OIoT: A Platform to Manage Opportunistic IoT Communities

David López, Dolors Royo, Esunly Medina, Roc Meseguer
Department of Computer Architecture
Universitat Politècnica de Catalunya
Barcelona, Spain
{davidanl, dolors, esunlyma, meseguer}@ac.upc.edu

***Abstract*—Opportunistic Internet of Things (IoT) extends the concept of opportunistic networking combining human users carrying mobile devices and smart things. It explores the relationships between humans and the opportunistic connection of smart objects. This paper presents a software infrastructure, named *Opportunistic IoT Platform (OIoT)*, which helps developers to create and manage opportunistic IoT communities between smart devices. The platform enables the creation of opportunistic IoT communities that support the AllJoyn communications framework, for IoT devices and applications. Results from a preliminary evaluation of the OIoT platform indicate that this infrastructure is useful to manage and share data across opportunistic IoT communities.**

Keywords—internet of things; opportunistic networks; social mobile collaboration; smartphone-based sensing.

I. INTRODUCTION

The Internet of Things (IoT) is a computing concept that describes everyday physical objects connected to the Internet and being able to communicate with other devices. "Things", in the IoT, can refer to a wide variety of devices such as heart monitoring devices, biochip transponders on farm animals, cars with built-in sensors, devices that assist fire-fighters in search and rescue operations, cell phones, coffee makers, washing machines, TVs, headphones, lamps, wearable devices and almost anything else you can think of. This concept also applies to components of machines, for example, a jet engine of an airplane or the drill of an oil rig.

Due to the fact that most of IoT devices are low power and resource constrained, many new initiatives are being taken for designing lighter communication protocols for this type of devices. Following this trend, protocols like *CoAP* [1], *XMPP* [2] and *MQTT* [3] have been developed, claiming to be real-time publish-subscribe protocols that can connect thousands of IoT devices. On the other hand, other solutions have emerged that propose more complex and sophisticated services for the IoT protocols, like *AllJoyn*[1] or *IoTivity*[2]. These protocols offer advanced functionalities such as device discovery, permanent connectivity and session management, which make them too heavy to be used in small sensors. However, these IoT protocols can be a suitable option for other kind of devices, with more resources and higher capabilities, such as smartphones and tablets. Connecting sensors and objects opens up an entirely new world of possible applications, and it is precisely the specific context of application the thing that will determine the appropriate protocols to be used.

In general, the IoT is viewed as a global infrastructure where all objects are permanently connected. Nevertheless, in this paper we will address the IoT from a different perspective: the opportunistic IoT. The opportunistic IoT is defined in [6] as "an infrastructure which addresses information dissemination and sharing (through IoT devices) within opportunistic communities that are formed based on the movement and opportunistic contact of human beings". Therefore, this aspect of the IoT explores the relationships between humans and the opportunistic connection of smart objects.

In that line, the concept of Opportunistic Mobile Social Networks [4] gains special relevance since it explores some aspects related to opportunistic relationships between humans. Opportunistic Mobile Social Networks are a form of mobile ad hoc networks that exploit the characteristics of human social interactions (e.g., daily routines, mobility patterns, and interests) to route messages and share data. In such networks, users with mobile devices are able to form on-the-fly social networks to communicate with each other and share information.

Opportunistic IoT extends the concept of Opportunistic Mobile Social Networks by merging human users (carrying mobile devices) and smart things belonging. Consequently, this paper proposes a framework to support the connection of human social networks with smart objects in an opportunistic fashion.

An interesting alternative to capture human social interactions is the *Collaborative Sensing Platform (CoSP)*, described and evaluated in [12]. *CoSP* is a mobile software infrastructure designed to automatically measure and share both, low-level information gathered from smartphone sensors, and high-level characterizations of the users' behaviour (built using these low-level data). The *CoSP* platform facilitates data gathering and sharing of information from mobile users, helping to reduce the energy cost related to such processes.

Considering the benefits that the *CoSP* infrastructure offers to characterize human interactions by collecting and disseminating information about mobile users, we based this

[1] https://allseenalliance.org/
[2] https://www.iotivity.org/

paper in the *CoSP* platform, extending it to allow opportunistic connection between the mobile devices of the users and smart objects. Therefore, we enhanced the *CoSP* with a new module, the *Opportunistic IoT Manager*, which allows developers to create and manage opportunistic IoT communities. In addition to the services provided by the *Opportunistic IoT Manager*, we included other functionalities that are useful for the management of opportunistic IoT communities.

The new services offered to support opportunistic IoT is what we called the *OIoT* platform, which includes:

1) User/device profile editing. The profile can contain the user's preferences, interests, sensed and shared data, etc.
2) Managing opportunistic IoT networks.
3) Publishing and subscribing services.
4) Dissemination of opportunistic data.

OIoT framework is also used as a base to create an opportunistic IoT social application, *OpportunisticMeeting***.** This application was designed with the purpose of helping students and lecturers to establish opportunistic contacts with other students, who share the same hobbies, subjects, preferences, etc. The *OpportunisticMeeting* also allows the dissemination of questions, requests, adverts, etc., as well as the replies between acquaintances and students who have not necessarily established any previous contact.

In that sense, the capabilities of the *OIoT* platform to support opportunistic communication and data sharing within a given community was also analysed and discussed based on the *OpportunisticMeeting* application. The obtained results indicate that the services of the platform are useful to assist end-users and developers of mobile applications, regardless of the technical restrictions imposed by the use of *AllJoyn* as a communication support.

Next section presents the related work. Section III describes the proposed collaborative sensing platform. In Section IV, the *OpportunisticMeeting* application is described. Section V presents the evaluation conducted to determine the usefulness of the proposed infrastructure. Finally, Section VI presents the conclusions and the future work.

II. Related Work

Several research studies have faced the challenge of defining new architectures for the Internet of Things [10], [11]. As a result, several platforms have appeared in order to provide connectivity between all kinds of IoT actors [9]. Some of these studies introduce the opportunistic aspect of IoT systems [6]. In [5] the authors develop the Magic Broker 2, a lightweight middleware that supports spontaneous interaction between smart devices (public displays, smartphones). This platform offers a uniform web-based API for building IoT applications.

Various recent works explore human factors of IoT systems [7], especially the interaction between online and offline social communities. A reference architecture for developing opportunistic IoT systems is proposed in [6]. This work also shows how to use the proposed architecture to design applications for opportunistic mobile social networking. In addition, this study considers opportunistic IoT ad hoc networks, formed with smartphones and vehicles using Bluetooth and Wi-Fi devices.

Another interesting work reported in [8] addresses the problem of enabling interaction between smart objects and mobile users in the IoT. It proposes the use of direct connections between object and users through Bluetooth in order to mitigate the need for internet connectivity and the pre-installation of user interfaces. The aim of this work is the dissemination of the smart object interfaces through opportunistic mobile networks.

As explained previously, we extended the *CoSP* infrastructure, presented in [12] to enable the creation and management of opportunistic mobile IoT communities. Therefore, this paper presents the *Opportunistic IoT Platform* (*OIoT)*, which encapsulates a set of opportunistic IoT services. The integration of *OIoT* with *CoSP* can support the development of mobile applications to inform the users about events and context information collected from their smart devices and also to facilitate the dissemination of such information within opportunistic communities. This article also evaluates *AllJoyn* and *CoAP*, which are emerging communication protocols in the IoT area.

III. The OIoT Platform

In this section we describe the *OIoT* platform developed to support data sharing among opportunistic communities. Moreover, we describe the integration of the *OIoT* with the *CoSP* platform. In this description we include details of both the functionalities previously implemented in the *CoSP* as well as the new features developed in the *OIoT*.

As shown in Fig. 1, the *CoSP* platform, composed by four main modules (the *Sensing Module*, the *Data Manager*, the *Users and Groups Manager* and the *Sharing Manager*) was extended with a fifth module, the *Opportunistic IoT Manager*. Furthermore, the *OpportunisticMeeting* application was developed using the services provided by the *CoSP* and the *OIoT* platforms. Hereafter we will refer to the integration of both platforms as the *OIoT-CoSP* platform.

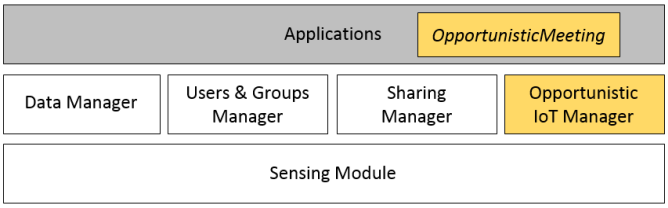

Fig. 1. Software architecture of the *OIoT-CoSP* platform

A. Sensing Module

This module supports the data collection process from the smartphone's sensors. It has a modular design, where hardware, software and human-based data can be collected from independent sensing units. This module is implemented as an *Android* library that calls a number of services for accessing the sensors available on the device. The *Sensing*

Module (SM) can obtain raw sensor data from the smartphone's physical and virtual sensors but also higher level context information from other sources. It also implements two different communication frameworks, *AllJoyn* and *CoAP*, as services in order to provide shared sensor data as input for the system. In addition, the *SM* allows both, remote and local activation of the all the sensing services, and also a flexible configuration of the sensing frequency and waiting times.

B. *Data Manager*

The *Data Manager (DM)* is in charge of specifying the sources that the system will use to collect the input data. Therefore, users can capture data at different levels and from different sources, and they can also decide what information they want to share with others. In other words, the system allows a selective distribution of data from different sources.

Fig. 2 illustrates the data sharing process conducted by three different users who have activated various kind of sensors at different levels. Level 1 corresponds to physical and virtual sensors, level 2 includes all the sources of static contextual information and level 3 represents high level sources that aggregate or process information.

In Fig. 2, *user A* shares data that comes from all three levels: information from two physical sensors, contextual information from one data source, and also high-level data from a logical sensor. *User B* only shares data from one high-level source. Finally, *user C* shares information from sensors of level 1 and level 2.

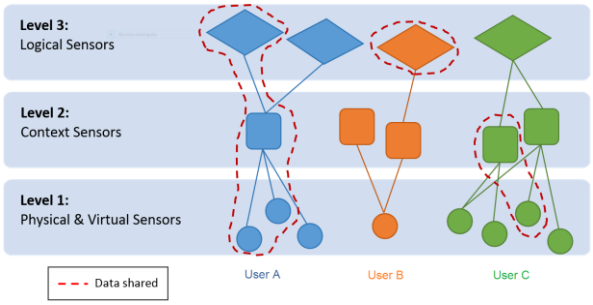

Fig. 2. Example of data sharing at different levels

Next, we specify the different sensors and data sources supported:

1) ***Physical sensors:*** They are devices embedded in most commercial smartphones, and we differentiate between three kinds of physical sensors:
   a) *Hardware sensors*, such as accelerometer, *GPS*, ambient light, dual microphones, proximity sensor, dual cameras, compass and gyroscope.
   b) *Communication sensors* that correspond to the several built-in communication interfaces of modern devices, e.g., Bluetooth, infrared, Wi-Fi and cellular antennas.
   c) *Performance sensors*, such as battery level, network traffic, and *CPU*, memory and disk utilization.
2) ***Virtual sensors:*** In this group we consider information that can be obtained from the smartphone's applications and services; e.g., the screen status, the user's touch inputs, applications status, log files and notifications.
3) ***Context sensors:*** These modules collect contextual information from existing repositories; for instance, the user's profile, preferences, or team members.
4) ***Logical sensors:*** These sensors provide high-level information and they can combine data from several sources. Information from this category usually involves some type of aggregation and processing to interpret the sensed data and contextual information. An example of logical sensors could be a service that interprets raw data from an accelerometer to infer users' mobility patterns.

C. *Users and Groups Manager*

The *Users and Groups Manager (UGM)* is responsible for the creation and administration of non-opportunistic mobile users and groups within the system. Every user must be registered through this module for accessing the platform services. Although, a minimum group size of 1 is allowed, it is necessary to belong to a group of two or more members in order to be able to share information. This module is composed of the *CoSP Server*, which is connected to the *Google Cloud Messaging (GCM)* service. The users must register through the *GCM* service to get a registration ID. Once the ID is received by the users' application, it must be sent to the *CoSP Server*, which can use the *GCM* service to send messages to the users.

The *CoSP Server* can use information from the context sensors (e.g., the users' profile, their sharing preferences or group memberships) in order to arrange the different groups registered in the system. It can also establish a set of rules for the different groups to define how, when and what messages will be sent to the users. The *CoSP* platform allows sharing context information and sensor data between members of the same group, and among different groups.

D. *Sharing Manager*

The *Sharing Manager* (SM) controls the data sharing process and establishes the sharing policies. *CoSP* implements two different data dissemination mechanisms. The selection of one mechanism or the other depends mainly on whether the devices are nearby, and if they are connected through a Wi-Fi access point to the same local network or if they are in different networks. However, there might be some cases where the selection of the sharing mechanism does not depend on the physical proximity of the devices. In general, for distant users whose smartphones are connected to different local networks (but they are reachable through the Internet), the data sharing is mediated by the *CoSP Server*.

The users can decide to utilize the *CoSP Server* to perform some resource intensive aggregation or processing tasks to optimize the use of local hardware resources. For instance, if a group of users want to share high-level representations of their behaviour, collaboration or interaction patterns, they would typically require the *CoSP Server* to perform some processing

on their behalf. As a consequence, they would also have to use the *GCM* service to disseminate data.

On the contrary, for nearby devices connected to the same access point, the platform offers two different communication frameworks, *AllJoyn* and *CoAP*. Both offer different mechanisms to create a *Peer-to-Peer* proximity network amongst devices. This means that both frameworks enable devices to share information directly among them, without depending on any centralized server. The main advantage of using this mixed procedure for data sharing is that we can make a very efficient use of hardware resources when the users are physically close.

This fact opens up the possibility to integrate this platform with *Internet of Things*, allowing, for instance, that the users' devices could interact with nearby networked objects, such as printers, electronic whiteboards, public displays, etc.

Regarding the sharing policies, the *SM* allows the definition of a set of rules to determine what data to distribute, and with whom. According to these rules, any user and group registered in the system can configure their personal sharing preferences, by choosing one of the following options: (1) share information with any nearby user registered in the system, (2) share data only between members that belong to the same group, (3) share everything, (4) share data collected from only a subset of sensors, or (5) share data according to the result of the *Group Resource Optimization Algorithm (GROA)*.

The *GROA* algorithm determines, for the members of a given group, who has to collect data from specific sensors and share the results with the rest of the members. Thus, it helps optimize the use of the resources of the whole group. The *GROA* can determine, depending on the battery level and the memory usage of the users' devices, which user have to capture data from microphone and who have to sense *GPS* data, whereas the rest of the users should deactivate their sensors and wait until they receive the sensed information from their group mates.

Depending on the sharing policies applied, the platform establishes (for every sensor in the smartphone) how it will obtain the corresponding data. Accordingly, the platform defines two basic data collection methods: *direct* and *indirect*. In the direct method the device's sensor is responsible for capturing data without relying on any other source. In the second case, the sensor receives and processes data that has been previously collected by other devices. The system uses both, *GCM*, *AllJoyn* or *CoAP* to receive such information.

E. Opportunistic IoT Manager

This *Opportunistic IoT Manager (OIM)* is in charge of managing the opportunistic network, which will allow any device connected to the network to publish the services that it is offering, to subscribe to the services from which the device wants to receive and share data. All communication services implemented in this module are based on *AllJoyn* or *CoAP* framework, depending on the device restrictions.

This module offers four basic functions: *initialization of opportunistic IoT networks*, *service publishing*, *service subscribing*, and *data dissemination*.

Initialization of Opportunistic IoT networks. With this service opportunistic IoT mobile networks can be created. A different name-identifier pair is assigned to each network. This identifier can be chosen by the user or can be set by default. If the identifier is set by default, the system assigns a name-identifier that matches with the name of the application that created the network. This service was implemented using *AllJoyn* because it allows designers to create easily a fault-tolerant system for publishing and discovering services.

Publishing service. This service allows new devices connected to the network to publish their profile, which contains their preferences and the sensed data that they want to share with others. This service was implemented using the *AllJoyn* framework, and it can also be used by *CoAP* devices to publish their services.

Subscribing service. This function allows devices to subscribe to the services offered by other users or devices. It was implemented using both protocols *AllJoyn* and *CoAP*.

Data dissemination. This service enables users that are connected to an opportunistic IoT network not only to share data and profile information, but also to exchange information between subscribers of a specific service. Devices can store profiles and sharing information and forward it later to other opportunistic IoT networks. All messages that are re-send between opportunistic networks are assigned a TTL (Time-To-Live). The system defines two different TTL mechanisms according to (i) the number of hops (e.g., number of opportunistic IoT networks to be re-sent) and (ii) the number of hours or days. The service uses *AllJoyn* or *CoAP* depending on the device restrictions.

All the modules were embedded into the *Unity* framework [3] to facilitate the development of mobile applications. Although *Unity* is multiplatform, the current version of the *OIoT-CoSP* only allows the development of applications for *Android* and *Windows* OS.

IV. USE CASE: THE OPPORTUNISTICMEETING APPLICATION

As mentioned before, the *OIoT-CoSP* platform can be easily embedded into opportunistic mobile social sensing applications. As a proof of concept, we implemented the *OpportunisticMeeting* mobile application for *Android* devices.

This section describes a case study exploring the use of such application by students enrolled at the *Castelldefels School of Telecommunications and Aerospace Engineering (EETAC)* of the *Universitat Politècnica de Catalunya (UPC)*, Spain. The *OpportunisticMeeting* is intended to help these students to stablish opportunistic contacts with other peers during their time at University. Such opportunistic interactions are based on interests shared by students, such as sports,

[3] http://www.unity3d.com

hobbies, courses, learning materials, etc. The application allows the dissemination of questions, requests, adverts, etc., as well as the replies from students who have not necessarily established any previous relationship. This way, the *OpportunisticMeeting* could facilitate the integration of new students and the creation and strengthening of bonds of collaboration and cooperation between students.

Fig. 3 shows some screenshots of the *OpportunisticMeeting* application running in *Android* smartphones. Similar graphical interfaces were also implemented for other types of devices such as laptops or tablets.

The *OpportunisticMeeting* application was implemented in a way that allows any smart device (e.g., computers, mobiles, tablets, etc.) running it to create a new network or connect to a predetermined set of networks. Currently, the system only supports three different types of opportunistic networks:

- *Class network*. This kind of network facilitates sharing of information between lecturers and students within the context of a particular class. Therefore, only users involved in the class (e.g., lecturers and students attending the class) can be connected to the network and it only exits during that specific timeframe.

- *Course network*. This type of network supports cooperation among students and lecturers within the context of a particular course. Consequently, this kind of network allows the connection of the smart devices of the users involved in a common course (e.g., lecturers and students enrolled) anytime and not only during the class sessions of the course.

- *Campus network*. This kind of network allows any smart device on the *EETAC* campus to be connected and share data at any given time.

Moreover, Fig. 3.a depicts the main menu of the *OpportunisticMeeting*, where the two basic functionalities of the application are displayed: *Set Profile* and *Select Network*.

The *Set Profile* function loads the profile form, as shown in Fig. 3.b. In this form the users must introduce an e-mail, select some topics of interest (e.g., courses enrolled, questions, etc.) and specify what we called "trade information". The trade information is what the students are offering or requesting (the "trade" field shown in the screenshot) from/to their peers. For example, a user can offer his solution for a given class assignment in exchange for a lecture's notes. Once the users fill in this form, the next step (Fig. 3.c) is to select the resources (e.g., pictures, GPS coordinates, profile information, etc.) that they are going to share across the opportunistic network.

Only when the profile information is completed, the users can access to the *Select Network* functionality (disabled by default). Then, a new menu is loaded in the phone' screen (Fig. 3.d). In this menu the users can select the opportunistic network to which they are going to connect to. In the example of the figure, there are three different opportunistic networks created: the *"Operative Systems Class"*, the *"Programming Course"* and the *"Campus Community"* networks. These networks correspond to the three types of opportunistic networks supported by the *OpportunisticMeeting* application, the *Class*, *Course* and *Campus* networks, respectively. Notice that these opportunistic networks depend on the Access Point (AP) to which the user is connected and that only users connected to the same AP can disseminate information within that particular network. Moreover, it is possible to have different networks of the same type belonging to different APs. Therefore, users connected to those different networks cannot communicate between them in real time. However, these users would be able to communicate at some point if they change location and get connected to the same type of network and AP.

Access Points can be either routers installed in a fixed network infrastructure or mobile devices of the users running the *OpportunisticMeeting* application. When a user tries to connect to an opportunistic network and there is none available, the application can create the network by either (i) connecting to an existing AP or (ii) configuring automatically the user's device as AP to provide wireless access to the network. As a result, opportunistic networks can be created anywhere and anytime.

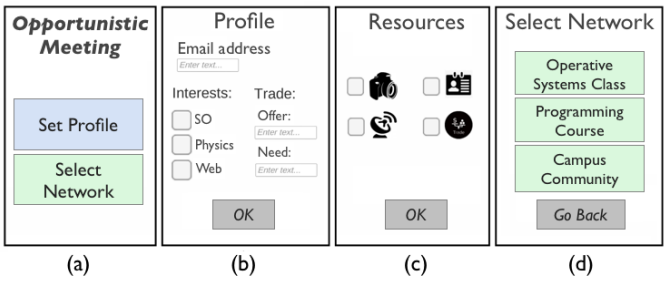

Fig. 3. *OpportunisticMeeting* interface

As an example of data dissemination, let us imagine the following scenario as depicted in Fig. 4. Supposing that in our campus we have identified three main locations where students spend long periods of time: the *Classrooms*, the *Cafeteria* and the *Library*. Several opportunistic networks with the *"Campus Community"* identifier can be established in each zone at the same time.

Then, suppose that *User A* and *User B* are in the *Classrooms* zone, *User C* is in the *Cafeteria* and finally, *User D* and *User F* are in the *Library*. First, *User B* shares its profile and trade information with *User A* through the *"Campus Community* "opportunistic network of the *Classrooms* zone, (Fig. 4.a). After a while, *User B* goes to the *Library* in order to do some homework. When *User B* connects to the *"Campus Community"* opportunistic network existing in the *Library*, he automatically shares the profile and trade information of *User A* (collected while these users were together in the *Classrooms* zone) with *User D* and *User F*, as shown in Fig. 4.b.

In the case of the *"Campus Community",* either the user can choose to join to an existing opportunistic network (as shown in Fig. 3.d) or it is the application itself which every

certain time interval searches for opportunistic networks to join, sharing information related to the own user and disseminating other users' information. By contrast, the data dissemination in *"Operative Systems Class"* and *"Programming Course"* networks only can be done manually by the user.

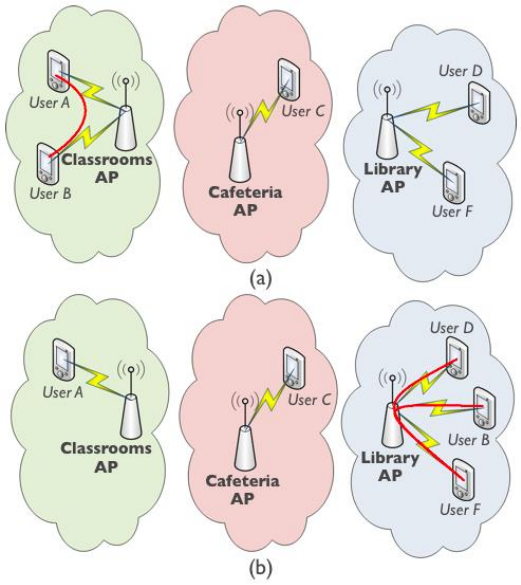

Fig. 4. Example of information dissemination

It is possible for a user to change, at any time, the information (e.g., profile, shared resources, etc.) that he had introduced previously in the application. If this information had already been shared, the application sends an update message to all peers with the new data. If a shared resource is no longer available, all the remote users automatically unsubscribe from it.

The development of this application was quick, easy and involved a low risk, because most of the required services were already provided by *OIoT-CoSP* platform. Therefore, the challenge was mainly to use the services and develop the user interface to show the information to the users. Developers of several types of opportunistic IoT applications can take advantage of this platform to reduce the development costs, complexity and risks of their projects.

V. Evaluation Process

AllJoyn is an open source framework, originally developed by *Qualcomm*, which is being promoted by the *Allseen Alliance*. It provides a mechanism for IoT devices to discover and communicate with each other. *AllJoyn* offers more sophisticated services than other frameworks, such as device discovery, permanent connectivity and session management that make it too heavy for small sensors, but can be a suitable option for devices with higher capabilities like smartphones or tablets.

In order to evaluate the efficiency of *AllJoyn* protocol in opportunistic IoT environments we conducted an experimental study to evaluate the energy consumption of this protocol. We also compared the performance of *AllJoyn* with the *CoAP* protocol, which is a low overhead IoT protocol, intended to be used in lightweight devices.

To test the energy consumption of *AllJoyn*, several mobile devices were monitored during an interval of 20 minutes in a two round process. In the first round, four fully charged mobile devices in its normal state (without the *AllJoyn* process activated) were evaluated in terms of battery consumption. In the second round, the same mobiles where evaluated but now with the *AllJoyn* activated. Notice that all these devices were connected to the same opportunistic network. From these tests we obtained very small variations between the two rounds, with differences from 0.1% to 0.3% of total battery consumption, depending on the device (newer devices present smaller variations). Therefore, these results clearly show that *AllJoyn* is very efficient in reference to the energy costs involved.

Although the battery consumption using smartphones was not very high, it is important to note that there might be other kinds of IoT small autonomous devices where it is crucial to maximize the battery lifetime and therefore even small percentages can be significant. For this reason, our framework also supports the use of lightweight IoT protocols like *CoAP*.

To compare the battery consumption of *AllJoyn* versus *CoAP*, two applications that share geolocation data from different devices were developed using the *OIoT-CoSP* framework. The first one uses *AllJoyn* and the second one uses *CoAP* as communication protocols.

Fig. 5 depicts the percentage of battery consumption over a period of 20 minutes for each application and using five different devices (Smartphones 1, 2 and 3 are *HTC Desire* devices; Smartphone 4 is an *HTC One* device and smartphone 5 is a *Samsung Note II* device). As shown in the figure, the consumption is lower in the case of *CoAP*. Nevertheless, for some devices the energy drain is very similar for both protocols. In addition, it is important to consider the fact that the main drawback of the *CoAP* protocol is that it is very simple and it does not include session and discovering functionalities. Therefore, it is expected to consume less energy than a more complex protocol.

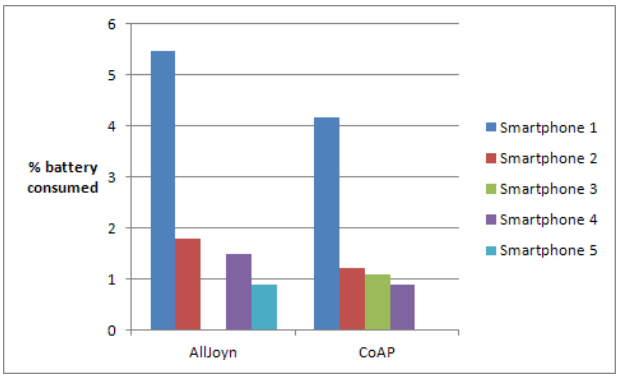

Fig. 5. Battery consumption *AllJoyn* vs *CoAP*

The results obtained suggest the potential benefits of implementing a hybrid platform that uses *AllJoyn* for management and session discovery and *CoAP* for data dissemination. We believe that in the case of high performance devices, such as smartphones, tablets, etc. it would be interesting to have both protocols available and use the most appropriate one depending on the circumstances.

Furthermore, we tested the usefulness of the *OIoT-CoSP* platform for message dissemination through the "Campus Community" network. We considered the following scenario: Three students, Bob, Alice and Charles, are on campus and use the *OpportunisticMeeting* application to interchange messages between them. Charles sends a message to the *"Campus Community"* opportunistic network. Charles wants his message to be distributed across campus and reach the maximum number of people. The application monitors the information exchanged between all the devices connected to the network. For each information exchange, it records a unique identifier, the sender and the receiver identifiers, the geolocation of the sending device, the time when the message was sent and if it has was received or not. Fig. 6 shows an example of these data as recorded in the application database.

id	id_sender	id_receiver	timestamp	longitude	latitude	has_message
80	5c520b5b4cb4b884	774f2b8e6fe50218	2015/04/28 09:31:35	1.985253096626831	41.27535581588745	false
85	774f2b8e6fe50218	5c520b5b4cb4b884	2015/04/28 09:32:33	1.985226273536821	41.27536654472351	false
81	5c520b5b4cb4b884	774f2b8e6fe50218	2015/04/28 09:45:10	1.985253096626831	41.27535581588745	false
88	774f2b8e6fe50218	5c520b5b4cb4b884	2015/04/28 09:45:15	1.985226273536821	41.27536654472351	false
82	5c520b5b4cb4b884	774f2b8e6fe50218	2015/04/28 09:55:34	1.986132784978371	41.27551493074396	false
89	774f2b8e6fe50218	5c520b5b4cb4b884	2015/04/28 09:55:38	1.986064482413495	41.275541545654	false
86	cb600c2965ca2553	774f2b8e6fe50218	2015/04/28 10:06:31	1.9875174863736	41.275280568	true
87	774f2b8e6fe50218	cb600c2965ca2553	2015/04/28 10:06:31	1.9874816800000001	41.27531399	true
83	5c520b5b4cb4b884	774f2b8e6fe50218	2015/04/28 10:07:12	1.9875174863736	41.275280568	true
90	5c520b5b4cb4b884	cb600c2965ca2553	2015/04/28 10:07:17	1.9874816800000001	41.27531399	true
84	774f2b8e6fe50218	5c520b5b4cb4b884	2015/04/28 10:07:34	1.987548878928	41.2752813539	true
91	cb600c2965ca2553	5c520b5b4cb4b884	2015/04/28 10:07:39	1.987548878928	41.2752813539	true

Bob
Alice
Charles

Fig. 6. Data sent over the opportunistic network

Moreover, Fig. 7 depicts in a map the different locations where Bob, Alice and Charles interchange data (using coloured markers) until the moment when Charles' message reaches Bob and Alice. It clearly shows that when the students meet in the same physical location, opportunistic information exchange processes take place.

Fig. 7. Campus map showing the locations of different opportunistic data exchanges

Considering the information represented in Fig. 6 and Fig. 7, we can draw the following conclusions: Bob is in the *Cafeteria* and the *OpportunisticMeeting* application running in his smartphone creates a *"Campus Community"* opportunistic network in this location. Alice happens to go to the *Cafeteria* as well so her device joins the opportunistic network created by Bob (at 9:31, as shown in Fig. 6). After 5 minutes from the creation of the network, the system sleeps for an interval of 5 minutes, and then all the process starts again. Neither Bob nor Alice have received Charles' message yet.

After a while (around 9:55), both of them go to the *Classrooms* zone, where they meet Charles. Once again, Bob's smartphone creates a new *"Campus Community"* opportunistic network (around 10:06). Charles's joins this network and sends the message that he wants to disseminate to Bob's device. Alice's device also joins the network and receives the message from Bob's at 10:07 (Bob's device is able to send Charles' message because it had received the message previously). In the end, Charles message was received by Alice and Bob's. Thus, the dissemination over the opportunistic networks worked as expected.

As shown in the previous example, from a developer point of view, the use of *AllJoyn* has some clear advantages, facilitating the processes of (i) creation and management of communication sessions, (ii) discovery of devices, (iii) management of the session state (e.g., when a user is connected or disconnected), and (iv) data dissemination. Our experimental evaluation also indicates that *AllJoyn* is very robust against connection loss.

On the other hand, the current implementation of *AllJoyn* has an important drawback: all users must be connected to the same AP in order to be able to join the same opportunistic network. Therefore, developers that want to use the *OIoT-CoSP* platform to implement mobile opportunistic applications must consider this constraint. However, *AllJoyn* developers claim that it will support, in a near future, other communication protocols like Wi-Fi Direct, which could help overcome this limitation.

In spite of some restrictions imposed by the use of the *AllJoyn* protocol, results from our evaluation indicate that the *OIoT-CoSP* platform is a suitable alternative for the development of different types of social opportunistic applications.

VI. Conclusions and Future Work

This paper introduces *OIoT* framework, a software infrastructure that facilitates the creation and management of opportunistic IoT communities. A preliminary evaluation of the platform indicates that it is an effective communication support for opportunistic IoT networks. The *OIoT* helps developers of mobile collaborative sensing systems to reduce the complexity, cost and risks of their projects.

As part of the future work we plan to evaluate more in-depth the communications protocols; particularly, developing a hybrid system that combines both *AllJoyn* and *CoAP* protocols simultaneously. This systems would we useful to provide

support to heterogeneous and demanding ecosystems in a more efficient way.

Finally, we intend to perform a field study using *OpportunisticMeeting* to support opportunistic social interactions between students at the *Castelldefels School of Telecommunications and Aerospace Engineering (EETAC)* of the *Universitat Politècnica de Catalunya (UPC)*, Spain. This study would be useful to evaluate the *OIoT* framework in a real-world University environment and determine its usefulness in promoting cooperation and collaboration within the students' community.

ACKNOWLEDGMENT

This work was partially supported by the Consolidated Research Group 2014-SGR-881 of the Generalitat de Catalunya.

REFERENCES

[1] M. Kovatsch, M. Lanter, Z. Shelby, "Californium: Scalable Cloud Services for the Internet of Things with CoAP," The 4th International Conference on the Internet of Things (IoT 2014).

[2] M. Kirsche, R. Klauck, "Unify to Bridge Gaps: Bringing XMPP into the Internet of Things," Pervasive Computing and Communications Workshops (PERCOM Workshops), IEEE International Conference, 2012, pp. 455-458.

[3] M. Collina, G. Corazza, A. Vanelli, "Introducing the QEST broker: Scaling the IoT by bridging MQTT and REST," Personal Indoor and Mobile Radio Communications (PIMRC), IEEE 23rd International Symposium, 2012, pp. 36-41.

[4] N. Vastardis, K. Yangi, "Mobile Social Networks: Architectures, Social Properties, and Key Research Challenges". Communications Surveys & Tutorials, IEEE, 2012, pp. 1355-1371.

[5] M. Blackstock, N. Kaviani, R. Lea, A. Friday, "MAGIC Broker 2: An Open and Extensible Platform for the Internet of Things," in Conference Internet of Things (IOT), 2010, pp. 1-8.

[6] B. Guoa, D. Zhanga, Z. Wanga, Z. Yua, X. Zhoua, "Opportunistic IoT: Exploring the harmonious interaction between human and internet of things," Journal of Network and Computer Applications. Volume 36, Issue 6, November 2013, pp. 1531-1539

[7] A. Monares, S. F. Ochoa, V. Herskovic, R. Santos, and J. A. Pino, "Modeling interactions in human-centric wireless sensor networks," in Proc. of CSCWD, 2014, pp. 661-666.

[8] H. Wirtz, J. Rüth, M. Serror, J. Ágila Bitsch Link, K. Wehrle. "Opportunistic interaction in the challenged internet of things," Proceedings of the 9th ACM MobiCom workshop on Challenged networks, pp. 7-12.

[9] J. Mineraud, O. Mazhelis, X. Su, S. Tarkoma, "A gap analysis of Internet-of-Things platforms", unpublished.

[10] M. V. Moreno, J. L. Hernandez; A. F. Skarmeta, M. Nati, "A Framework for Citizen Participation in Internet of Things," 28th International Conference on Advanced Information Networking and Applications Workshops, pp. 815-820.

[11] J. Rodriguez-Covili, S. F. Ochoa, J. A. Pino, J. Favela, D. Mejia, and A. L. Moran, "Designing mobile shared workspaces by instantiation," in Proc. of Int. Conf. on Computer Supported Cooperative Work in Design (CSCWD), 2009, pp. 402-407.

[12] E. Medina, D. López-Nuevo, D. Royo-Vallés, R. Meseguer, S. Ochoa. "CoSP: A Collaborative Sensing Platform for Mobile Applications," CSCWD, 2015, in press.