is the largest available network interface on mobile phones and can be easily accessed through J2ME dedicated APIs, such as the well-known JSR 82 ([8]). Hence, we choose to rely on this technology for achieving localized peer-to-peer data exchanges among mobile nodes. Obviously, Bluetooth is not properly designed for opportunistic communications, given the amount of time typically required for establishing a connection between two devices. To shorten up this connection time, we leveraged on some assumptions and on few properties of the Bluetooth technology. At first, we have assumed that all the devices running U-Hopper are assigned with a friendly name starting with a given prefix (for instance they all start with 'uhopper'): this simple assumption allows U-Hopper clients to recognize immediately a candidate that runs the same service, avoiding useless attempts of connection with other devices. We assume also that all devices keep a list of recently visited devices in order to avoid too frequent connections with the same peers. A peer is periodically removed from this list, when a given timer is elapsed.

In U-Hopper, each device is always working in server mode, i.e. it is waiting for incoming connections from U-Hopper clients. Alternately, a device can work in client mode, periodically inquiring close-by devices, and consequently inhibiting its server mode until the client operations are not completed. Whenever another device is found during an inquiry, if its friendly name starts with the given prefix and it was not recently visited, the inquiry process is stopped and the service discovery operation is performed on that device. This peer is now inserted in the list of recently visited devices. If the U-Hopper service is found and it is available, a connection is established and the communication handshake previously described can start. The node working in server mode updates its list of recently visited devices, in order to avoid a new exchange of information with the same peer in a brief time. Following this procedure, two peers can establish a connection in approximately 1 or 2 seconds (depending on the number of Bluetooth devices in radio range), a time suitable for having opportunistic communication on Bluetooth enabled devices.

The second issue we had to face was to design a proper persistent storage on each device. Typically, in smartphones data are saved locally using the Record Management Store (RMS), where information can be easily stored as an array of bytes and retrieved using easy-to-use matching methods, similar to common data base queries. We design then a RMS storage for each device, allowing U-Hopper to maintain persistent data, interests and profiles through different execution of a service running on top of the middleware.

Concerning the implementation of U-Hopper on laptops, we used the Avetana Bluetooth library, an open source Bluetooth stack that allows applications designed for JSR-82 to run on Linux devices. Although we had to adapt some of the components to a not-embedded environment, as for instance the local storage and the user interfaces, most of the modules of U-Hopper were easily installed and run on several laptops. Laptops were used as T-Nodes generating advertisement information or as collection points, for showing application statistics, running the same communication module described before.

Tab. 1 presents a concise summary of the technologies and devices used in this demonstration.

HW platform	Nokia $E65$, Nokia N80, Dell600
OS	Symbian OS 9.1, Ubuntu 6.11
SW platform	J2ME (MIDP 2.0), J2SE 1.5
BT version	Bluetooth 1.2

Table 1: The demo in a nutshell.

4. DEMONSTRATION OVERVIEW

The proposed demonstration is depicted in Fig. 5. It consists in (i) few smartphones and laptops acting as data sources (T-Nodes), (ii) 4 smartphones running U-Hopper (U-Nodes), (iii) 1 laptop acting as a mobile node (U-Node) for collecting application's statistics. In order to reproduce the considered pervasive application domain, we have used a few Bluetooth enabled smartphones acting as environmental data sources (T-Nodes). Such smartphones are regularly broadcasting physical data, such as measured temperatures or snapshots of the conference site periodically taken by smartphones' cameras. Few laptops are broadcasting context data such as conference social and technical activities.

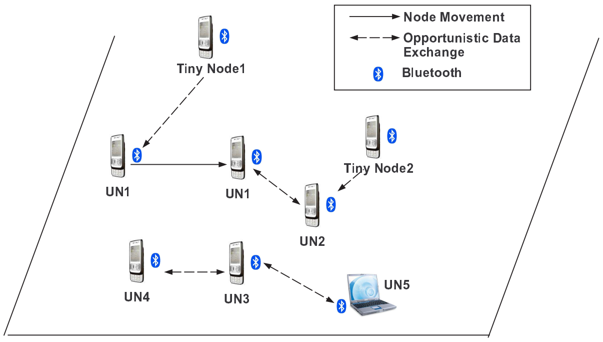

Figure 5: Demonstration scenario.

Context data is then collected by other smartphones (U-Nodes) passing by, as for instance *UN1* gathering data from *Tiny Node 1* in Fig. 5. Information is then epidemically diffused by means of opportunistic P2P data exchanges among U-Nodes that take place transparently to the user.

After installing and starting the application, users will be asked to select their preferences. Such preferences will influence the data gathering and data exchange process. Users can then visualize the collected information, such as snapshots and advertisement information, at any time on the smartphone's graphical interface Furthermore, U-Hopper users will have the opportunity to share some information in a totally P2P fashion, directly acting as source of data. For instance they can share some personal midi files (e.g. ring tones), exchange them with other peers if interested in entertainment information and then listen to the received files through a simple UI. In Fig. 6 two users are shown while using U-Hopper.

In order to show the outcomes of the demo (such as collected information and statistics on users P2P communications) in a more user-friendly way, users can visualize at any time such information through an application running on the laptop acting as a user node shown in Fig. 7), available for everyone interested in our application .

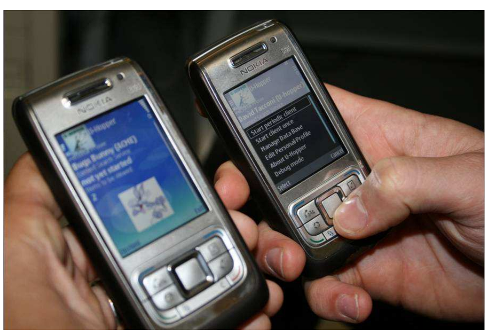

Figure 6: U-Hopper on Nokia E65 smartphones.

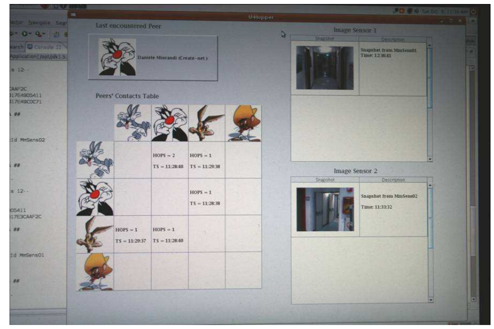

Figure 7: The laptop interface, showing contacts among 4 users and data gathered from T-nodes taking snapshots of the site.

5. CLOSING REMARKS

The developed U-Hopper is an attempt to provide a reusable software platform capable to transparently handle opportunistic data dissemination among mobile nodes over off-the-shelf hardware and software platforms. Fully distributed pervasive services can leverage such *disappearing network infrastructure* [4] for retrieving context data without the need of any human supervision. In the presented demo, we have shown a simple application of the U-Hopper middleware. In particular, it is the capability of U-Hopper to exploit the physical mobility of user in order to gather information and diffuse it to other mobile nodes. The platform has been developed over java-enabled smartphones, leveraging Bluetooth connectivity for achieving proximity communications.

6. ACKNOWLEDGMENTS

This work has been partially supported by the EC within the framework of the BIONETS project IST-FET-SAC-FP6-027748, `www.bionets.eu`

7. REFERENCES

[1] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Comm. Mag.*, vol. 44, no. 11, Nov. 2006.

[2] M. Nicolai, N. Behrens, and E. Yoneki, "Wireless rope: An experiment in social proximity sensing with bluetooth," in *Proc. of PerCom*, March 2006.

[3] X. Zhang, J. Kurose, B. N. Levine, D. Towsley, and H. Zhang, "Study of a Bus-Based Disruption Tolerant Network: Mobility Modeling and Impact on Routing," in *Proc. of Mobicom*, September 2007, pp. 195–206.

[4] BIONETS, "Biologically-inspired autonomic networks and services," http://www.create-net.eu, 2005.

[5] I. Carreras, D. Tacconi, and D. Miorandi, "Data-centric information dissemination in opportunistic environments," in *In Proc. of MASS*, Pisa, Italy, October 2007.

[6] I. Carreras, I. Chlamtac, F. D. Pellegrini, and D. Miorandi, "Bionets: Bio-inspired networking for pervasive communication environments," *IEEE Trans. on Vehicular Technology*, vol. 56, no. 1, pp. 218–229, Jan. 2007.

[7] Jsr-000118 mobile information device profile 2.0. [Online]. Available: http://jcp.org/aboutJava/communityprocess/final/jsr118/

[8] Jsr-000082 javatm apis for bluetooth. [Online]. Available: http://jcp.org/aboutJava/communityprocess/final/jsr082/