MY-DIRECT: A MIDDLEWARE FOR P2P MOBILE SOCIAL NETWORKS

Luiz Marcus Monteiro de Almeida Santos[1] and Admilson de Ribamar Lima Ribeiro[1]

[1]Department of Computer Science, Federal University of Sergipe, São Cristovão, Brazil

ABSTRACT

In recent years, middleware for mobile social network 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 API so that developers can access this information quickly and build new social applications. With a view to supporting this new trend of research, this paper presents a middleware for mobile social networking called My-Direct which makes use of Wi-Fi Direct technology together with the Bluetooth, aiming to provide flexible communication between the nodes of the mobile social network. Along with My-Direct also was created a mechanism for user privacy based on information available on his mobile device.

KEYWORDS

Peer-to-peer, Middleware, Mobile Social Network, Mobile Device

1. INTRODUCTION

The growing interest in social networks and smartphones has resulted in a new trend called MSN (Mobile Social Network). The MSN is a subclass of social networks, where the users make use of mobile devices to access your social networks. Thus, a user, from your mobile device, can read, publish, and share contents, aiming at their social relations [1].

Are considered MSN, social networks that have been developed for the Web but offer mobile applications for access to content such as Facebook[1] and Twitter[2], and social networks that were designed, from the beginning, targeting the mobile environment such as Google Latitude[3], Path[4], Instagram[5].

Aiming to support to the process of development of MSN, the middleware for MSN (MMSN) emerged. This kind of middleware has attracted the attention of academia, leading the design and development of various approaches of researchers such as [2], [3], [4] and [5].

[1] http://www.facebook.com

[2] http://www.twitter.com

[3] http://www.google.com/mobile/latitude

[4] http://www.path.com

[5] http://www.instagram.com

DOI : 10.5121/ijcnc.2014.6314

A MMSN should consider the limitations of mobile devices such as limited power, low memory capacity, limited processing power, scalability and heterogeneity [2]. Furthermore, a MMSN should provide a layer that provides common services needed by different MSN applications and separate social network management concerns from application requirements [6]. The middleware should enable efficient operations of MSN applications by being self-configuring, self-adapting, self-optimizing and self-protecting.

Being MMSN a recent topic of research, many of the problems mentioned above don't yet have an ideal solution. Thus, current middleware solutions are incomplete, and doesn't exist a perfect middleware infrastructure that meets all requirements [7].

Within this context, MMSN solutions that are ad hoc have presented problems related to loss of connectivity between the nodes of the social network, since Bluetooth does not support large distances and transmits at low speed. Furthermore, middleware with centralized architecture that make use of technologies such as Wi-Fi has the difficulty of requiring access point for the effective use of services.

In addition to issues involving communication, it is important to note that in a design of MMSN must also be taken into consideration what kind of privacy mechanism will be used. The specification of this type of mechanism is a complex task, since, in social networking projects, as higher the sociability offered more chances of having privacy issues [1]. The privacy mechanism must prevent leakage of information and at the same time, cannot restrict socialization of the user. Thus, keep in tune the relationship between socialization and privacy is the main difficulty in developing this type of mechanism.

In order to contribute to this new trend of research, in this paper, we present a MMSN called My-Direct that makes use of WI-Fi Direct [8] together with the Bluetooth, aiming to provide flexible communication between the nodes of the mobile social network. For that, the My-Direct performs the management of these two technologies, so that, when one their is not available, the other is used in your place and vice-versa. The change between the technologies is transparent to the user. The My-Direct also presents a mechanism for privacy to the user based on the information provided on your mobile device such as phone calls and text messages (SMS). From these data, the My-Direct builds a decision tree, which is used to classify each partner from one of the available pre-defined groups (Family, Friend, Work, Acquaintance and Other). In addition, each group of partners can be assigned a set of social activities. A social activity may be a chat, picture share or a network game, for example. Despite the classification of the partners be automatic, the allocation of social activities is free and can be made according to the needs of mobile social network being developed above of the My-Direct.

The remainder of the paper is organized as follows. Section 2 presents the requirements for a MMSN. Section 3 presents the middleware My-Direct. In Section 4 is described the case study. The evaluation of My-Direct is presented in Section 5. The Section 6 presents the related works. Finally, in Section 7 are the conclusions.

2. REQUIREMENTS FOR A MIDDLEWARE FOR MSN

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 [7], 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 suffer 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.

3. THE MIDDLEWARE 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, in the maintenance of social relationships between users, in data privacy and social communication between network devices.
One of the problems presented by the MMSN in the literature is the loss of connectivity in P2P architectures. A major cause of failures in connectivity solutions for decentralized middleware was associated with the distance between the mobile devices, as they make use of Bluetooth that has a limited range. Another problem encountered was that in centralized middleware solutions, there was always a need for association in access infrastructure such as Wi-Fi routers, for the user could access the MSN.

Thinking of solving these problems, this paper presents a MMSN, called My-Direct, where communication between nodes is accomplished by combining two ad hoc network technologies. One such technology is Bluetooth, since it is used by most of the MMSN. The other is the Wi-Fi Direct, a recent specification for P2P communication, little discussed in the literature and that still is little applied in the development of social network applications.

By providing support to two network technologies, the My-Direct becomes more flexible the communication between the nodes in the MSN, since the exchange of information between them can be performed through the network that is available. Thus, if a user of a social network built over the My-Direct, are performing a social activity with your partner via Bluetooth and there is a failure in connectivity, the service will continue, however, via Wi-Fi Direct.

The My-Direct also features a privacy mechanism for users, since according to [7], a MMSN should be responsible for managing user information and provide control policies for the exchange of such data. This mechanism is based on analysis of phone calls and text messages (SMS) contained in the user's mobile device, so that the extracted information is used to identify the degree of affinity for each partner.

3.1. Architecture

The architecture of the My-Direct was designed in a decentralized manner, having as main objective to support private P2P communication between mobile devices via Bluetooth and WI-Fi Direct. To achieve this goal, the architecture was divided into modules, as shown in Figure 1, where each is responsible for managing a part of the resources of the middleware. The modules that make up the My-Direct are: *Communication Module*, *Privacy Module*, *Persistence Module* and *API Module*.

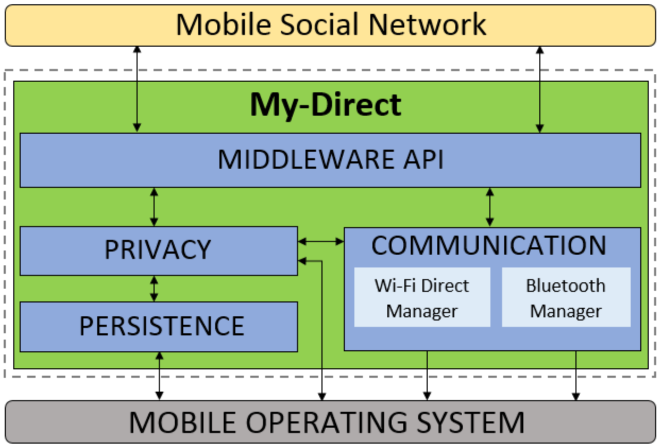


Figure 1. The My-Direct architecture

In Figure 1 also is possible to notice that the My-Direct is positioned between the mobile OS and the MSN. The adoption of this model allows it to have access to resources that a mobile OS generally provides as access to hardware and device data. Thus, when accessing these resources, the My-Direct makes monitoring and managing them in order to use them in internal routines and provide methods to be used in the development of MSN.

3.1.1. Communication Module

The Communication Module is responsible for detecting the nearby mobile devices, enabling the association and communication between the nodes of the social network. To perform these activities, this module makes use of artifices available by the OS in order to manipulate the network technologies, in this case, Bluetooth and Wi-Fi Direct.

For better coordination of network technologies provided by My-Direct, the management of these was divided between the sub-modules: *Wi-Fi Direct Manager* and *Bluetooth Manager*.

The *Wi-Fi Direct Manager* is responsible for accessing the features of Wi-Fi Direct on a mobile device in order to monitor the status that network. These statuses can vary between active or inactive, connected or disconnected, available or unavailable. This monitoring is also used to get the list of available partners. Thus, the *Wi-Fi Direct Manager* also is used to handle the Wi-Fi Direct, so that the user can establish a connection or close it, when a partner is nearby.

The *Bluetooth Manager* is the sub-module responsible for monitoring and operating resources related to Bluetooth technology. It works the same way that *Wi-Fi Direct Manager*, monitoring the status of the network and enabling communication between nodes via Bluetooth.

From the functionality provided by *Wi-Fi Direct Manager* and *Bluetooth Manager*, the Communication Module is able to know, for example, if Wi-Fi Direct is no longer available, and then immediately activate a Bluetooth connection, to maintain connectivity between nodes.

3.1.2. Privacy Module

Since the goal of My-Direct is to provide communication between nodes in an MSN in your project is necessary to take into account that the association between mobile devices should be performed if their owners possess some degree of affinity.

An alternative to checking the degree of affinity between users, would be the access to online social networking sites, as described [9]. However, in the case of My-Direct, the independence with respect to the access infrastructure does not allow obtaining this information from Web Services, for example. Thus, we chose to evaluate the affinity between users through information already contained in the mobile device. For this, the *Privacy Module* works in two phases: identification (1) and classification (2) of the user.

(1) User Identification

To identify users, the *Privacy Module* uses two information. The first information is the MAC address of the mobile device. The MAC was chosen because it is a unique identifier, which avoids repetition of information associated with a particular mobile device. Thus, My-Direct can compare the obtained MAC with the addresses already stored and evaluating if the device in the vicinity belongs to someone known.

The second information used to identify the user is the phone number. The phone number is used when the mobile device is not identified from the MAC. Thus, the *Privacy Module* accesses the agenda of the user's mobile device in order to find out if the incoming number belongs to some contact contained therein.

Importantly, the fact that the phone number to be a personal information requires care. Therefore, to be used in the identification phase, the phone number is encrypted. The encryption algorithm chosen for this purpose was the SHA-1 (Secure Hash Algorithm).

(2) User Classification

The goal of classification is to obtain the bond between the users who want to communicate. The bond is nothing more than the degree of relatedness between two individuals. To infer that bond, the *Privacy Module* analyzes the persisted information on the user's mobile device.

The model of classification of *Privacy Module* is based on [10]. This choice was made because it had already presented the labels for classification validated by real users, which are: Family, Friend, Work, Acquaintance and Other. Also, spite of on the literature, there are classification models with accuracy rates higher than 80%, as in [11] and [12], in these studies there is the use of data extracted of Facebook, which by the architecture of the My-Direct is not feasible. Thus, aiming to use a methodology for automatic grouping that suited the goals of the My-Direct, the model shown in [10] was adopted by owning accuracy rate above 70%, using only data of the device mobile.

Therefore, to classify users, the *Privacy Module* makes use of groups shown in [10] which are assigned according to the output of a decision tree. This decision tree is simple and uses as input data of the agenda associated with the partner nearby, such as the total number of calls and SMS, call duration and size of SMS.

3.1.3. Persistence Module

The *Persistence Module* is responsible for storing user's personal data and its partners in a DBMS. Among the data kept by this module are the degrees of affinity used to classify partners. These data are available for use by other modules of the My-Direct. Thus, the *Privacy Module* can, for example, makes use of *Persistence Module* in order to obtain information about a partner or persist the degree of affinity assigned to it.

It is important to emphasize that the *Persistence Module* also maintains records of functionalities developed on the My-Direct. This record is required for the storage of the association between these functionalities and the degrees of affinity. Thus, it becomes easier, for example, the search for functionalities that are allowed to a partner.

3.1.4 API Module

This module is designed to allow developers to take advantage of the features of My-Direct. Thus, this module provides an access to the My-Direct interface which can be used by developers to explore the features of this middleware.

3.2. Operation

Assuming that there is an MSN implemented on top of the My-Direct and two users are using the devices D1 and D2, which support Bluetooth and Wi-Fi Direct, the steps of operation of middleware are as follows:

a) When the My-Direct is installed on D1 and D2, it initializes its modules and stays at the disposal of MSN as an OS service (Figure 2).

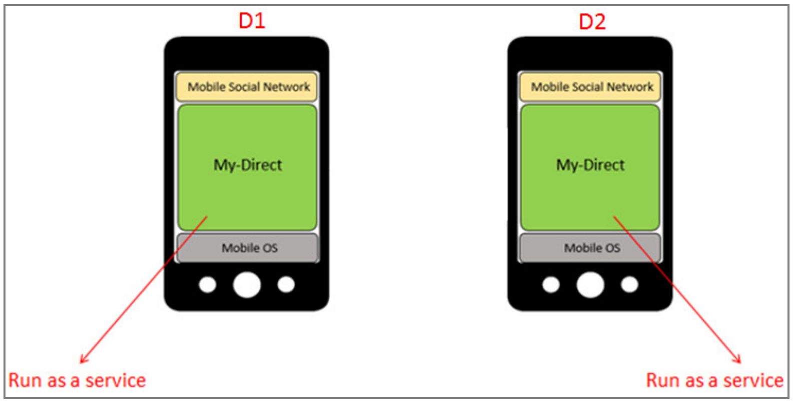


Figure 2. My-Direct running as an OS service

b) When MSN is installed and run in D1 and D2, the first procedure performed by it is joining the My-Direct, through the *Module API*, to access the resources available for it (Figure 3).

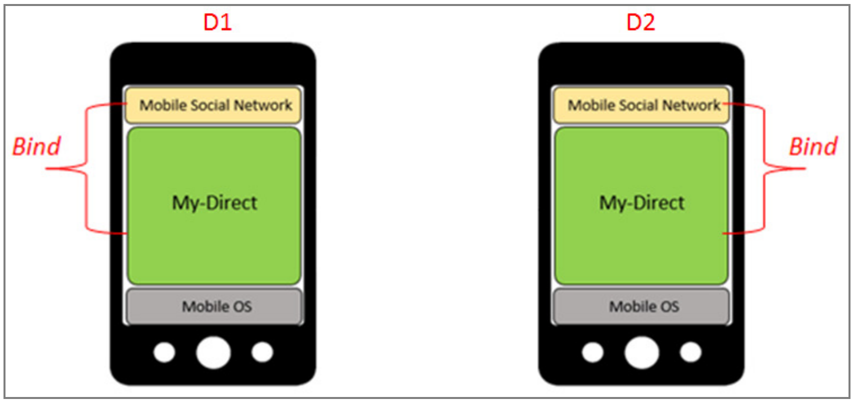


Figure 3. Association between MSN and My-Direct

c) With the conclusion of the association between the My-Direct and the MSN, the second procedure is to create a profile for the user. The phone number, email and status are entered by the user in the MSN and stored in the My -Direct. The user's name and photo are obtained automatically by the middleware through a search in the internal information of the device. Also, the phone number, which has been encrypted to be persisted, is combined with the user's name to form the device name for the Bluetooth and Wi-Fi Direct connections, presenting a similar format to this: *John/3829a748*. This name identifies the mobile device when it is found in another neighborhood (Figure 4).

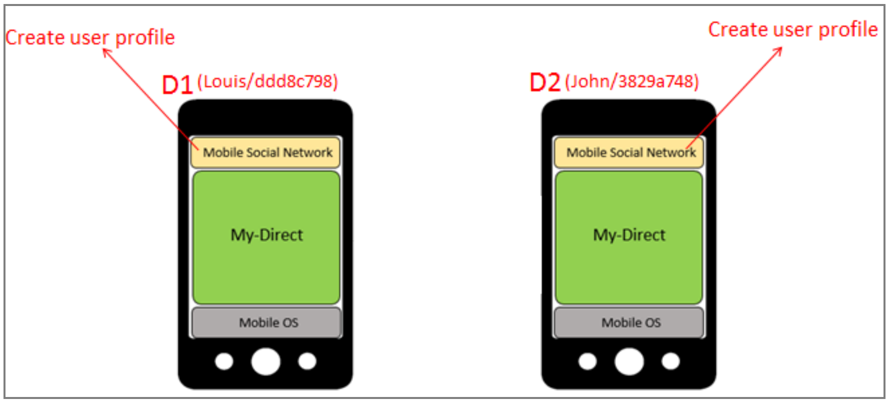


Figure 4. Creation of user profile and device name that identifies the user in the neighborhood

d) Assuming that D1 and D2 are only with the Bluetooth active, when device D1 is within reach of D2, in this moment, the *Communication Module* of the My-Direct present in D1 gets the MAC. Besides the MAC, also is acquired the encrypted number phone of D2. This number is obtained by reading the name of the device D2. The process of obtaining the MAC and the device name of the partner can be seen in Figure 5.

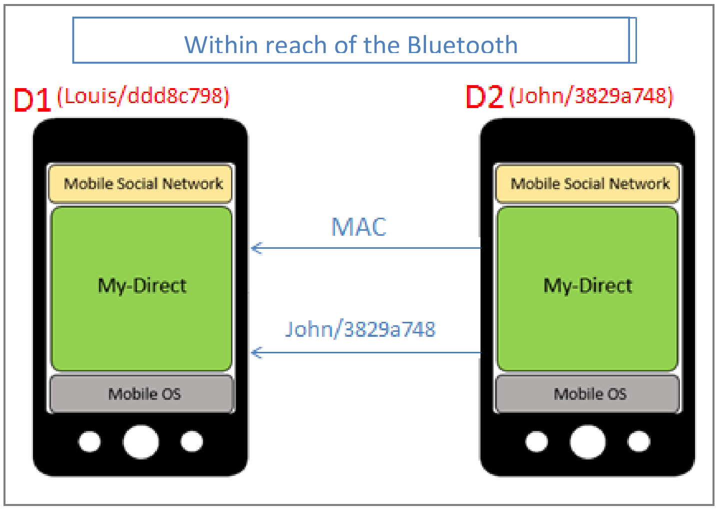


Figure 5. Device of user getting the MAC and the device name of the partner

e) From the collection of this information, the My-Direct fires the *Privacy Module*, in order to find out if the owner of the device D2 is known.

f) The *Privacy Module* makes a research on data maintained by the *Persistence Module*, with the goal of finding a partner related to the MAC obtained. As this is the first time that D1 finds D2, the registry is not found (Figure 6).

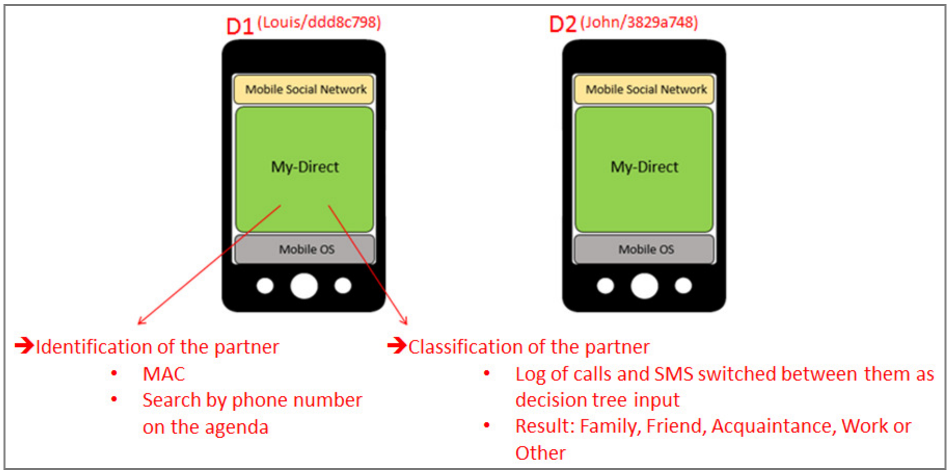

D1 (Louis/ddd8c798)
D2 (John/3829a748)
Mobile Social Network
My-Direct
Mobile OS
Mobile Social Network
My-Direct
Mobile OS
→Identification of the partner
• MAC
• Search by phone number on the agenda
→Classification of the partner
• Log of calls and SMS switched between them as decision tree input
• Result: Family, Friend, Acquaintance, Work or Other

Figure 6. My-Direct running the process of identification and classification of the partner

g) As the related MAC record is not found, the *Privacy Module* tries to find the partner in the agenda of D1 through of the encrypted phone number of D2, as shown in Figure 6. If the contact of the D2 is found on the agenda of D1, the *Privacy Module* uses the log of calls and SMS exchanged among them as a decision tree entry. From the output of the tree, the partner is bound to one of the groups (Family, Friend, Acquaintance, Work, Other). If the contact of D2 is not found, the partner will be linked to the group "Other".

h) With partner already classified, the My-Direct creates a new record for the partner and stores it via *Persistence Module*. This record contains the name, the encrypted phone and the group it belongs to the partner.

i) After identifying and classifying the partner, the device D1 can connect to D2. Note that steps 3, 4, 5 and 6 are also conducted in D2. Thus, the owner of D1 is also identified and classified, and this classification may be assigned to the same or different from the owner of D2.

j) As the two devices are within range of the Bluetooth this network is used for the connection. The connection process may be implemented by the developer to be triggered by a button, for example. If the devices are not paired, the communication module makes pairing, requiring only confirmation from users (Figure 7).

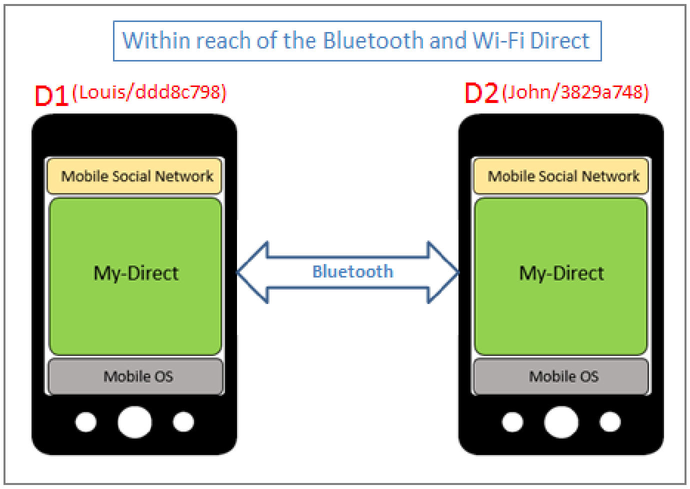


Figure 7. Execution of the Bluetooth connection between the devices

k) With D1 and D2 already connected, users can perform social activities provided by MSN as chat and sharing photos, for example (Figure 8). The use of these activities is released by the Privacy module according to the group to which the partner belongs. Importantly, it is up to the developer to implement the MSN social activities, and the task of the My-Direct only grant funds for the building and management of them.

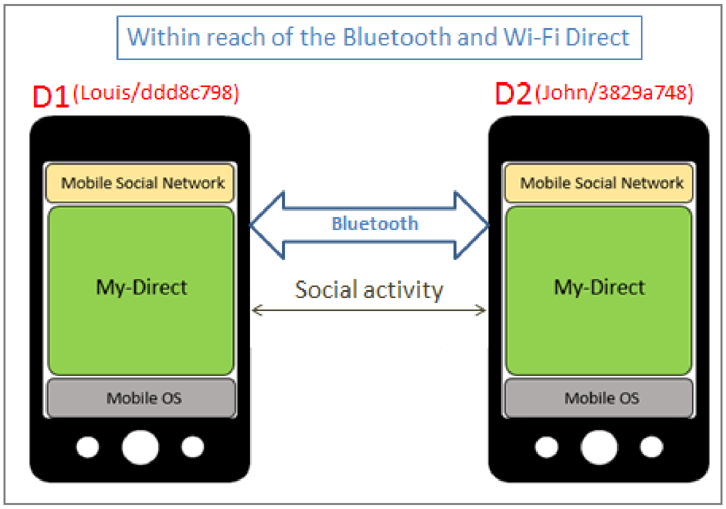


Figure 8. Devices running the social activity via Bluetooth

l) If during the execution of the chat, D1 and D2 stay away a distance not supported by Bluetooth, for example, the *Communication Module* will replace it by Wi-Fi Direct, assuming that it is active (Figure 9). If the Wi-Fi Direct is not connected, the connection will be required. If already exist a Wi-Fi Direct connection between the devices, the activity continues normally.

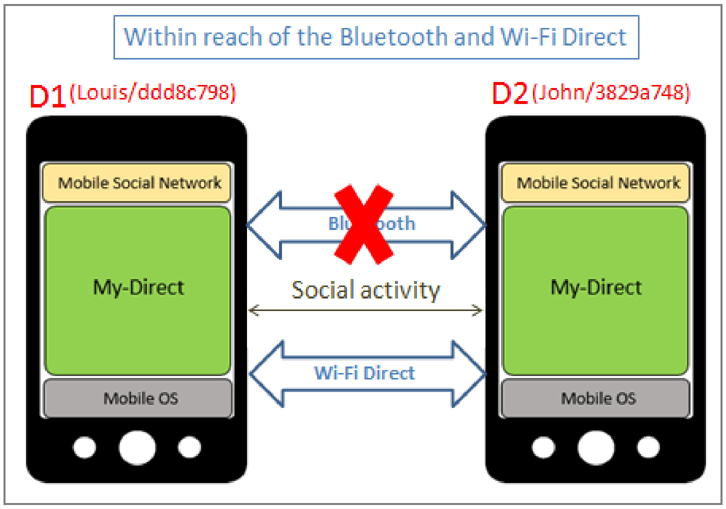


Figure 9. Failure in Bluetooth connection and creation of a Wi-Fi Direct connection by My-Direct to maintain the execution of social activity

m) At the end of the execution of the chat, D1 can perform another activity or request the disconnection to D2. This disconnection is only, in other words, regardless of which network technology is being used at the time, the disconnection will occur.

An important observation regarding the operation of My-Direct is that if the two devices are within the scope of Bluetooth and Wi-Fi Direct, priority is given to Bluetooth, since, according to [13], this technology consumes less battery. Additionally, the My-Direct only supports connections one by one. It can not run applications with multiple users simultaneously.

Another relevant consideration is that, although the My-Direct automatically assigns a group to the partner, this classification can be changed by the user.

3.3. Implementation

The My-Direct was implemented in a P2P architecture targeted the Android platform [14]. The fact that he has as target the Android platform, implies the choice of a minimal version of operating system support. For this, we chose to develop the My-Direct to version 4.0 of Android, since the support and the manipulation of the capabilities of Wi-Fi Direct are present in your SDK. It is important to remember that the development of applications for the Android platform is done by the Java programming language.

The My-Direct was implemented to execute as a service of the Android OS. Thus, it performs all its tasks in the background. For an MSN has access to the resources of My-Direct, it is necessary to perform a join operation known as *bind*. After of this association, the relationship between MSN and the My-Direct is similar to that of a client-server architecture, where the client makes requests and the server sends replies.

The *Communication Module* has been implemented through of Bluetooth and Wi-Fi Direct APIs, which are provided by the Android SDK. From these API were developed Java classes responsible for the association and effective communication between devices. To monitor the status of Bluetooth and Wi-Fi Direct in the user's mobile device, these classes make use of a component of the Android SDK called BroadcastReceiver [14]. This component serves to event management. Thus, if Bluetooth is disabled, for example, the OS sends a broadcast message, which can be captured through a BroadcastReceiver. When it gets the message, it can implement an answer to it. Besides tracing messages related to Bluetooth and Wi-Fi Direct, the *Communication Module*, to receive them, generates its own messages and sends in broadcast, in order to warn other components of the My-Direct about events. Some these messages can be seen in Table 1.

Table 1. Some of the messages used by *Communication Module.*

Message	Description
mydirect.contactFound	Sent when a contact is found.
mydirect.wifiDirectDisabled	Sent when the Wi-Fi is disabled.
mydirect.wifiDirectEnabled	Sent when the Wi-Fi is activated.
mydirect.bluetoothDisabled	Sent when the *Bluetooth* is disabled.
mydirect.bluetoothEnabled	Sent when the *Bluetooth* is activated.

The *Privacy Module* consists of Java classes that use methods of *Persistence Module* and *Communication Module* to identify the partner through the MAC of the device found in the neighborhood. Moreover, these classes also use the Android SDK to access the user's agenda in order to find the contact whose phone number is equal to that received by My-Direct during the identification phase. For classify the partners found during the identification phase, the classes of *Privacy Module* makes use of a decision tree. The decision tree was built from a survey conducted with 15 volunteers, where they provided the logs of calls and SMS from their mobile devices. These data were collected from an Android application built exclusively for this purpose. After being installed and started, the Android application showed to the volunteer of the research a list of up to 50 contacts with whom he communicated at least once. These contacts were chosen randomly in the agenda. Each contact could be ranked among one of the groups shown in [10]. After classifying all the contacts listed, the volunteer sending of data calls and SMS for each contact rated to a database. Through the Android application were classified 439 contacts, including 133 as "Family", 174 as "Friend", 39 as " Acquaintance ", 78 as "Work" and 15 as "Others". Associated with these contacts, were obtained 2899 calls and 1242 SMS. The attributes utilized for training the decision tree can be seen in Table 2. To generate the decision tree, the *Privacy Module* uses the algorithm J48 of the Weka [15] API.

Table 2. Description of the attributes used in the construction of the decision tree.

Attribute	Description
Ratio between SMS and calls	The ratio between exchanges SMS and calls with a contact in the whole communication history.
Number of calls	Total calls made between the volunteer and specific contact
Number of SMS	Total SMS made between the volunteer and specific contact
Average SMS length	Average size of SMS exchanged with a contact. The size of SMS was accounted from the amount of characters.
Average Call duration	Average duration in seconds for calls made between the volunteer and a specific contact.
Hour	The time at which was made the most of the communication between the volunteer and your contact. The values of this attribute ranged from: Dawn, Morning, Afternoon and Night.

Weekday	The day on which was made the most of the communication between the volunteer and his contact. The values of this attribute ranged from: Weekday and Weekend.

The *Persistence Module* was implemented using Java classes that make access to the database tables to handle and store or retrieve information about the user and your partners. For this, the database used is SQLite [14]. This database was chosen because come already embedded in the Android OS, which avoids problems of installation or configuration. Moreover, the manipulation of SQLite features is done by own Android SDK.

Lastly, the *API Module* was developed to the paradigm of communication between processes, the IPC (Inter-Process Communication). Thus, when the MSN makes a request, objects of complex types and the primitive types contained in the response may be transmitted of the My-Direct to MSN. For the definition of the My-Direct API interfaces was used the AIDL (Android Interface Definition Language) [14]. This language is provided by Android for specifying interfaces for the IPC protocol. From the definitions of interfaces made in AIDL, is automatically generated by the IDE the code required for remote communication between the client (MSN) and remote service (My-Direct). The Figure 10 shows the AIDL interface of My-Direct API.

```
interface IMyDirectService{

       void connect(in Contact contact);
       void disconnect(in Contact contact);
       void discovery();
       void addContactToPhone(String name, String number, String mail);
       String getOut();
       void send(in Contact contact, int code, String content);
       void finishSocketConnection();
       boolean isConnected();
       java.util.List<Contact> getContactFound();
       String getContactPhoto(String number);
       Profile getOwnerProfile();
       Profile setOwnerProfile(String phone, String email, String status);
       String getOwnerName();
       String getOwnerPhoto();
       boolean updateOwnerProfile(String id, String name, String phone,
String email, String status);
       java.util.List<Contact> getContacts();
       boolean deleteContact(int id);
       boolean updateContact(int id, String name, String phone, int
idbond);
       Functionality registerFunctionality(String name);
       void linkFunctionalityToBond(int id, int idbond);
       boolean updateFunctionalitiesToBonds(in Functionality
functionality);
       java.util.List<Functionality> getBondApplications(int idbond);
       java.util.List<Functionality> getAllFunctionalities();
       int[] getPartnerFuncionalities();

}
```

Figure 10. My-Direct API interface

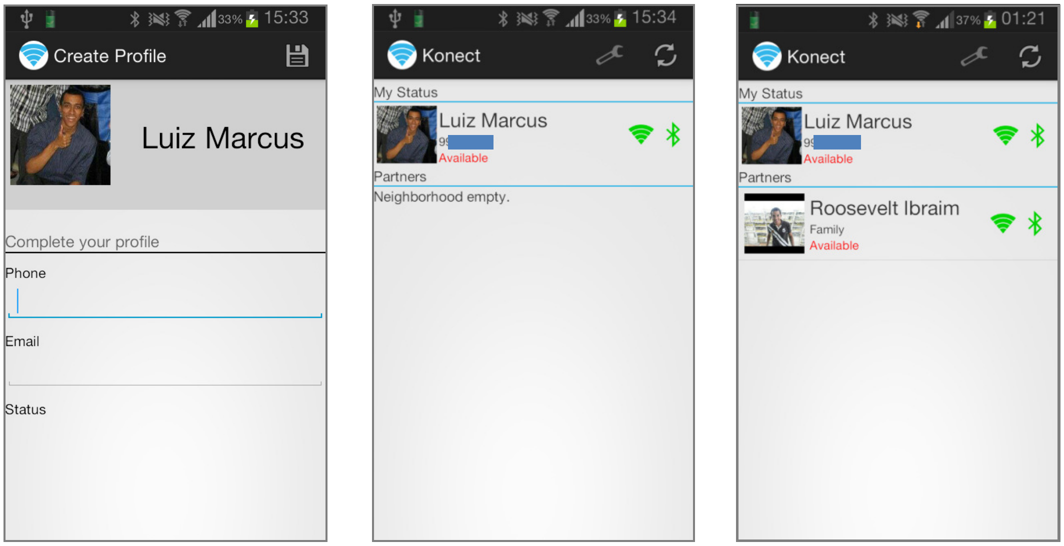
4. CASE STUDY

For evaluation of My-Direct, we decide to create a P2P MSN in its top. An MSN built through the use of resources of the My-Direct was named Konect. The Konect is an Android application and aims to provide social interaction among people who are in the same place. Thus if two people who do not know are in a shopping mall, for example, they can make use of Konect to create a new bond of affinity, through conversations via chat and exchange photos.

The Konect was implemented in Java and XML (eXtensible Markup Language) through the IDE (Integrated Development Environment) Eclipse. The Java language was used in the development of Activities. The Activities are Java classes where are implemented screens Konect application and how the user interacts with it. Are these Activities that are triggers the My-Direct routines according to the features accessed by the user. In turn, the XML was employed in the positioning of elements such as menus, buttons and images on the application screen. Importantly, the Konect works on mobile devices running Android 4.0 or higher.

When Konect is first executed in the user device, the screen shown in Figure 11a appears. This screen is displayed when the Konect through the *Persistence Module* of the My-Direct, verifies that the user does not have a registered profile.

After the data be saved, the Konect presents to the user its main screen named "Konect" as shown the Figure 11b. This screen shows the user photo and data such as name and phone number (covered by a blue bar on image for privacy). Furthermore, this screen is also possible to see the user's status in relation to connecting to some partner. The status can vary from "Available" (while awaiting future connections) and "Connected" (when it is associated to any partner).
Also in Figure 11b, it is possible to note the presence of two icons of the right information to the user. These icons are used to identify which network the user is active while he is using Konect. The first icon represents the Wi-Fi Direct, so that if this network is active the icon turns green, otherwise the color changes to gray. The second icon is the Bluetooth and works the same way as the Wi-Fi Direct. Thus, the network that is active will be used in the search for partners. If both are active, the search is made through the two.



Luiz Marcus
Complete your profile
Phone
Email
Status

My Status
Luiz Marcus
[redacted]
Available
Partners
Neighborhood empty.

My Status
Luiz Marcus
[redacted]
Available
Partners
Roosevelt Ibraim
Family
Available

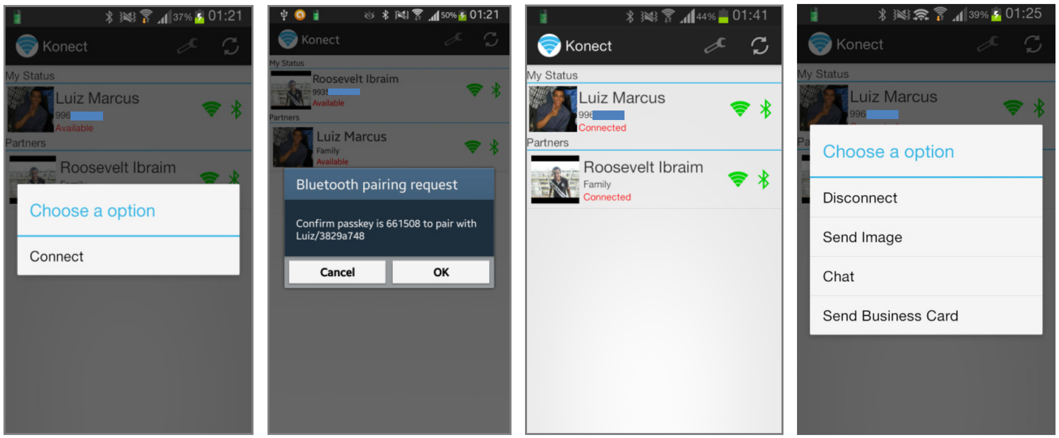
a) b) c)

Figure 11. Main screen of Konect

When the search for partners is initiated and some mobile device of vicinity is found, the Konect uses the *Privacy Module* of My-Direct for identifying the owner and classify him. After this process, the partner’s information (name, photo, degree of affinity and status) are shown in the main screen as in Figure 11c.

In Figure 11c it is also possible to note the existence of two icons right side to the partner information. These icons are used to identify which network the partner is currently using. If the partner is using only the Wi-Fi Direct, only the corresponding symbol is green, being the same principle used for Bluetooth. If both networks are active, both symbols become green. If the partner disable Bluetooth, for example, the icon for this technology is grayed out, the same goes for Wi-Fi Direct. Thus, in the case of Figure 6, the partner was only with Wi-Fi Direct and Bluetooth active.

When the user selects the partner with whom he wants to connect, it displays a dialog box and in it there is the option "Connect" (Figure 12a). Clicking on this option, the My-Direct starts the process of connecting through the network that is active in the partner. Reminding that in the case of the two technologies are active, the priority is given to Bluetooth. During the connection process it is necessary that the partner confirm the connection request from the user (Figure 12b). When the process is completed both the user and the status of the partner change to "Connected" (Figure 12c).



Confirm passkey is 661508 to pair with Luiz/3829a748

a) b) c) d)

Figure 12. Connection process on Konect

After the connection is made the user can interact with the partner. For this, he selects the partner and, shortly thereafter, appear the activities that can be performed (Figure 12d).

In the case of Figure 12d, according to the preferences of the partner, the user can activate the chat, send a picture or send a business card with your personal information. These social activities are implemented in Konect, which register at My-Direct each activity created and which stakeholder groups can use them. Furthermore, it is important to remember that the sending and receiving of information, the Konect makes use of the *Communication Module* of My-Direct, which ensures that if the connection fails, the network technology is replaced by another that is active and activity will continue to be executed. For the user disconnects the partner, he uses the "Disconnect" option in the menu of social activities.

5. Performance Evaluation

To perform the evaluation of performance was measured in a systematic way the time that the My-Direct led to: (i) find a partner in the neighborhood, (ii) establish a connection between the user and your partner, (iii) send an image of user to your partner. Furthermore, it was also evaluated the percentage of correctness of the decision tree used by the *Privacy Module*.

In the experiments, for each parameter variation on the performance evaluation, each test was performed 10 times to permit an analysis of the times of each run and calculate the average time most faithful. This time was measured by employing the use of static method *currentTimeMillis* of *java.lang.System* class, which was inserted into the Konect source code.

In order to evaluate the performance of My-Direct, to ensure greater credibility to the results, were used real smartphones instead of emulators. For this, two smartphones were chosen: the Samsung Galaxy S2 (GT-I9100) and Samsung Galaxy S3 (GT-I9300). The Galaxy S2 had version 4.1.3 of Android installed, while the Galaxy S3 had the 4.3 version of the same OS. Both also had support both Bluetooth as the Wi-Fi Direct.

5.1. Experiment: Search for Partners

The first performance test used a scenario where two users, A and B, were at a distance of 3 meters from each other. From there, the user A, making use of Konect on your mobile device, began the search for a partner. In the results of this test were taken into account the time interval in milliseconds (ms) between the beginning of the search and return on partner found at My-Direct. Furthermore, it is important to clarify that the partner found during the search was previously stored and classified, so the time measured in this experiment only refers to search and identification of the pair. The results of the experiment can be seen in Figure 13.

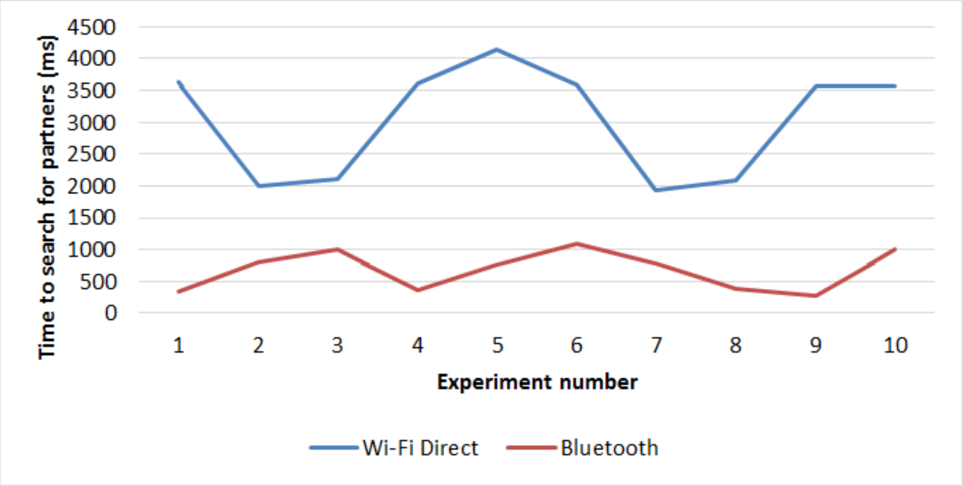


Figure 13. Results of the search for partners

In Figure 13, the X-axis indicates the number of test runs and the Y-axis indicates the time taken by My-Direct in the search for partners. In this figure, is possible to see that in the search for partners via Wi-Fi Direct, the My-Direct presented a expenditure of time greater than when Bluetooth was used. In that experiment, while for Wi-Fi Direct the average was 3022.4 ms in executing the search, for the Bluetooth the average was 682.1 ms for the same action.

5.2. Experiment: Establishing a Connection

In the second performance test was used a scenario where two users, A and B, were at a distance of 3 meters from each other. Moreover, in this test, the Konect already indicated to the user the existence of a partner B in your neighborhood. Thus, it was possible A to request the connection to B. In the results of this test were taken into account the time interval in milliseconds (ms) between the request and the actual connection between the user and your partner through My-

Direct. Also, connection requests sent to the partner took an average of 1000 ms to be confirmed once the process has been done manually. It is also important to note that in the case of Bluetooth, the measured time was of the pairing. The results can be seen in Figure 14.

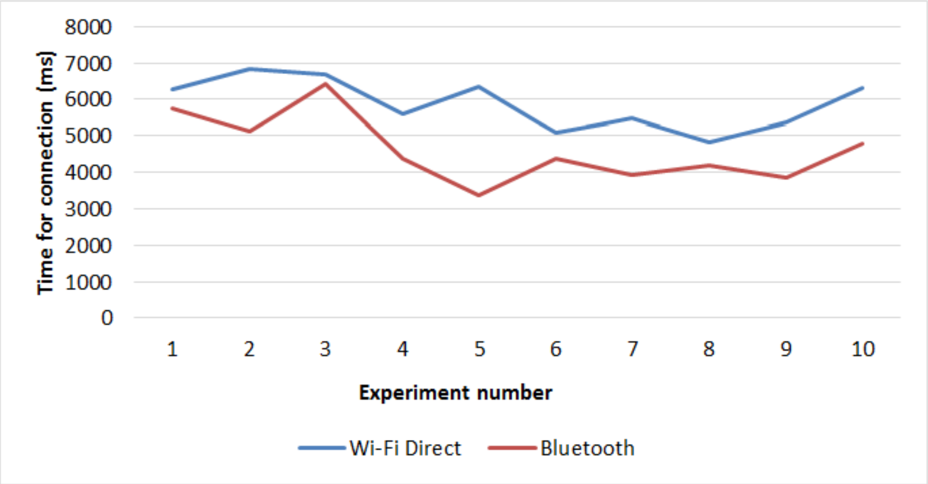


Figure 14. Results of the time for connection

In Figure 14, the X-axis indicates the number of test runs and the Y-axis indicates the time the user took to connect to your partner via My-Direct. By observing this figure, is possible to see that the My-Direct, when the Bluetooth was used for the connection, it was slightly faster than when made use of Wi-Fi Direct. This conclusion becomes more evident when the average time spent by each technology is analyzed, where for the Wi-Fi Direct was 5880.2 ms and the Bluetooth was 4619.7 ms, Thus, the mean difference between them was about 1260.5 ms.

5.3. Experiment: Sharing Image

The objective of this third performance test was to evaluate the performance of My-Direct regarding the transmission of data from the user's mobile device to your partner. For this, we constructed a scenario where two users, A and B, were at a distance of 3 meters from each other. In addition, in this test A was already connected to the B through the My-Direct. Thus, it was possible for users to run social activities together.

In the results of this test were taken into account the time interval in milliseconds (ms) sending a 1MB image in JPG format of the user's mobile device to your partner through My-Direct. These results can be seen in Figure 15.

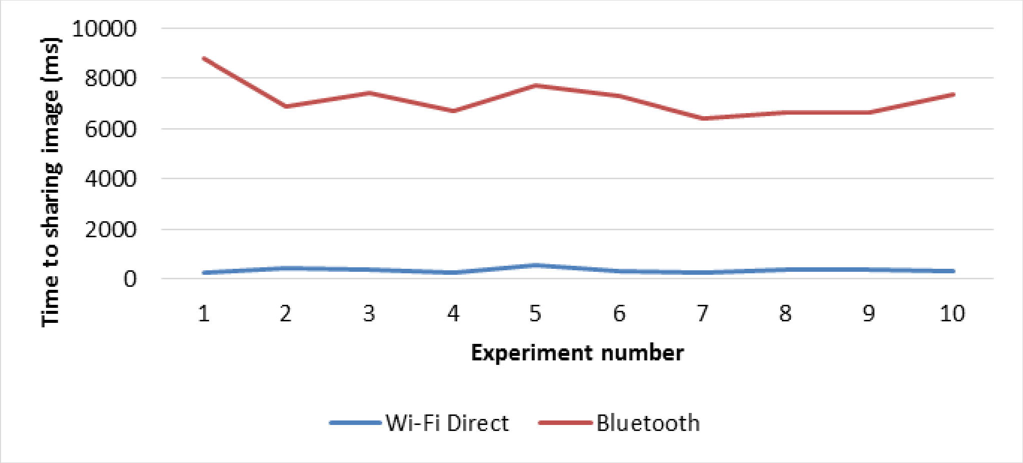


Figure 15. Results of the time for sharing image

In Figure 15, the X-axis indicates the number of test runs and the Y-axis indicates the time taken by My-Direct to send the user image for your partner. Through the analysis of this figure, we note that in all experiments the image upload done by My-Direct via Wi-Fi Direct was extremely faster than Bluetooth. Thus, while the Wi-Fi Direct had a mean of 363.4 ms to send the image, the Bluetooth had an average of 7185.8 ms. This result was expected since the Wi-Fi Direct has a transmission velocity greater than the Bluetooth.

5.3. Experiment: Evaluation of the decision Tree

In this experiment we evaluated the performance of decision tree of the My-Direct, which is used by the *Privacy Module* in the allocation of groups to partners of the user. The evaluation of this tree consisted of two steps.

The first step of the rating was performed in Weka. To this, was used as the test set 40% (177 random instances) of the training set. The confusion matrix of Table 3 shows the result of applying the test set in the decision tree. Of the total of instances present in the test set, 107 (60.5%) instances was classified correctly and 70 (39.5%) were incorrectly classified.

Table 3. Confusion matrix with the results of the evaluation of the decision tree of the My-Direct on Weka.

a	b	c	d	e	As classified
30	20	0	1	0	a = Family
7	**63**	0	0	0	b = Friend
5	8	**3**	0	0	c = Acquaintance
8	16	2	**9**	0	d = Work
0	3	0	0	**2**	e = Other

Although the results generated from the Weka serve as a good assessment parameters, they may not reflect the performance of the decision tree when applied to a real scenario. Thus, it was necessary to perform a second step of evaluation with real users.

To perform the second step five volunteers were recruited. They have been asked to install on your device a mobile application for Android. This application was developed exclusively for this experiment and its source code contained in an implementation of the same decision tree used in My-Direct. Through this application, each volunteer was given a list of up to 20 of your phone contacts classified by the decision tree. Each contact in the list had their rating reviewed by the volunteer, in other words, it was checked whether the group assigned by the classifier was correct. If the group assignment was incorrect, the volunteer corrected it. After evaluating the entire list, the volunteer sending data to a server. The results of these assessments can be seen in the confusion matrix in Table 4.

Table 4. Confusion matrix with the results of the evaluation of the decision tree of My-Direct made by real users.

a	b	c	d	e	As classified
17	16	1	5	0	a = Family
10	**26**	1	2	3	b = Friend
1	1	**3**	1	0	c = Acquaintance
1	2	1	**3**	0	d = Work
0	1	1	0	**1**	e = Other

In the second step, from a total of 97 contacts obtained in the experiment, 50 (51.6%) were correctly classified and 47 (48.4%) were classified incorrectly. These results indicate more precisely the current hit rate of the decision tree.

6. Related Works

In the literature, there are several solutions whose goal is to assist in the development of targeted social environment for mobile applications process. These solutions make use of different architectures, network technologies, methodologies privacy, among other features.

In [7] it is possible to find some of these solutions. Among them, the most common type of architecture is P2P as in [2], [3], [4], [5], [6] and [16], but there are also those that use centralized architecture as [2] or hybrid as [17].

In the mobile environment, the user wants to access information anytime and anywhere. Thus, the construction of middleware with a centralized architecture like [2] is not appropriate. The best option for middleware architecture aimed at mobile environment, according to [7], is the P2P, 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 [2], 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 [18]. Furthermore, transmission rate of this technology is limited, which influences the performance of the middleware.

In the literature also exist middleware that uses two network technologies. In the case of [9], for example, Bluetooth is used for interactivity between nodes, while the Wi-Fi (infrastructure mode) is used to obtain information from Facebook about the affinity between these nodes. In this mode of communication there are two problems: dependence on social networking site for information about the partner and the necessity for access infrastructure.

The MMSN also deal with confidential data such as social relationships, and user activities that can be used to infer other sensitive information about user preferences over time. Thus, as in [3], [5], [17] and [19], a MMSN must manage user data and have policies appropriate control over the exchange of such information, to ensure user privacy. A privacy policy rather widespread in the MMSN projects is assigning users to groups, allowing each group perform only the activities related to it. In the case of [3] and [5], the groups are created and defined by the user. However, none of these works are able to assign users to groups autonomously, taking into account the degree of affinity between them, for example.

Therefore, to correct these deficiencies, this paper presented the My-Direct a MMSN that make use of the Wi-Fi Direct together with the Bluetooth as network solution for the relationship between the nodes and a privacy mechanism that allow the user to sort their peers according to the degree of affinity.

7. CONCLUSIONS

In view of the heterogeneity of middleware solutions for MSN, this paper proposed, substantiated and presented the implementation of a new middleware architecture called My-Direct. This architecture was implemented for the Android platform and evaluated through a case study, the Konect. During this evaluation the My-Direct was subjected to various scenarios where their behavior and performance could be observed.

The My-Direct brought some contributions to the area of middleware for MSN. Among them is the introduction of the use of Wi-Fi Direct in the development of MMSN. From the tests performed was possible to see that this technology can be used to build social applications and its main quality is your rate of data transmission. Moreover, it can also be used in P2P applications as a complement to other network technologies such as Bluetooth, aiming at minimizing problems related to low transfer rate and limited range.

The My-Direct also presents a privacy mechanism that makes use of decision trees for the establishment of social relationships based on the level of affinity between the user and their partners. This level of affinity is measured from the call log and SMS stored on the mobile device.

REFERENCES

[1] A. S. Teles, D. Pinheiro, J. Gonçalvez, R. Batista, V. Pinheiro, J. da S. Silva, M. Endler, (2013) "Redes Sociais Móveis: Conceitos, Aplicações e Aspectos de Segurança e Privacidade", *31º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC).*

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

[3] D. N. Kalofonos, Z. Antoniou, F. D. Reynolds, M. Van-Kleek, J. Strauss, P. Wisner, (2008) "MyNet: A Platform for Secure P2P Personal and Social Networking Services", *Pervasive Computing and Communications*, pp. 135-146.

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

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

[6] D. Bottazzi, R. Montanari, A. Toninelli, (2007) "Context-Aware Middleware for Anytime, Anywher e Social Networks", *IEEE Intelligent Systems*, vol. 22, pp.23-32.

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

[8] WI-FI ALLIANCE, (2010) "Wi-Fi CERTIFIED Wi-Fi Direct, Personal, portable Wi-Fi technology". 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. Accessed in 04 January 2013.

[9] G. Castelli, A. Rosi, F. Zambonelli, (2012) "Design and implementation of a socially-enhanced pervasive middleware", *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference*, pp.137-142.

[10] A. Sahami, H. V. Le, N. Henze, S. Albrecht, (2013) "Automatic Classification of Mobile Phone Contacts". *Proceedings of Mensch & Computer*.

[11] J. Min, J. Wiese, J. Hong, J. Zimmerman, (2013) "Mining smartphone data to classify life-facets of social relationships". *Proceedings of CSCW*.

[12] R. Grob, M. Kuhn, R. Wattenhofer, M. Wirz, (2009) "Cluestr: Mobile social networking for enhanced group communication". *Proceedings of CSCW*, pp. 81-90.

[13] G. Kalic, I. Bojic, M. Kusek, (2012) "Energy consumption in android phones when using wireless communication technologies". *MIPRO, 2012 Proceedings of the 35th International Convention*, pp.754-759.

[14] M. Gargenta, (2011) Learning Android, 1ª ed. O'Reilly.

[15] WEKA. http://www.cs.waikato.ac.nz/ml/weka/. Accessed in 12 December 2013.

[16] S. Kern, P. Braun, W. Rossak, (2006) "MobiSoft: An Agent-Based Middleware for Social-Mobile Applications". *On the Move to Meaningful Internet Systems: OTM 2006 Workshops,* vol. 4277, pp. 984-993.

[17] D. N. Kalofonos and Z. Antoniou, (2008) "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.

[18] G. Singhal and H. Shi, (2009) "Design and Development of µMeter - An Innovative Bluetooth Application". *International Journal of Computer Networks & Communications (IJCNC),* Vol. 1, No. 3, pp.10-20.

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

Authors

Luiz Marcus Monteiro de Almeida Santos is graduated in Computer Science from the Universidade Federal de Sergipe (2011) and Masters in Computer Science from the Universidade Federal de Sergipe (2014). Has experience in Computer Science, with an emphasis on distributed systems, mainly in the following themes: Middleware, Mobile Computing, and Social Networking.

Admilson de Ribamar Lima Ribeiro is graduated in Electrical Engineering from the Universidade Federal de Sergipe (1981), Masters in Computer Science from the Universidade Federal de Pernambuco (2001) and Ph.D. in Electrical Engineering from the Universidade Federal do Pará (2007). Is currently professor in the Department of Computer Science of the Universidade Federal de Sergipe. Has experience in Computer Science with emphasis in Distributed Systems and Computer Networks.