Position Papers of the 2013 Federated Conference on
Computer Science and Information Systems pp. 57–62

A Middleware Architecture for Mobile Social Networking with Intelligent Energy Saving

Luiz M. M. de A. Santos, Maria A. S. N. Nunes, Admilson de R. L. Ribeiro
Computing Departament
Universidade Federal de Sergipe, UFS
São Cristóvão, Sergipe, Brazil
Email: {luizm1000, gutanunes}@gmail.com, admilson@ufs.br

Abstract—In recent years, middleware for social mobile networking has attracted the attention of academia, causing the design and development of various approaches by researchers. This type of middleware facilitates and makes more efficient the development process of mobile social networking applications. Furthermore, middleware solutions also abstract the communication process with other applications, allowing the acquisition, persistence and reuse of social context information and location of users, besides providing an API so that developers can access this information quickly and build new social applications. With a view to supporting this new trend of research, in this paper we propose a middleware architecture for mobile social networking called My-Direct which will make use of Wi-Fi Direct as a solution to improve the connectivity between the nodes of the social network. Along with My-Direct also created an intelligent mechanism for the management of mobile resources in order to reduce energy consumption.

I. Introduction

THE growing interest in social networks and smartphones has resulted in a new trend called MSN (Mobile Social Network). The goal of these social networks is to contribute to human interactions, allowing users who are socially related to use their mobile devices to perform activities of common interest. Some examples of MSN are: Google Latitude[1], Path[2] and Instagram[3].

In recent years, the introduction of a middleware on the development of MSN has attracted the attention of academia, leading the design and development of various approaches by researchers. In [1] it is possible to find some of these solutions, already in [2] describes a solution developed by Brazilian researchers.

A Middleware for MSN (MMSN) should consider the limitations of mobile devices such as limited power, low memory capacity, limited processing power, scalability and heterogeneity [3]. Furthermore, MMSN should provide a layer that provides common services needed by different MSN applications and separate social network management concerns from application requirements [10]. A MMSN should enable efficient operations of MSN application by being self-configuring, self-adapting, self-optimizing and self-protecting.

Being MMSN a topic of recent research, many of the problems mentioned before doesn't have an ideal solution. Thus, current middleware solutions are incomplete, and doesn't exist middleware infrastructure that solves all problems [1].

In order to contribute to this new trend of research, in this paper we propose a middleware architecture for MSN called My-Direct that will make use of Wi-Fi Direct [4] as a solution to improve the connectivity between the nodes of social network. This work also aims to create an intelligent mechanism for the management of mobile device resources to provide a reduction in power consumption.

II. Requirements for a Middleware for MSN

According to [5], an MSN is the social network where one or more individuals with common interests, chat and connect through the use of a cell phone. Also according to [5], there are two possibilities for MSN. The first possibility is that the network be designed from the first moment to be used on mobile devices. The other possibility is a hybrid, that is, the network was first developed for the Web environment and, over time, was migrated to the mobile platform.

Another observation regarding the design of an MSN is related to how it will be accessed by users. Basically, there are two approaches each of which has its positives and negatives. The first would be the distribution of client software, this approach has the benefit of easy access information persisted in the user's device, but it ends up consuming more resources and requires compatible versions with various mobile platforms. The second approach would be the availability of social networking on the Web, where the user would use the browser to access your profile. However, despite access the browser extend the use of different network platforms, contextual information about the user will not be easily extracted and the use of native

[1] Google Latitude. http://www.google.com/mobile/latitude/
[2] Path. http://www.path.com
[3] Instagram. http://www.instagram.com

features of each mobile platform, such as the camera, is restricted.

The MSN must also submit at least two context information related to users: the social context and location context [6]. Through these information, users have the possibility to realize the proximity of friends, members of a group, or profiles that have similarity to user in real time.

The introduction of a middleware facilitates and makes more efficient MSN application development process. Furthermore, the middleware also eases access to data, so that heterogeneous applications have access to them. However, developing a middleware layer for mobile social applications is not an easy task, because it presents a number of issues and challenges that need to be taken into consideration, which, according to [1], are as follows:

- **Simplifying the development process:** A MMSN should be responsible for simplifying the process of developing an application MSN. The simplification of this process can be obtained by providing high-level abstractions with lightweight interfaces to mobile application developers. Facilitate application integration and reuse must also be functions of the middleware.
- **Energy Efficiency:** A MMSN should provide mechanisms to efficiently use the resources of the battery and ensure that the application performs well on mobile devices with limited resources.
- **Privacy:** A MMSN should provide a simple technique that consumes fewer resources to implement appropriate control policies on the exchange of social data, to ensure user privacy.
- **Scalability:** MSN applications suffers increased number of users constantly. This number of nodes should not affect application performance. Thus, a MMSN must be flexible, so that it can manage the increasing number of nodes without compromising system performance.
- **Fully distributed architecture:** A MMSN should be designed to be fully distributed without centralized control. Specifically, it should be built to be used in ad hoc networks without assuming centralized servers.
- **Heterogeneity and dynamic nature of mobile devices:** A MMSN should be designed in a fully distributed fashion, considering the heterogeneous and dynamic nature of mobile devices, as well as privacy concerns. A MMSN should hide the heterogeneity of applications and allow adaptation to dynamic environments.

III. The Middleware Architecture My-Direct

The construction of mobile social networks is a complex task. The introduction of middleware in the field of mobile social networking aims to facilitate the development of this type of network by providing features that help developers manage the users, the maintenance of social relationships between users, in data privacy and social communication between network devices.

With the emergence of new technologies for communication between devices such as Wi-Fi Direct, becomes interesting introduction and evaluation of these new solutions in the middleware environment for MSN. Thus, this work aims to develop the My-Direct, a middleware for MSN that will make use of Wi-Fi Direct, aiming to bring independence of access infrastructure and improvements in the coverage of the social network and in the transmission rate of data between nodes.

The Wi-Fi Direct specification was developed by the Wi-Fi Alliance and operates in 802.11 devices, but is not linked to any specific standard 802.11. This specification introduces the ability to direct connection to millions of devices already have Wi-Fi deployed [4].

According to [4], the introduction of Wi-Fi Direct devices extends the Wi-Fi in order to provide a new connectivity experience. This technology increases the portability of content and applications across all devices of the user through a single and common specification, allowing users to access movies, music and photos point-to-point. Equipment vendors and content providers also benefit from the development of multiple applications that can be seamlessly interconnected wherever you go.

The Wi-Fi Direct is also based on the strengths of Wi-Fi such as performance, security, ease of use and ubiquity, and it adds features such as no need for access to a network infrastructure [4]. Instead of first connect to a network infrastructure and then connect to another device on the network, users can connect directly to devices that offer the services they need. This allows, for example, a user show the photos on your smartphone to your friends by connecting to a television and viewing the images, regardless of the presence of a network infrastructure that is available for both devices.

Although the My-Direct already have their specified architecture, it is still in the implementation phase. In this section we will only describe its layers, its main features and their modules. We will leave the complete encoding and the results for another occasion.

A. Architecture

The middleware My-Direct is being implemented in a P2P architecture targeted the Android platform. The fact that it targeted the Android platform implies the choice of a minimum version of operating system support. For this, we chose to develop the My-Direct to version 4.0 of Android, since the manipulation of resources and support Wi-Fi Direct are present in their SDK (Software Development Kit).

The Fig. 1 shows the My-Direct architecture. Here, we can see that the architecture is composed of four layers: (i) interface, where will stay the set of classes responsible for building the GUI (Graphic User Interface), (ii) communication, where will get located the API that will assist in linking between devices, (iii) privacy, that will

serve to identify users and verify their degree of friendship (iv) modules, which will serve to extend the middleware.

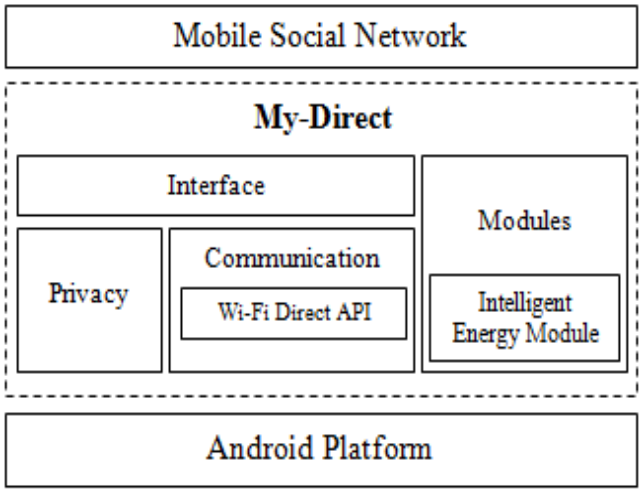

Fig. 1. The My-Direct architecture.

1) Interface Layer

As mentioned earlier, the My-Direct will run on the Android platform. This implies that all classes of the interface will have to follow the pattern of this platform. Thus, the interface layer will contain the implementation of classes that extend the Activity class of the Android SDK. These classes will be responsible for the interaction between the user and My-Direct. Thus, all functions of the My-Direct will be available on the screen and the user only has to select one of them and see the result.

It is important to remember that in the Android platform the design of classes is defined in XML (eXtensible Markup Language). These XML also serves to construct menus, the definition of strings and assigning images to display an Android application. Therefore, in the My-Direct interface layer will be stored implementations of the GUI classes and their respective XML.

2) Communication Layer

The communication layer consists of the classes responsible for the association and effective communication between devices. To perform these actions, classes of this layer will make use of Wi-Fi Direct API, which is provided by the Android SDK. This layer will possess key features like detection of mobile devices with Wi-Fi Direct, the association between them, extracting information (name, IP and MAC) of these devices and effective data exchange.

When occurs the proximity between two devices, the communication layer will check if they are with Wi-Fi Direct enabled. If so, will allow the devices to perform the association and begin the process of information exchange. This whole process will be controlled and monitored by users via interface layer.

3) Privacy Layer

Since the goal of My-Direct is to provide communication between nodes in an MSN regardless of the presence of an access infrastructure in your project we need to take into account the association between mobile devices and their owners have some affinity before starting social activities in the network.

There are several works in the literature such as [7] and [8] that address the issue of how to infer social relationships in a mobile environment. In the case of My-Direct, the independence with respect to the access infrastructure does not allow access to online social networks for information extraction, for example. Thus, it would be wiser to evaluate the affinity between users through information already contained in the mobile device. For this, we chose to identify users using a tuple containing the MAC address of the device, the name that identifies the device during connection and the bond between users. The schema of the tuple can be seen below:

(MAC, DEVICE NAME, BOND)

The MAC address was chosen because it is a unique identifier, which will avoid repetition of information associated with a particular mobile device. As the MAC address is not simple information to the user decorating, we also chose to store and use it along with the device name as information to be displayed on the screen of middleware.

The bond is nothing more than the degree of affinity between two individuals. It works the same way as the concept of circles used in the social network Google Plus[4]. We decided to use this trick because we noticed that the middleware solutions for MSN there were no reference to a management mechanism affinity allowing the user to classify a person according to the degree of affinity. Thus, the user can specify whether a person belongs to the family circle, circle of friends, circle of colleagues or circle of acquaintances. Based on this classification, the user can, for example, share data only with individuals of a given circle.

The information contained in the tuple will be persisted to the mobile device through files. The use of files justifies our choice to keep only three pieces of information, since we know the restrictions of mobile storage. It is also important to note that the act of communication these files will not be transmitted. The information contained therein will be obtained from your own Wi-Fi Direct. The operation of the privacy mechanism My-Direct can be better understood through the following steps:

a. The device D1 is within reach of D2. So at any given time D1 calls D2 to the association.
b. From Wi-Fi Direct, the device D2 gets the MAC and name of D1. With the MAC, D2 checks on your file if there is some input tuples related to this address.
c. How is the first time that D1 finds D2, a new line in the tuple is added. This line is filled with the MAC, the name of D1 and the bond of friendship. For the

[4] Google Plus. http://plus.google.com

bond, the middleware asks through a dialog box to D2 which friendship circle D1 belongs.

d. Steps b and c also happen to D1. Thus, D1 can classify D2 as coworker, D2 can embed D1 in your circle of friends. Thus, it is explicit that the characterization of bond is private, so neither D1 or D2 know how to were classified.

e. Thereafter social activities can be performed.

f. After the end of the communication, if the devices re-find, the information already stored locally will expedite the process of association.

The privacy layer will be responsible for managing the tuples of user data in order to use them as input for mechanism privacy. Thus, with these data the privacy mechanism determines, for example, if a device found in the neighborhood belongs to a friend of the user.

4) Modules Layer

Another goal of My-Direct is to facilitate the development of MSN. For this, the layer modules can be used by developers to introduce new features to the My-Direct. This will allow developers to create different types of MSN.

a) Intelligent Energy Module

In studies such as [16] and [17], it is possible to observe the energy cost of each feature of a mobile device. The application of energy efficiency in mobile devices is very important to their use. Thus, the management of energy consumption can give the user a long period of use of your phone. Thus, the first module developed for the My-Direct is the Intelligent Energy Module (IEM). This module has the responsibility to manage resources of the mobile device as wireless networks and the brightness of the screen, so that the battery usage is optimized.

The IEM was designed to make use of intelligent agents. The multi-agent approach was chosen because agents are autonomous, that is, have the ability to dynamically react to events occurring in the environment in which they were entered. Furthermore, the use of multi-agent in this context allows, for example, every available resource on the mobile device can be controlled by an agent, that allows the implementation of different algorithms for each agent, aiming at optimal solution to the problem.

The agents of IEM are responsible for the execution of pre-defined actions aimed at optimizing the use of the battery. These agents are being implemented through an API called Andromeda (ANDROid eMbeddED Agent platform)[5].

The Andromeda is a platform developed specifically to provide the use of agents with the Android operating system. The agents developed by the Andromeda acts as services in Android. For the specification of behavior and task to be performed by the agent are used Java classes.

The Fig. 2 shows an excerpt of the code of an agent of IEM implemented by Andromeda. It is possible to see that the agent has a behavior (Behaviour class) called "Monitoring". Besides behavior, is also defined capacity (Capability class) named "Turn Off 3G", which will be the trigger for the task associated with it. The trigger fires when the condition contained in the expression method implemented in the Condition class is met. In this case, the action is to turn off the 3G network.

```
public class NetworkAgent extends Agent {

    @Override
    public void init() {

        Behaviour behaviour = new Behaviour("Monitoring");

        Capability mCapac1 = new Capability("Turn Off 3G",
                "intelligentenergysaving.TurnOff3g", new Condition() {

                    @Override
                    public boolean expression(Intent arg0) {
                        if (arg0.getAction().equals(
                                "intelligentenergysaving.TurnOff3g")) {
                            if (wifiIsConnected()) {
                                return true;
                            } else return false;
                        } else return false;
                    }
                });
        mCapac1.addTaskRun(task.class);
        behaviour.add(mCapac1);
```

Fig. 2. Excerpt of the code of an agent of IEM.

The IEM has two agents. One for the management of wireless networks (NetworkAgent) and another for managing the display (DisplayAgent). When the My-Direct is active, the DisplayAgent will periodically check the battery level of the mobile device and the current set the screen brightness according to this information. Meanwhile, the NetworkAgent will turn off or on the wireless networks, with the exception of Wi-Fi Direct, according to pre-established rules, to minimize battery consumption.

B. Example of Use

When the My-Direct get ready, developers will be able to build social P2P applications such as chat and photo sharing, in addition to can extend it through modules. Thus, in a simple chat application developed with My-Direct, for example, the user will be able to use the GUI (interface layer) to see your list of partners, the bond with them, and which theirs is available to connect. If some partner has not yet been classified, the user will have the option to add the bond (privacy layer). When choosing the partner, the communication layer, through Wi-Fi Direct, will go to associate the mobile devices of users. Thereafter, the chat can be performed.

[5] Andromeda. http://www.gti-ia.upv.es/sma/tools/Andromeda/

IV. Related Works

Although middleware for MSN is a new area of study, there are several works in order to resolve issues related to this topic. Each approached the problem from a point of view, however, some of them has similar characteristics.

In [1] it is possible to find some of these solutions. Among them, the most common type of architecture is P2P as in [9], [10], [11], [12], [14] and [15], but there are also those that use centralized architecture as [11] or hybrid as [13]. With regard to network technology used for communication between nodes, the most cited is the Bluetooth. However, there are also works that make use of Wi-Fi (infrastructure mode).

In mobile environment, the user wants to access information anytime and anywhere. Thus, the construction of middleware with a centralized architecture like [11] is not appropriate. The best option for middleware architecture aimed at mobile environment is P2P [1], where there is not centralized control and communication is done directly between devices.

The decentralized architecture requires network technologies that support interaction P2P. Most of the aforementioned studies, with the exception of [11], make use of Bluetooth for communication between nodes of mobile social network. However, the use of Bluetooth requires that the devices are relatively close so that the communication occurs. Furthermore, transmission rate of this technology is limited, which influences the performance of the middleware.

MMSN also deal with sensitive data such as social relationships, activities, and user preferences that can be used to infer other confidential information about the user over time. Thus, as in [12], [13], [15] and [2], a MMSN must manage user data and possess adequate control policies on the exchange of such information, to ensure the user's privacy. However, none of these works allows the user to have the option to sort the people you want to relate according to the degree of affinity.

It is important to note that a MMSN must also be able to minimize battery consumption during its execution. Among the solutions found in the literature only in [11] has this feature.

From the analysis of these works can be seen that the mobile environment requires a decentralized architecture. So to connect the nodes of the network is necessary to use a type of wireless network, independent of central infrastructure, which allows for a fast and efficient communication between nodes of the social network. During this process of communication the battery consumption should also be considered and a solution for energy saving becomes necessary. Furthermore, it is also clear that a user may want to classify one or more persons that relates in a different way, as a friend and colleague for example.

Therefore, to correct these deficiencies, this paper proposes the My-Direct a middleware architecture that will make use of the Wi-Fi Direct and network solution for the relationship between us and a privacy mechanism that will allow the user to sort their peers according to the degree of affinity. My-Direct also allow the use of modules, which may be developed to meet the specific functionality such as managing energy consumption.

V. Conclusion

MSN is a new and rapidly evolving field, and has attracted the attention of both the market and the academy. This makes appear several middleware solutions with different characteristics. Among the main features these solutions are: privacy of users' data, inference unknown social standards, application development and integration with social networking sites.

The middleware solutions in the literature show that there is an approach which satisfies all the requirements that the development of an MSN requires. Furthermore, within this set of approaches each author tries to solve one or more problems, present in a mobile social network environment, using various techniques, methods and tools. Given this middleware solutions heterogeneity, this paper proposed a new architecture called My-Direct. This solution aims to bring new resources to the area middleware for MSN as the introduction of the use of Wi-Fi Direct for P2P communication between mobile devices, a privacy mechanism that will help users to establish social relationships with people of interest and a module for energy saving.

At the end of the design and implementation of the My-Direct, we plan to provide a tool to assist developers in the process of building social networks P2P.

References

[1] A. Karan and N. Mohamed, "Middleware for Mobile Social Networks: A Survey," System Science (HICSS), 2012 45th Hawaii International Conference, pp.1482-1490, Jan. 2012.

[2] M. Holanda, A. Araújo, A. Queiroz, and J. de Castro; "MobiSN: A framework for developing mobile social networks in MANET". Information Systems and Technologies (CISTI), 2012 7th Iberian Conference, pp.1-5, June 2012.

[3] C. Borcea, A. Gupta, and A. Kalra. “The mobisoc middleware for mobile social computing: challenges, design, and early experiences”, 2008.

[4] WI-FI Alliance; “Wi-Fi Certified Wi-Fi Direct, Personal, portable Wi-Fi technology”. 2010. Available: http://www.wi-fi.org/knowledge-center/white-papers/wi-fi-certified-wi-fi-direct%E2%84%A2-personal-portable-wi-fi%C2%AE-technology-2010.

[5] H. Nguyen, “Mobile Social Network and Its Open Research Problems”. Available: http://www2.cs.uh.edu/~rzheng/course/COSC7388sp11/MSN.pdf, 2010.

[6] D. B. A. Santos, “Infraestrutura para o Desenvolvimento de Aplicações Pervasivas Cientes de Redes Sociais”. Dissertação de Mestrado. Universidade Federal de Campina Grande. Centro de Engenharia Elétrica e Informática. Campina Grande. Maio de 2011.

[7] G. Castelli, A. Rosi, and F. Zambonelli, "Design and implementation of a socially-enhanced pervasive middleware," Pervasive Computing and

Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference, pp.137-142, March 2012.

[8] M. Endler, A. Skyrme, D. Schuster, and T. Springer, "Defining situated social context for pervasive social computing" *in Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference*, pp. 519 –524, March 2011.

[9] S. Kern, P. Braun, and W. Rossak, "MobiSoft: An Agent-Based Middleware for Social-Mobile Applications," *in On the Move to Meaningful Internet Systems: OTM 2006 Workshops*, vol. 4277, pp. 984-993, Springer Berlin/Heidelberg, 2006.

[10] D. Bottazzi, R. Montanari, and A. Toninelli, "Context-Aware Middleware for Anytime, Anywhere Social Networks," *in IEEE Intelligent Systems*, Vol. 22, pp.23-32, 2007.

[11] A. Gupta, A. Kalra, D. Boston, and C. Borcea, "MobiSoC: A Middleware for Mobile Social Computing Applications," *Mobile Networks and Applications*, Vol. 14, pp. 35-52, 2009.

[12] 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," *in Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, pp. 135-146, March 2008.

[13] D. N. Kalofonos and Z. Antoniou, "A Hybrid P2P/Infrastructure Platform for Personal and Social Internet Services". *Proceedings of the 19th IEEE Personal, Indoor, and Mobile Radio Communications Symposium - Applications, Services & Business Track (PIMRC '08)*, Cannes, France, pp.1-6, September 2008.

[14] A. Pietilainen, E. Oliver, J. Lebrun, G. Varghese, and C. Diot, "MobiClique: Middleware for Mobile Social Networking," *in Proceedings of the 2nd ACM Workshop on Online Social Networks*, Barcelona, Spain, pp. 49-54, 2009.

[15] M. Basuga, R. Belavic, A. Slipcevic, V. Podobnik, A. Petric, and I. Lovrek, "The MAgNet: Agent-based Middleware Enabling Social Networking for Mobile Users," *in Proc. 10th International Conference on Telecommunications (ConTEL)*, pp. 89-96, 2009.

[16] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," *Proceedings of the 2010 USENIX Conference*, 2010.

[17] G. P. Perrucci, F. H. P. Fitzek, and J. Widmer, "Survey on Energy Consumption Entities on the Smartphone Platform," *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, pp. 1–6, May 2011.