Smart City Framework based on Distributed Mobile Devices System Design and Implementation

Daniel Soleimani Pour

Master Student of Department of Electronics, Informatics and Bioengineering
Polo Regionale di Como
Politecnico di Milano
Email: daniel.soleimani@mail.polimi.it

***Abstract*—Smart Cities and smart services delivered by them are considered a new paradigm in computer engineering. Due to the daily increase in usage of portable smart devices, as smartphones or smartwatches, which provide different sensors and connectivity facilities, they were used to design a framework and implement a wide-area network with distributed nodes in order to provide continuous and non intrusive interaction with various urban facilities. We focus on three main modules: hybrid human localization (Indoor and Outdoor) using Bluetooth Lite Energy (BLE), GSM (Global System for Mobile Communications), GPS (Global Positioning System) localization and the Smart City Framework (SCF), which handles interaction with various external services (i.e., transportation organization, municipal and traffic authorities). The communication between the two aforementioned modules is done via a mobile application, which interacts directly with the user and the remote SCF framework itself.**

Keywords*—*Assistive technology; Smart City; Mobile computing; Smart homes; Wearable computers; Helping services; Internet of things.

I. Introduction

More than half of the worlds population lives in urban areas. The growth of population, which needs various urban and social services, due to increase in people aging, causes several problems for the society [1]. In this case, Smart Cities are suggested as a partial solution for the new era metropolitan areas. As of today, there are several definitions for smart cities, which analyses and groups them according to different aspects of their study.

- A city that monitors and integrates conditions of all of its critical infrastructures, including roads, bridges, tunnels, railways, subways, airports, seaports, communications, water, power, even major buildings, can better optimize its resources, plan its preventive maintenance activities, and monitor security aspects while maximizing services to its citizens [2].
- A city connecting the physical infrastructure, the IT (Information Technology)infrastructure, the social infrastructure, and the business infrastructure to leverage the collective intelligence of the city [3].
- A city combining ICT (Information and Communication Technology) and Web 2.0 technology with other organizational, design and planning efforts to dematerialize and speed up bureaucratic processes and help to identify new, innovative solutions to city management complexity, in order to improve sustainability and livability [1].
- The use of Smart Computing technologies to make the critical infrastructure components and services of a citywhich include city administration, education, health care, public safety, real estate, transportation, and utilitiesmore intelligent, interconnected, and efficient [4].

The common property in these definitions is the presence of an IT backbone which interconnects and communicates with the various parts and modules of a Smart City, in order to benefit its citizens with a unified framework and handling optimally all social needs present in the city. Meanwhile, mobile and wearable devices are becoming much more available and popular, as they are integrating various hardware/software capabilities on a single device. Integration of different services and smart devices opens several opportunities for citizens which could hardly be achieved, if even possible, without using them. As an example, a blind citizen registered in the system can trigger the street light and stop the traffic on an intersection just by standing near the sidewalk and receive a confirmation on his/her device about the moment when it is safe to cross; also, the street light could be notified about the moment when the person is far and safe enough in which to switch back the light to green for cars. We can also imagine an elderly traveling underground in a subway line. The smart city framework will be capable of notifying the person prior to arrival at the place and the time of getting off. To achieve the desired goals, there is a need for a localization scheme, which offers indoor and outdoor localization. Currently, GPS and GLONASS (GLObal NAvigation Satellite System) are considered as the diffused outdoor localization technologies. GPS, and similarly GLONASS, which uses a similar technology, are attractive options for outdoor environments, but are not suitable for indoor applications because they need a clear line-of-sight to orbital satellites in order to track position. This requirement makes them unsuitable for indoor uses, and a different approach must be taken for indoor localization. Several solutions have been developed in research, implementing different techniques as use of RF (radio frequency) signals, WiFi localization, ZigBee, etc. Most RF solutions are sensitive to the number of humans present in the environment and use proprietary and/or expensive components with high power consumption [5]. Another common issue in indoor localization systems is peer2peer communication between nodes, which is a problem in urban areas due to the high number of nodes and

 ISBN: 978-1-61208-446-6

the distance between them, which limits reach of a node with its neighbours. Bluetooth has been used among other Radio Frequency (RF) technologies for indoor localization since it is a cost effective and easy-to-deploy solution [6] with an open core specification [7]. In this paper, we propose a Bluetooth Low Energy (BLE) localization solution integrated with a Smart City backbone framework. The proposed localization solution has a low cost (8.5 USD per node). It is easy to deploy as it does not need node configuration, and it has a high battery autonomy (from 20 months up to 6 years depending to the battery type used). It is built with commercial off-the-shelf (COTS) components. The Smart City backbone framework is a modular system, which handles both data storage and reasoning based on available data it receives by its clients. The system is based on distributed servers, which are divided per zones for increased efficiency, while continuously synchronizing data between each other. In this work, first we discuss the general system architecture, then in Section III we describe our hybrid indoor/outdoor localization. Following, in section IV, we talk about the framework backbone and its design. Also the services provided by the framework are discussed. In Section V, the client side application is explained. This application is responsible for the communication between the server and the user. In Section VI, we discuss about power consumption issues and solutions provided by our system and finally, in Section VII, we discuss experimental results, and conclude in Section VIII with a brief overlook to the work.

II. System Overview

The system is based on three separated modules. The first module discussed, the localization system, has been specifically developed for this work. This module, by itself, is based on two separated localization systems. Outdoor localization, which in our case is GPS, and indoor localization system, which has been developed based on bluetooth low energy modules. These two outdoor/indoor localization systems combine together to form a hybrid system capable of continuous localization whether outdoor or indoor. The second module is the Smart City Backbone Framework, which acts as the central server and processes the received data, handles storage and reasoning and integrates the services offered by third party providers. It is a distributed application, which is installed and running on multiple servers. This is to avoid network slowdowns for the occasions in when it experiences high network traffic. These two modules communicate together using mobile applications installed on smartphones, which is considered the systems third module (client side application).

III. Localization

The localization method will be studied for a hybrid setting, including both indoor and outdoor environments. BLE transmitters will have known positions registered on the server and BLE receivers without known positions. In our test case, two Bluetooth 4 low energy android devices (OnePlus One and Motorola Moto G), will act as BLE receivers moving freely among them, carried by users. There are different methods for localization such as Cell Identity (CI), Angle of Arrival (AOA), Time of Arrival (TOA), Time Difference of Arrival (TDOA), and received power level based localization. All of these methods have their own advantages and disadvantages, but in general received power based method may be considered as an easy way to do localization, considering both hardware and software requirements [8]. The main problem with RF techniques in indoor environments is that physical obstacles absorb the signals and cause RF techniques to be unreliable for indoor measurements[5]. Bluetooth devices measure the received power level indirectly by using RSSI (received signal strength indicator), which is a measurement of the power present in a received radio signal, and it is implemented in the Bluetooth module and can be read easily [6]. Consequently, the received power level based localization seems to be the most applicable one for Bluetooth [9]. As mentioned previously, as any RF signal, we experience the problem of power absorbment by physical obstacles. To handle this situation, we have converted our measurement values from a continuous system to a discrete system with a three level threshold, which are: Immediate (below one meter), Near (above one meter and below 10 meters) and Far (more than 10 meters). The accuracy of the wireless localization systems relies heavily on calibrated distance metrics. Using RMSE (root-mean-square error) is a good choice due to availability and relatively stable performance it provides [9]. In our case, each BLE node, broadcasts its UUID (universally unique identifier), Bluetooth MAC address, transmit power (txPower) with a frequency of 0.1 Hz and a major-minor pair, which are used to uniquely identify a node. Similarly a UUID is used for uniquely identifying information. It identifies a particular service provided by a bluetooth device. RSSIs are calibrated according averaging for 60 seconds at a distance of 1 meter. Measuring various RSSI's at known distances, we did a best fit curve to match our data points. A node was installed within 1 meter from the mobile smartphone, which was acting as a receiver. The txPower was set to the maximum available by the BLE module and the RSSI of the signal was measured. The measured value was set as the txPower for the node. At this point, for various distances (0.5 m, 1m, 2m, 3m, 10m, 12m and 20m) the RSSI was measured. Over a 20 second average, we applied a filter removing the 10% top and bottom values that resulted a stable trade off. At last, we fitted a best fit curve and obtained the following equation in java code:

```java
double ratio = rssi*1.0/txPower;
distance = (0.89964)*Math.pow(ratio,7.7083) +
    0.112;
```

Each node, operates in an advertisement mode to notify the nearby devices its presence by using advertisement frames. An example advertisement frame could be as following:

```
ch1d64v5-4578-26 bv-327f-85j955go12a6 Major 1
    Minor 2
```

As presented in Figure 1, each mobile device, receives one or more beacons from surrounding nodes. It initiates a session with the server using a secure SSL connection in order to prevent unauthorized access and faking users. Because of the architecture of BLE, nodes could be easily forged by cloning UUIDs, Major/minor and macs. To prevent this, the server keeps a log of the previous node locations sent by each mobile client, in order to detect any anomaly, as duplicates or sudden jumps to locations far apart. At this point, it calculates the estimated distance, and communicates it to the Smart

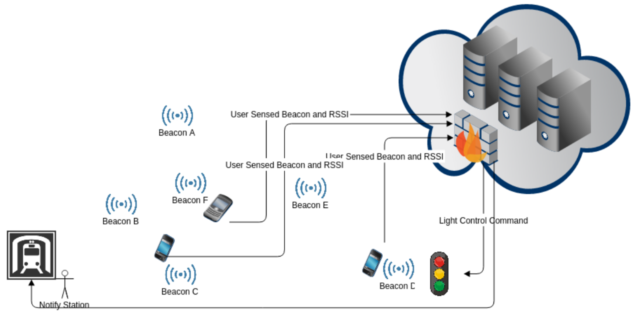

Figure 1. Smart City Framework Architecture

City server, binding it with any available GPS data for more precision and robustness against frame cloning.

IV. Smart City Framework Backbone

The Smart City Framework (SCF), as mentioned before, has the objective of processing, storing and triggering different service actions. It is composed of three separate entities: Event Handler, Condition processor and Action unit as in Figure 3. Events are generally BLE beacons combined with user IDs and optional GSM data packets such as CID (cell-ID), MCC/MNC (mobile country code/mobile network code), LAC (Location area code) network and signal strength and GPS acquired position when available (outdoors). The cell-ID is the unique identification number of the GSM cell tower (aka GSM base station, Base Transceiver Station) the phone is currently connected. MNC is a number used to uniquely identify the GSM network provider in combination with the MCC. MCC is a unique country code of the GSM network the phone is currently connected to. The SCF, after receiving a BLE event, it stores it into its database, which is consequently compared and mapped with nodes available in it, finding the closest nodes and an estimate of the users position through trilateration by using their signal strength as in Figure2.

Through the BLE RSSI, the SCF will be able to categorize the user location within a three distinct ranges:

- Immediate: Within a few centimeters
- Near: Within a couple of meters
- Far: Greater than 10 meters

The SCF has the ability to determine whether a user has entered, exited or remained in a region through RSSI level changes. For example, if a received signal starts to loose power through time (and not peaks), it indicates that a user is getting further from the base station. On the other hand, if a received signal is getting stronger, it indicates that a user is getting close. Power peaks or looses, that are less than the time frame defined by the mobile application, are opted as they could be due to sudden signal absorbment by passing objects, humans, etc. and not a user behavior. Each received beacon is considered separately according to the user which has submitted it and the services subscribed. The SCF will decide whether it needs to invoke an external service (i.e., switching a traffic light at a certain intersection), notify the user about an event or neglect it. External services are available to SCF by mean of a REST (Representational State Transfer) API. In this way, SCF is capable of notifying or

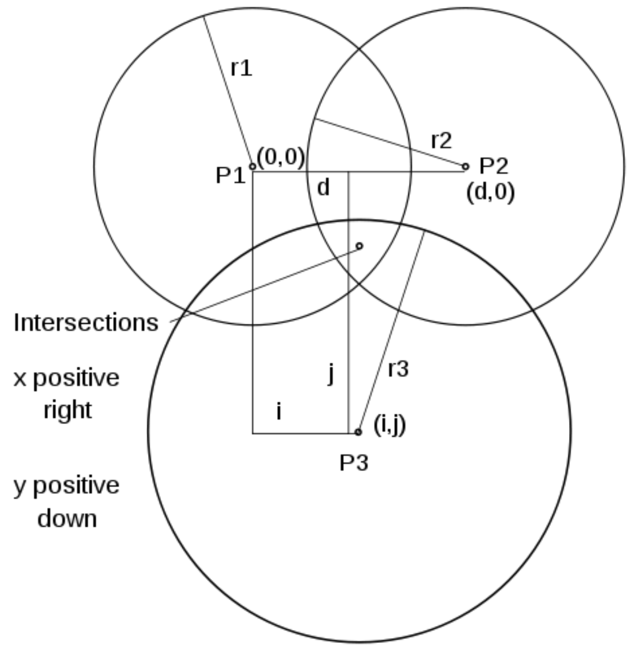

Figure 2. The two intersections of the three sphere surfaces are directly in front and directly behind the point designated intersections in the z = 0 plane.

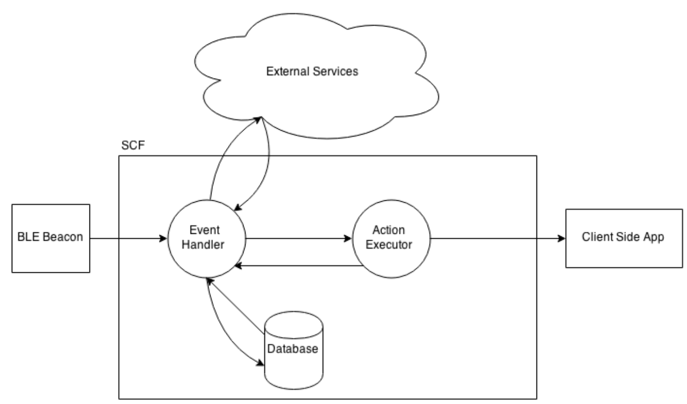

Figure 3. Interactions within SCF

querying services about their current state, change their state and available actions, which are available for a specific user, as sending a push notification or controlling a device.

The system provides a RESTful API, which allows external services, as bus services, metro lines, street lights, hospitals, stores and generally, any city available service to provide their functionalities to the clients. In order to an external service to work with the framework, the framework can provide the following parameters: a user ID, indoor and outdoor location, nearing or distancing state of the user. External APIs could be registered to be triggered according to the desired events by the user. Two different beacon categories are pre registered in the SCF database, Static Beacons and Dynamic Beacons. The Static beacons are used to present data such as location or tags (product tags) and usually are unchanged. Dynamic beacons information in the mapping system, instead could be changed

 ISBN: 978-1-61208-446-6

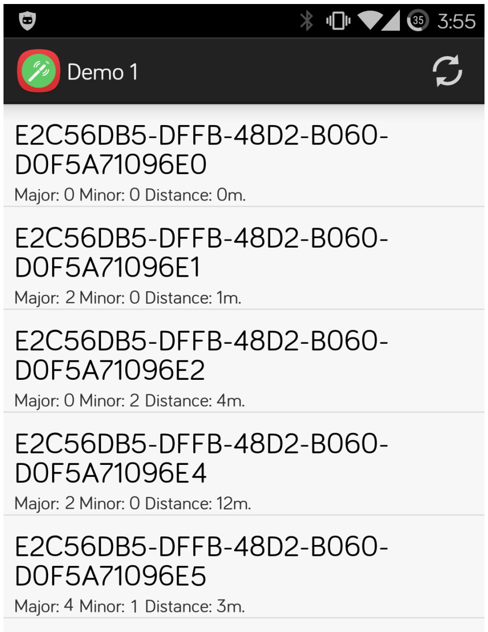

Figure 4. Client Side Application

(manually by service provider/automatically according to pre-defined scheme), which leads to a beacon to provide new data using the same constant BLE data frame.

V. Client Side Application

The client side application was developed using the Android SDK. It constantly scans for BLE advertisement beacons with a scanning window open for 10 seconds each. Proximity for each beacon node is then calculated and it is sent to the remote SCF server for processing along with users authentication keys. If a user is constantly in an environment, while results in no change to the Immediate-Near-Far distant approximation, no data is communicated to the server. The application sends the captured data frames though any available network connection to the the server, whether WiFi, 2G or 3G/4G networks). In the case where the server needs to send a notification to the client app, i.e., an intersection that is safe to cross due to a green traffic light, the mobile app will receive the push message and using android text2speech API, converts it to synthesized human voice. The data frame, which is communicated to the SCF server consists of all the received BLE advertisement packet and optional data, which depending to the use case, could be applied, i.e., cellular IDs or GPS data as mentioned previously in section "Smart City Framework Backbone". A screenshot of the application is seen in Figure 4, which shows the received the beacons, and their distances from the receiver.

VI. Power Consumption

Power consumptions on BLE nodes depends to both advertising frequency and transmit power. A study was done on 16 different BLE module vendors following the iBeacon specification [10]. The report demonstrates that the battery life range could between 1-24 months, as seen in Figure 6.

Also, there are ready to deploy modules with two or three batteries, which their lifespan could range up to 6 years. In our

UUID	MAC	Major
E2C56DB5-DFFB-48D2-B060-D0F5A71096E0	B4:994C4F74A5	0X8810

Minor	RSSI	TX	Optional
0X7401	-77dbm	-59dbm	Optional data

Figure 5. Sample beacon packet received by a client, demonstrating UUIDs, major, minors and clients estimated distance from each

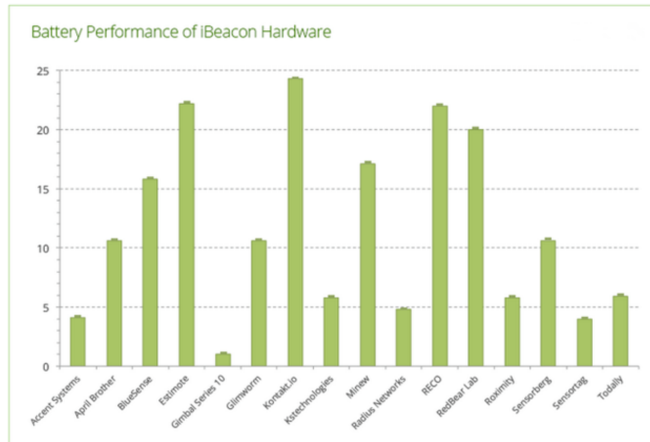

Figure 6. Comparison of 16 major beacon hardware of battery life in months (higher is better) by Aislelabs' The Hitchickers Guide to iBeacon Hardware[11]

Figure 7. BLE node used in this work, a W901 wellcore module

work, during the period of approximately 6 months (163 days), a 17% battery decrease was measured using a third party BLE module provider using a TI(Texas Instruments) cc2541 chip and a CR2477 battery, as seen in Figure 7.

Smartphone power consumption is another aspect which must be mentioned. The application transmits the advertisement frames only when changes are detected, so for example, when a person is in a library, as he/she is constant, no data will be transmitted to the server. Also, in next works, the scanning will take place only when movement is detected by the phones accelerometer module, which draws much less power. During our tests, during a full day of 24 hours, a smartphone (OnePlus One) with a 3100mAh battery, in continuous node scan and sync with server, has used 25% of its battery power for the specific application, measured by the battery manager of Android 5. It can further be improved by implementing a dedicated BLE chip which could scan nodes in deep sleep mode (low frequency).

VII. Experimental Results

To experiment the system structure, a set of 10 BLE nodes were deployed. This number of nodes would be sufficient for trying the system by itself, as the server would be deployed on a distributed cloud, and network traffic would be feasible to handle. All UUIDs, MACs, Major/Minor pairs were inserted into the database. In order to measure the distance with a proper accuracy, each beacon needed to be calibrated prior to use, as mentioned previously. The SCF server was written in Python using Flask micro-framework and later ported to Django for better distributed operation. At this point it was deployed on an Ubuntu Server on a cloud provider, listening to requests sent by the clients. Two test users with smartphones in their pockets (OnePlus One/Motorola Moto G), were allowed to freely move around the area, both indoors and outdoors. According to their movements, events were registered on the server, and connected ZigBee light switches through an external API were automatically turned on and off. At the mean time, a room was labeled as no-enter zone, and registered in the system, as an external service depending solely on location data. While standing by its entrance, a notification was automatically sent to the person alerting them about the prohibition through the vocal interface. The user could call the elevator, enter and get off at the ground floor without any physical interaction with the elevator command buttons, just by entering a zone near to the entrance of it, in order to call it.

VIII. Conclusion

A framework which can interact with users is crucial in order to have a smart city which can provide autonomous interaction and service to its citizens. In this paper, we have proposed and implemented a modular framework backbone, which permits its users to interact with it using their smartphones, and also the possibility of integrating with various services provided by external entities. This can lead to a more autonomous lifestyle for different groups of people living in future smart cities and permitting the governmental organization to automate several aspects of urban activities.

References

[1] H. e. a. Chourabi, “Understanding smart cities: An integrative framework,” in *System Science (HICSS), 2012 45th Hawaii International Conference on*. IEEE, 2012, pp. 2289–2297.

[2] R. Dameri and A. Cocchia, “Smart city and digital city: twenty years of terminology evolution,” in *X Conference of the Italian Chapter of AIS, ITAIS*, 2013, pp. 1–8.

[3] B. Bowerman, J. Braverman, J. Taylor, H. Todosow, and U. Von Wimmersperg, “The vision of a smart city,” in *2nd International Life Extension Technology Workshop, Paris*, 2000, pp. 1–7.

[4] C. e. a. Harrison, “Foundations for smarter cities,” *IBM Journal of Research and Development*, vol. 54, no. 4, pp. 1–16, 2010.

[5] F. Veronese, D. S. Pour, S. Comai, M. Matteucci, and F. Salice, “Method, design and implementation of a self-checking indoor localization system,” in *Ambient Assisted Living and Daily Activities*. Springer, 2014, pp. 187–194.

[6] K. Barzilai-Nahon, “Gatekeeping: A critical review,” *Annual Review of Information Science and Technology*, vol. 43, no. 1, pp. 1–79, 2009.

[7] “Bluetooth Core Specification 4.2,” https://www.bluetooth.org/en-us/specification/adopted-specifications, 2014, [Online; accessed 3-April-2015].

[8] M. B. Jamâa, A. Koubâa, and Y. Kayani, “Easyloc: Rss-based localization made easy,” *Procedia Computer Science*, vol. 10, pp. 1127–1133, 2012.

[9] E. Dahlgren and H. Mahmood, “Evaluation of indoor positioning based on bluetoothr smart technology,” 2014.

[10] J. Khan, “Best iBeacon hardware crowned following extensive stress tests,” http://9to5mac.com/2014/11/03/best-ibeacon-hardware-stress-test-battery-life/, 2014, [Online; accessed 5-April-2015].

[11] A. Inc, “The Hitchhikers Guide to iBeacon Hardware: A Comprehensive Report by Aislelabs (2015),” http://www.aislelabs.com/reports/beacon-guide/, 2015, [Online; accessed 5-April-2015].