Communitary Platform of Sensors

Author name
francisco.monteiro.vilaca.pereira@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2019

Abstract

This master thesis addresses the problem of data collection. Current solutions are implemented using technologies like LoRaWAN and GSM. The problem is that it comes attached with infrastructural costs that can be avoided with the utilization of a technology like Bluetooth Low Energy. This study is important because it evaluates a cheaper alternative for data collection that takes advantage of the people that brings their smartphone everywhere they go. To do this, a prototype system for gathering the sensor data called SmartCommunity was developed. The system consists of four components: the Sensor Nodes, an Android application, a Backend and an Endpoint for publishing the transported data. It was concluded that this can be a viable solution for sensor data monitoring that offers security and overall reliability in the delivery of data from sensors to the cloud with the participation of the smartphone as the bridge between the two, evaluating the possibility to use the application in cases where we are walking, driving an electric scooter or bike and also the power consumption of the smartphone while running the service, as well as the possibility to encrypt sensor data.

1. Introduction

The sensor monitoring architecture is divided into three layers: Sensors, Wireless Connectivity and Data Access. The main problem that this thesis addresses is related with the Wireless Connectivity. There are several options to tackle the connectivity between the sensors and the final destination. What is the most practical and affordable way to collect sensor data? Should it be using GSM? Should it be using LoRaWAN or could it be better to use WIFI? Can a ZigBee and Bluetooth take care of the job?

Nowadays everyone has a smartphone right? Can it be a possible way of transportation between sensors and the Cloud? Maybe Wi-Fi, Bluetooth Low Energy or Zigbee could be possible solutions to tackle the Wireless Connectivity problem.

This study will be focused on evaluating the possibility of gathering sensor data while casually driving an electric bike or a scooter. This evaluation is going to be important to validate a scenario where the developed application prototype could be piggybacked on an Application like Gira.

The proposed idea for this thesis is to design a monitoring system that should provide an alternative way for relaying information from sensors to the Cloud, in order to avoid costs that are inherent to solutions based on technologies like LoRaWAN and GSM [13]. The proposed way of transportation between the sensors and the Cloud is the smartphone.

The motivation for doing this research is to explore a new way for sensor monitoring based on the existence of a transport that should link each sensor to the Cloud.

The purpose for doing this thesis will be fulfilled by answering to this questions:

1. Is it possible to use people’s smartphone as
way of transport between the sensors and the Internet?

2. Can I capture sensor data while walking, cycling or even driving?

3. Can the information be encrypted?

4. Can power consumption be an issue?

By evaluating these research questions, this thesis has the goal of serving as a starting point for business value creation for the context of Smart Cities.

2. Related Work

Several studies were analyzed with the purpose of understanding how sensor monitoring systems are designed. Neither of them addressed the problematic of using a smartphone user as a way of transport between sensors and the cloud but they allowed me to understand how existing systems are implemented and what technologies are being used.

The first study shows how a classic Wireless Sensor Network is typically structured. The sensor nodes connect directly to the gateways using a short range communication channel, and each gateway autonomously connects to the server. The gateways process, store, and periodically send the field data to the application server using long-range communication channels. The application server provides long-term data storage, and interfaces for data access and process by end users (either human or other applications) [16].

The second study is important to my thesis due to the fact that the proposed idea is somehow similar to the one I propose with the slight difference that the sensors are not maintained in fixed positions in specific locations. The sensor is transported by the person, like the smartphone. A system demonstrator for environmental monitoring based on participatory sensing. Mobile environmental sensors carried by citizens are used to measure pollutant concentrations and provide this data to a Unified Sensing Platform [14]. The sensor used is a battery and solar cell powered mobile hand held device that is designed to be used by pedestrians and bicyclists. The microprocessor is connected to sensor components to detect pollutants. Measurements are taken and transmitted to a smartphone that have the Environmental Sensing App installed, via Bluetooth. The Environmental Sensing app was developed to receive measurements from the sensor device, showing measurements to the user and transmitting the information to the USP.

The third one is based on a beacon that is connected to Firebase taking advantage of a GSM module. The readings are backed up on the device under an encoded JSON file, this is how they choose to persist data[17].

The fourth one is based on the existence of a service that acts as MQTT Broker that is responsible for distributing messages to connected clients. This requires the use of a Wi-fi Module and the existence of an Access Point[15].

The fifth study describes the design of a prototype based on IoT premises for real time monitoring of environmental conditions using low cost sensors. The captured data is broadcasted through internet with the Wi-fi module from ESP8266. It is delivered to an API called Thingspeak over an HTTP protocol[12].

This chapter also addressed what should be the important factors to have in mind in order to choose the right microcontroller for the desired job. Things like energy efficiency, cost per unit, development kit, documentation and being easy to get are all important when it is time to choose which to use [11].

Choosing the right wireless technology is also important when developing a system for data collection. As many of the devices are small low-cost devices, the radio must not add too much additional cost, taking into consideration the fact that the radio and device applications in many cases need to share the same computing engine (microcontroller). Many use cases require a battery or some kind of energy harvesting technology as a power source. It must be easy to associate a device with the network and with the Internet Services. The authentication and the encryption must be adequately supported by the wireless technology. It needs to have the possibility to connect to smart phones. It is necessary to have the capability to cover enough of a range or to have some capabilities to extend the coverage without having too big an impact on the system cost. Zigbee, Wi-fi and Bluetooth Low Energy were the tech-
Technologies investigated. The studies show that the three technologies have built-in link layer authentication and encryption, which sometimes needs to be completed with end-to-end security from the sensor to the web application. Bluetooth Low Energy has the potential for less power consumption than IEEE 802.15.4. The lack of native support for IEEE 802.15.4 in mobile devices is a problem, especially for mobile or temporary mobile use cases. BLE and IEEE 802.11 devices are easy to associate with a network and mainly with Internet services, while this is often more difficult with IEEE 802.15.4 devices[1].

3. Proposed Solution
This chapter describes the solution that was proposed in order to tackle the problem of using the smartphone as a transport between sensor nodes and the cloud, using a low range wireless technology, guaranteeing privacy and avoiding spoofing of sensor nodes in order to provide a reliable way for collecting sensor data in the context of a Smart City.

The proposed scenario is based on the assumption that is possible to use the citizens smartphone to relay information to the Cloud. The idea is to collect data from Sensor Nodes that are distributed in specific places and that are periodically advertising data that they collected. The information transmitted from the sensor node to the smartphone should be encrypted. The data collected by each smartphone should be saved in a local database in order to be later relayed to the cloud when there is Wi-fi connection available. This is the typical scenario that drove this study.

Every Sensor Node has the same data structure. It is based on the Bluetooth Low Energy specification. Figure 6 shows how data is organized inside each device.

3.0.1 Algorithm
Each Sensor Node has a loop running, from time to time, that makes it collect data and later advertise it. It also has times when it isn’t doing anything to in order to lower power consumption.

The main loop is represented above. Initially the Sensor Node verifies if the difference between the current date and the initial data collect date was already reached. If the desired value was already reached, the initial date is refreshed.

After that, the micro controller asks the sensor to make data collections. Each collected value is written in persistence memory to avoid data loss if something happens to the Sensor Node.
The condition for stop advertising is represented above. It is based on the verification of the actual situation of the Sensor Node. It basically verifies if it is advertising and if the time interval between the current date and the date of the beginning of the advertisement has already been reached. The time interval is represented like that because for evaluation purposes it was a small one. During the actual scenario it is expected to be much bigger in order to make only advertise several times a day.

The Transporter Node is defined by each smartphone that has the job of collecting data that is being provided by each Sensor Node.

This operation is represented by a service that runs in the background and that, when it is triggered, initializes the Bluetooth Adapter. Then it gets the information from the Sensor Nodes that are saved in the Backend and then it starts scanning for those specific devices.

When the Transporter Node detects a device that is listed in the system it initiates a connection with him using the technology Bluetooth Low Energy. It creates a callback that later will be called, when the connection is successful established.

When a characteristic is successfully read, the Transporter Node proceeds to get it’s value saving it into the an SQLite database, together with the description of the value, the name of the transmitting device and with the specific timestamp. The description can be “Humidity” or “Temperature” for example. When the information is successfully persisted in local storage the android application sends a toast to notify the user if he happens to be with the smartphone screen on. Note that the toast
The values that are locally stored in the SQLite Database are all still encrypted. In order to decrypt them, and also, in order to send them to the endpoint in the Cloud there is a need to have Internet connection.

So, the designed solution has a Broadcast Receiver [2], that listens to changes in the Wi-Fi connection. When it is triggered, the local database is queried in order to get all the collected data. For each row in the table, the application calls a Cloud Function that is responsible for decrypting each value. This design choice was made based on the premise that sensitive data should not be exposed to the Transporter Node.

For each Cloud Function call, there is also another call to the Firestore to publish the values that were stored in the local DB with each value already in plaintext.

The Sensor Node software was developed with the help of PlatformIO. The application that should run on the Transporter Node was developed with Android Studio. The Backend, as well as the Endpoint, were developed using the Cloud Firestore provided by Google.

### 3.1. Stack of Technologies

The ESP32 was the microcontroller used to represent the Sensor Node in this study. The reasons for this decision are mostly based on the low price, Bluetooth and Wi-fi technology support, power saving features and the wide developer community associated with this particular device.

Huawei P8 Lite was the smartphone model used during this research. The main reason that led to this option was the fact that this was my personal phone.

Here you can see the specifications of this smartphone. Important aspects are the version of the Bluetooth, 4.1, and information related with the

<table>
<thead>
<tr>
<th>Device</th>
<th>Description</th>
<th>Device</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Node</td>
<td>Transporter Node</td>
<td>Backend</td>
<td>Endpoint</td>
</tr>
</tbody>
</table>

![Stack of Technologies](image)

![How collected data is saved](image)

![Backend Data Structure](image)

![SQLite data structure](image)

![WiFi Connection Scenario](image)

![Cloud Interaction](image)

![How collected data is saved](image)
Battery. The information about the battery is relevant because it will be used as reference point to the power consumption test. The tests were made with this smartphone but, with a Rule of Three, is possible to estimate a possible decrease in the battery percentage of other type of smartphone.

4. Results & discussion

A test with two sensors was executed in order to validate the scenario where there are more than 1 sensor transmitting information. The results of the test were the proper collection of data from each of the transmitting sensors. The mobile phone connects to the first sensor it discovers.

To understand the time that takes for the smartphone, when inside the range of the sensor, to successfully collect data, values gathered from logs were analyzed in order to understand the tendency. With the help of Android Studio Logcat [3] it is possible to see the time each value takes to be collected from the time the scan starts. To understand the time that takes for the smartphone, when inside the range of the sensor, to successfully collect data, values gathered from logs were analyzed in order to understand the tendency. Figure 22 shows a graphic that has 20 consecutive readings. The actual values are in the Appendix A. For the further analysis I will use approximate values to better explain the tendency.

By analyzing the graphic is possible to see the tendency is to take approximately 2000ms to get the data about the first value collected. The second value discovered takes another 2000ms to be collected, thus the 4000ms that appear at each two consecutive readings. The values that are here shown represent collections of data about two values, humidity and temperature. If we add three values to read, for example, humidity, temperature and CO2 levels, it would be 2000ms for each collected value and thus 6000ms to collect all the 3 values, assuming that the pattern maintains. The eleventh measure dictates that the sensor stopped advertising. In the fifteenth measure I turned the sensor off and turned it on a little bit later. From 16 to 20 it is possible to see that the pattern it’s still the same.

The range test is important to consolidate the conclusions from the previous test section. It was validated a range of approximately 50m. This mea-
surement includes walls so the scenario of indoor monitoring can also be validated.

The equation that allows the calculation of the maximum supported velocity based on the length of each rope is represented below.

The values used are the average scan time for the first and the second reading respectively. For \( r=50\,\text{m} \) and \( t=2.2\,\text{s} \), the relation between \( v \) and \( d \) can be seen by analyzing Figure 23. For \( r=50\,\text{m} \) and \( t=4.6\,\text{s} \), the relation between \( v \) and \( d \) can be seen by analyzing Figure 24.

The following images show the difference in consumption with 4G turned on and turned off in three difference cases, during the period of 1 hour. The first graphic can be used as a reference point in order to understand the normal decrease in percentage of the battery of my smartphone. The other two can then be compared with the first one in order to be possible to make conclusions.

The first one, Figure 26 is with the screen locked and without running the application. It shows that there was a decrease in the battery levels in 3 percent with 4G and 2 percent without 4G.

The second graphic, Figure 27 shows the decrease in battery life when the app is running but without the sensor running. So the app is not collecting any values. With 4G turned on there is a decrease in 4 percent and with 4G turned off there is a decrease of 2 percent.

The second graphic, Figure 28 shows the decrease in battery life when the app is running but without the sensor running. So the app is not collecting any values. With 4G turned on there is a decrease in 4 percent and with 4G turned off there is a decrease of 2 percent.

The two graphics shows that it is possible to still collect data if our velocity is between 25 m/s and 10 m/s. According to HealthLine the average walking speed is between 0.94 and 1.34 m/s [5]. According to Lime Wikipedia all of the scooter models that Lime uses can reach speeds of up to 25 km/h or 6.94 m/s.

According to [10] the maximum speed of a Gira electric bike is also up to 25 km/h or 6.94 m/s. All of these values are lower than the maximum supported for the average scan time assuming a range of 50m and a maximum distance from the sensor of 40m. It shows that in these particular cases is possible to collect data.

With two Huawei P8 Lite near the sensor node, each one of them was able to collect data with a difference of approximately 4 seconds between each device. The results show that the smartphone that is closer to the sensor node is able to receive the results beforehand.

GSM needs to know the number of the receiving phone in order so send an SMS with the data [8] [7]. This is something that we should avoid because this would require to have all the registred smartphones number in each sensor which is not a scalable solution. Another disadvantage is the fact that the there is a need for having a SIM900 GSM GPRS Shield, for example in the case of Arduino in order to be able to communicate with smartphones.
and that comes with a price [6]. Adding the price of the Arduino [9]

LoRaWAN architecture, as seen in the following figure, requires the existence of LoRa gateways, that are not cheap [4].

Each ESP32 comes equipped with BLE and Wifi, and all I had to spend were 11 euros. This is clearly an advantage in terms of cost comparing with the other two technologies. And it is also something practical, due to the fact that there isn’t a need for extra hardwere, the ESP32 takes care of it.

5. Conclusions

To tackle the problem of sensor data collection in big cities, and also in indoor locations, it was proposed an idea based on the use of Bluetooth Low Energy as the wireless technology that bridged the sensor data with the cloud. The designed prototype protocol is based on the occurrence of short connections between a smartphone and a microcontroller. The data transmitted from each Sensor Node is not sent in plaintext, it is encrypted with a key that should be unique for each device. This specific characteristic allows the collection of data in movement.

According to the evaluation it is reasonable to conclude that the energy consumption of the developed service is not too impacting on the battery life of the smartphone. This answers to the research question related with power consumption The results showed that we can collect data while walking, driving an electric scooter or driving an electric bike. If a car drives around the 40km/h mark with a sensor at a distance of approximately 40 meters it is possible to collect data.

This answers to the research question related to the situations where we can collect data. The information that sensor transmits to the smartphone is encrypted so we can also validate the question about securing data, even though it lacks the key management feature that will be left to future work. So answering to the first research question, yes, it looks like it is possible to use a smartphone as a way of transportation between sensors and the cloud. There are some questions that will need a more detailed perspective like evaluating the possibility of adding more services to each sensor to see how much time is needed in order to collect all services data.

Associated with an application like Gira, the usage scenario would be triggering the background service in order to start scanning when we unlocked the bike. The app, in the future, could tells us a route that includes sensors in order to make an incentive to the client to follow a specific path in order to collect sensor data and receive bonus time to use bikes.

The distribution and management of the keys is a theme that should be explored in the future, in order to increase the level of security of the advertised data.

To finalize, I hope the Smart City concept doesn’t fade leaving so much potential behind and that the design of future solutions have in consideration the privacy of the people that may want to be part of information relay.

References


