Simulated LoRa Sensor Network as support for Route planning in solid waste collection∗

Montañez Gomez, Miguel Angel  
 computer science & Industrial faculty   
 Universidad Nacional de Colombia  
 Bogotá  
 mmontanezg@unal.edu.co

Niño Vasquez, Luis Fernando  
 computer science & Industrial faculty   
 Universidad Nacional de Colombia  
 Bogotá  
 lfninov@unal.edu.co

ABSTRACT

Mass production and population growth in urban settlements has produced an increase in generation of solid waste, as consequence efficient treatment of waste has become a challenge for cleaning enterprises, possibly because they continue performing collecting operations using fixed periodic routes that exhaustively go across neighbourhood streets in search of every dumpster, further in this operations recyclable material is not separated from disposable one at least in Bogota, causing a negative impact to the environment. This work aims to prototype a sensing system that generates routes based on actual fulfilment level reported by dumpsters, to accomplish so dumpsters were equipped with a level measurement device based on a proximity sensor, information gets transmitted using an Long Range network (LoRa), collected data was used to determine which of the dumpster were required to be collected and which were not for the route being planned, according to fulfilments levels reported. Since mostly an operation requires many collecting trucks, a K-means method was use to group dumpsters that were geographically close, collecting sequence was calculated using an open Web service, results are shown on Android mobile application, the mobile app consumes Google Maps routing service, this demonstrates technical viability for implementing this sorts of systems in real operation environments, a single district of Bogota was selected as study Area.

KEYWORDS

IoT, Urban Waste Collection, wireless sensor network, LoRa, sensing system, routing, geographic information, mobile application**.**

Cite this Article::

MONTAÑEZ G, M, . NIÑO V. L. 2021. Sensing system prototype base on a simulated Wireless Sensor Network as support for Route planning solid waste collection: In *Proceedings of ACM Woodstock conference (WOODSTOCK’18). ACM, New York, NY, USA, 10 pages.* <https://doi.org/10.1145/1234567890>

1 Introduction

The updated template, user manuals, samples, and required fonts, all are available at the URL <https://www.acm.org/publications/proceedings-template>. This URL also contains some useful video links, which describe how to add the template, structure the paper, and generate the layout, in different clips. **Display Formula with Number**

 (1)

**Continuation part of Paragraph Text** The user must style this paragraph in **ParaContinue** style, which follows immediately after the **DisplayFormula** (numbered equation). The **DisplayFormula** style is applied only in case of a numbered equation. A numbered equation always has a number to its right. Insert paragraph text here. **Display Formula without Number**

Figure 1: Figure Caption and Image above the caption [In draft mode, Image will not appear on the screen]

The **Research question: ¿**How can we structure the logistic process of RSU collection according to the necessities of a single district of Bogotá, to achieve a more efficient collection operation that reduces container saturation and overflow?

Efficiency is measured with an optimization function that aims to collect more RSU in Kg reducing the travel distance in Km and reducing number of trucks. It is modeled in formula 1

(1)

Where:

i: is a container that needs to be served

j: is the route that truck j must travel to collect assigned containers

k: number of garbage trucks required to complete the operation.

To answer this question we will review related work in order to understand and compare the ways collection of RSU is being handle.

Select the tools and technologies that may lead us to find an efficient solution.

Design and implement a system prototype that fits the conditions in selected area, finally execute some test to validate the proposed system viability.

2 System Overview and Architecture

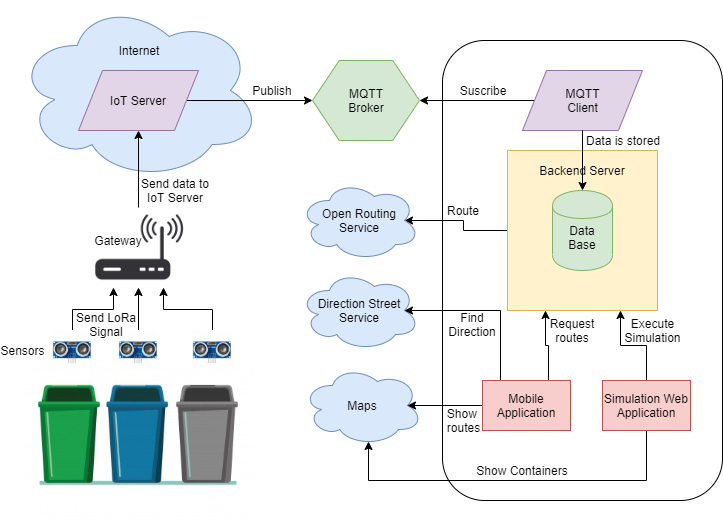


Figure 1: General Configuration and communication of system components

The proposed system is arranged as show in figure 1, as the level of RSU increases, the fulfillment level is sensed in each container. Containers are indexed in a database with a unique identifier and geographical location (Lat, Lon) in such a way that as data is received, it is possible to know almost the exact locations of the container whose level is being reported. This data is transmitted using LoRa modulation, LoRa Signal is decoded and repeated to internet by the gateway, where it gets routed to an IoT Server where data is managed, then an application manager publish this data to a topic of an MQTT broker.

Once the information is persisted in the broker, it can be query via an MQTT client running in a Backend server that subscribes to the topic and retrieves data, this data is stored in the database so the applications can access it when required. In order to search for routes with significant information a simulation was required because there is a limited amount of measurement devices for the selected area. Simulation is explained in section 5. Simulated data is used to plan daily operation, simply put, a priority is given to the streets base on the level reported by the containers located in that street using rules explained in section 6.

Having in to account the estimated volume of RSU to be collected and the number of trucks required to pick dumpsters content, a clustering method is executed in order to group streets which are geographically near, so they have any kind of order to reduce computing time in routing operations. Routing was managed using an Open Web Service, the response was stored in the database to have it ready and available when needed and avoid reprocessing and delays each time the same route was queried.

Calculated routes are displayed using an App Mobile, the App demands generated routes to the backend server, and then it shows the streets the trucks should go through and the locations of dumpsters that need to be served. A map provider was integrated so the information of routes can be easily interpreted, it also uses a direction service that shows trucks the path trying to avoid the trucks to turn around or be stuck in traffic jams.

**3. Methods**

3.1. Hardware Integration

Measuring the level is possible using a weight sensor however there are some discussions as depending of the material density the container might report high weight without being full, or vice versa. For that reason, we decided to use a distance sensor HC-SR04. The operation principle behind this sensor is to emit an ultrasonic signal, the signal travels in the air until it collides to the closest obstacle then the signal is reflected and the sensors receives the signal back. Distance is estimated measuring the delay between the signal emission and it is acquire, with following formula.

The sensor returns back the time and we know that sounds propagates at 334 m/s in the air this is to say 0,0334 m/µs it is split by 2 as the signal spends time to go and return. Then, we can calculate distance.

For simplicity, hardware was assembled using development boards integrating analogical and digital pins, for conceptual reasons three configurations for measure units were explored and analyzed, all these can be abstracted as described in Figure 2.



Figure 2: Block diagram Measure unit components

The first configuration was made on the top of an Arduino Mega Board with a LoRa Shield created by Dragino both devices where connected according to the pins mapping and digital pins 30 and 31 were used to connect proximity sensor as showed in Figure 3, table 1 states pin interfacing between HC-SR04 and Arduino.

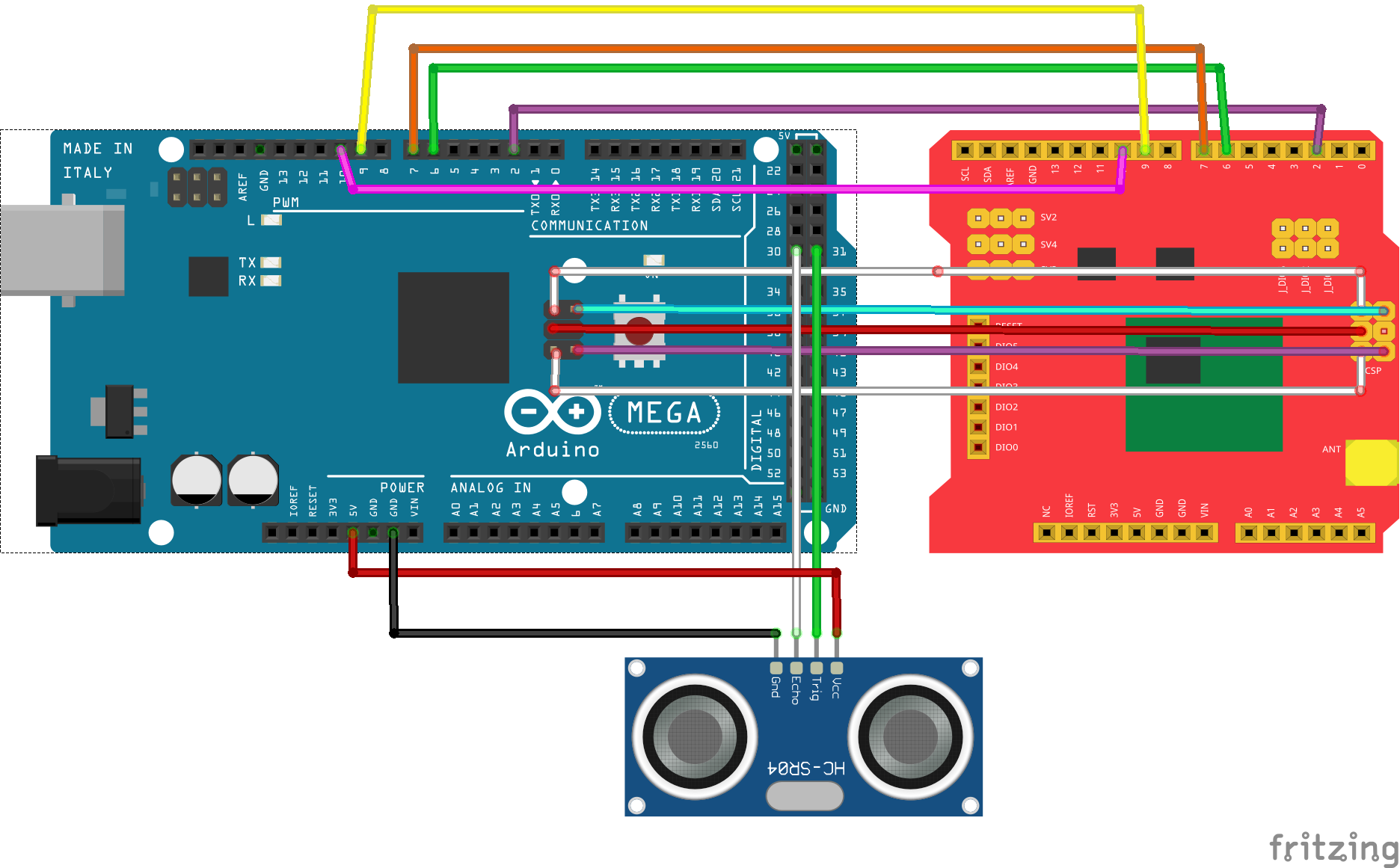


Figure 3: Connection diagram for Arduino and Dragino Shield

|  |  |
| --- | --- |
| **Arduino** | **HC-SR04** |
| 5 V | 5 V |
| GND | GND |
| D30 | Echo |
| D31 | Trigger |

Table 1: Arduino to HC-SR04 pin mapping

The next circuit was set using a board that integrates LoRa transmission module, which simplifies the connection of component as the only component that needs to be added to the board was the proximity sensor, as show in figure 4 table 2 states pin interfacing between HC-SR04 and TTGO.

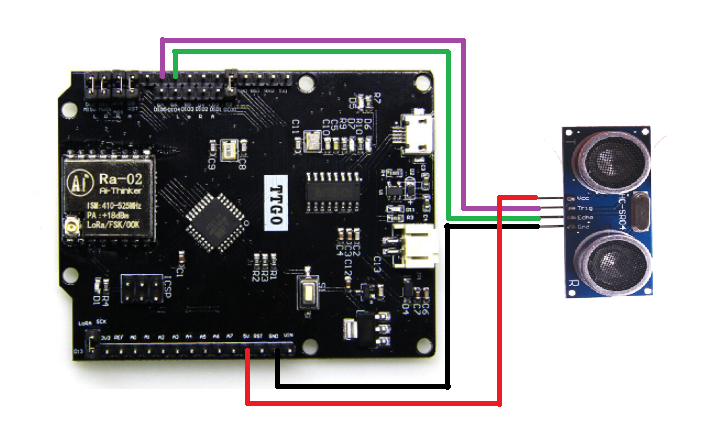


Figure 4: Connection diagram for TTGO and HC-SR04

|  |  |
| --- | --- |
| **TTGO** | **HC-SR04** |
| 5 V | 5 V |
| GND | GND |
| D6 | Echo |
| D7 | Trigger |

Table 2: TTGO to HC-SR04 pin mapping

The third circuit has a small difference compared to the circuits presented above as this one integrated a GPS module, in such a way that geographical location can be obtained precisely. this may help to be more accurate when planning the route, however this have significant impact in energy efficiency and in the airtime; which is the time a message takes to be transmitted, which is an important factor in LoRaWAN protocol. Apart from that as the container locations remains almost constant, updating the location each time a measure is taken is not recommended the circuit was assembled as shown in figure 5.

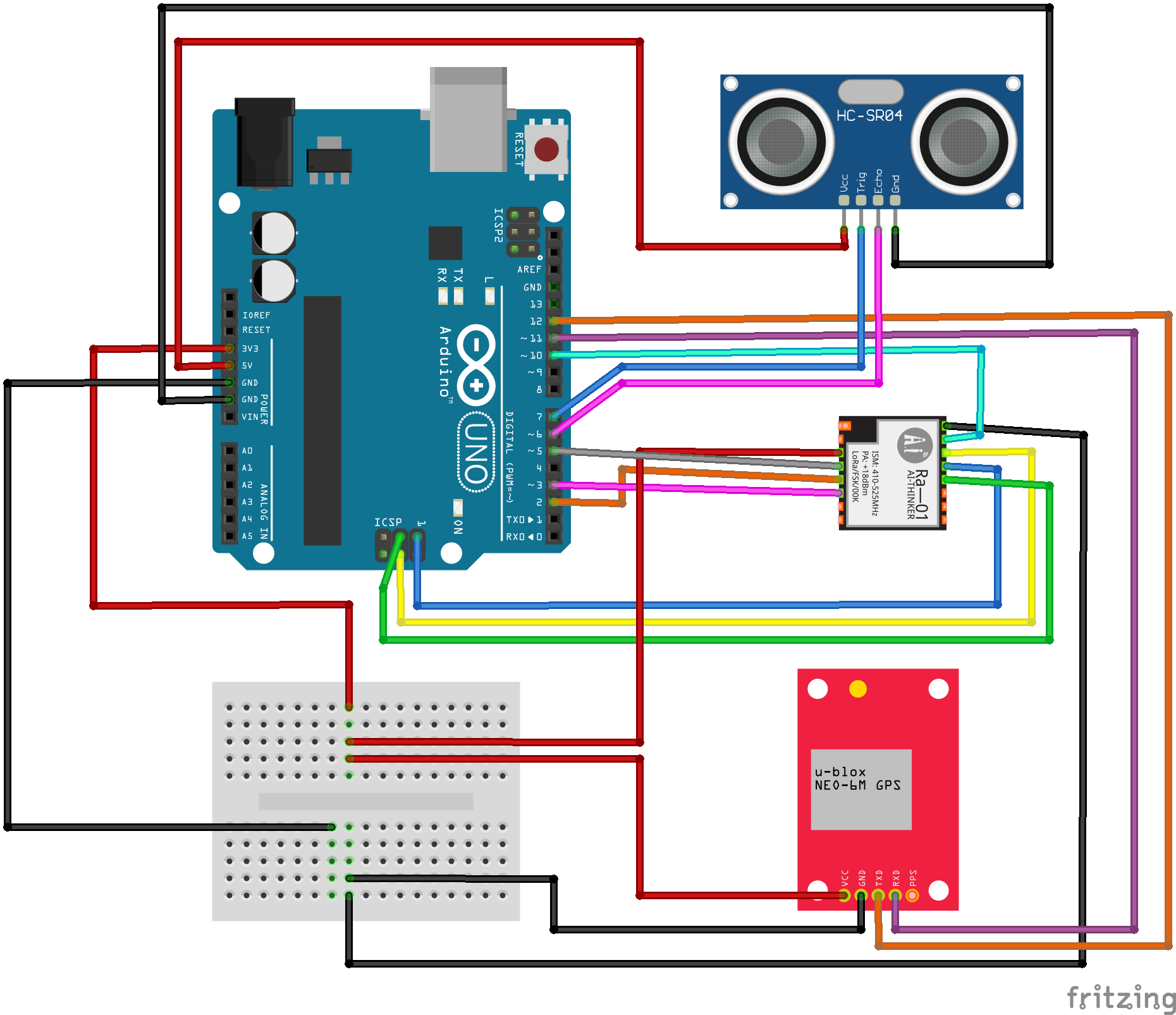


Figure 5: Connection diagram for Arduino, SX1276 neo 6M y HC-SR04.

|  |  |
| --- | --- |
| **Arduino** | **SX1276** |
| 3,3 V | 3,3 V |
| GND | GND |
| DI0 | D2 |
| DI1 | D3 |
| MISO | MISO |
| MOSI | MOSI |
| SCK | SCK |
| D10 | NSS |
| D5 | Reset |

Table 3: Arduino to SX1276 pin mapping

|  |  |
| --- | --- |
| **Arduino** | **Neo-6M** |
| 3,3 V | 3,3 V |
| GND | GND |
| D11 | RX |
| D12 | TX |

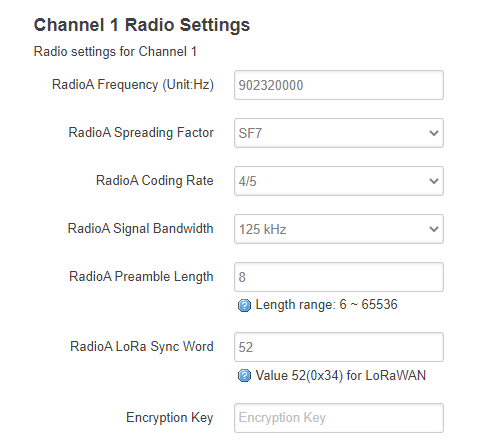
Table 4: Arduino to Neo-6M pin mapping

Connection between the arduino UNO and HC- SR04 are de same that stands in table 2. The controllers were programmed using a LoRa Library for arduino written by Matthijs kooijman [2] the library implements a listener method which is responsible to establish a communication session with the IoT server after joining request gets accepted, the distance measured by the sensor can be encoded and transmitted.

3.2. Data collection and transmission

Now that data has being capture and transmitted by measure units, we expect to obtain this data remote so it is also necessary to configure the gateway, the server and the broker.

We found commercial gateway that can fetch data being transmitted by measure units, the selected reference was LG02 from Dragino, this gateway offers a User Interface UI to set up all transitions parameters among them, Spreading Factor SF, band width BW, Frequency, coding rate and IoT server provider, the parameters selected for our reception channel are the ones show in figure 6. This gateway has 2 communication channels to operate in full duplex mode[[1]](#footnote-2).



**Figure 6:** Gateway settings for reception channel.

|  |  |
| --- | --- |
| **Channel 1** | **Reception** |
| Frequency | 902320000 |
| Spreading Factor | 7 |
| Coding Rate | 4/5 |
| signal Bandwidth | 125 khz |
| Preamble Length | 8 |
| Lora Sync Word | 52 |

**Table 5:** parameter for reception channel

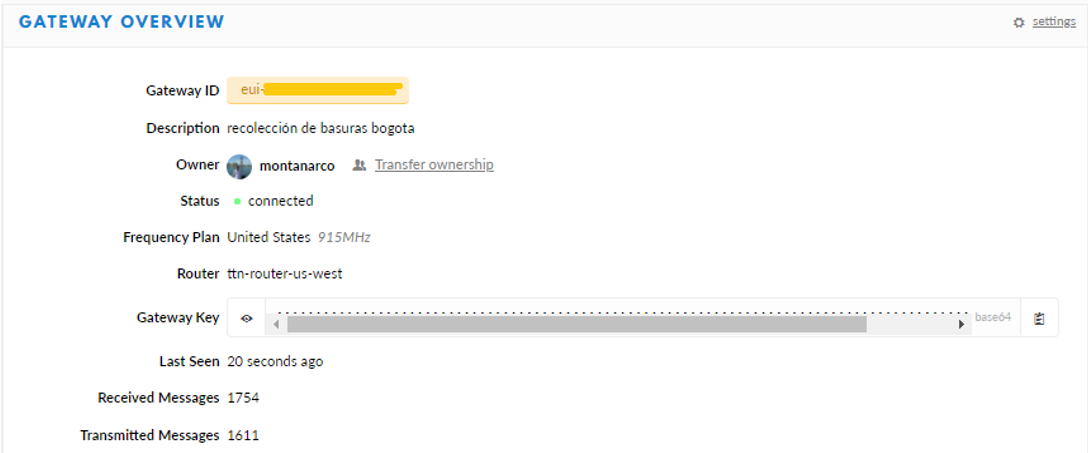
|  |  |
| --- | --- |
| **Channel 1** | **Transmission** |
| Frequency | 923300000 |
| Spreading Factor | 7 |
| Coding Rate | 4/5 |
| signal Bandwidth | 125 khz |
| Preamble Length | 8 |
| Lora Sync Word | 60 |

**Table 6**: parameter for transmission channel

When all the configuration was completed the gateway was started in listening mode, and connected to the network using the Ethernet interface.

For the IoT server we decide to use the things Network[[2]](#footnote-3) because it supports LoRaWAN protocol Operation e.g. activation by personalization ABP and Over the Air Activation OTAA as authentication methods, it simplify rules for the information being collected by nodes, such as data decoding, validation and post processing.

The very first step to use the server is to register the gateway in the administration console, there some information such as identification code, frequency plan and the type of antenna it has attached is required, after the gateway is configured it is show in the console see figure 7. Notice that when the Server established connection with the gateway the status is set to connected.



**Figure 7**: Gateway registry on TTN.

Other thing that was configured in the IoT server was the application, an application has 2 purposes first is to associate all the measure nodes that will be part of the network and second to define general rules and procedures that will be applied to the information reported by the nodes associated to the application, so that information can be manage in a generalized way e.g. decode functions, and integrations.

In our case the information captured by measure nodes is encoded in an array of 8 bytes byte 0 to 2 were used to encode latitude, bytes 3 to 5 were used to encode longitude and bytes 6 and 7 were used to encode fill level so we use a decode function that transforms this 64 base byte array into a JSON format which is more compressible for users:

function Decoder (bytes, port) {

var distance = ((bytes[6] << 8) | bytes[7])/10;

var lat = (bytes[0] | bytes[1]<<8 | bytes[2]<<16 | (bytes[2] & 0x80 ? 0xFF<<24 : 0)) / 100000;

var lng = (bytes[3] | bytes[4]<<8 | bytes[5]<<16 | (bytes[5] & 0x80 ? 0xFF<<24 : 0)) / 100000;

return {

distance: distance,

location: { lat: lat, lng: lng }

}

}

**Figure 8**: JavaScript code snippet for decode function in TTN

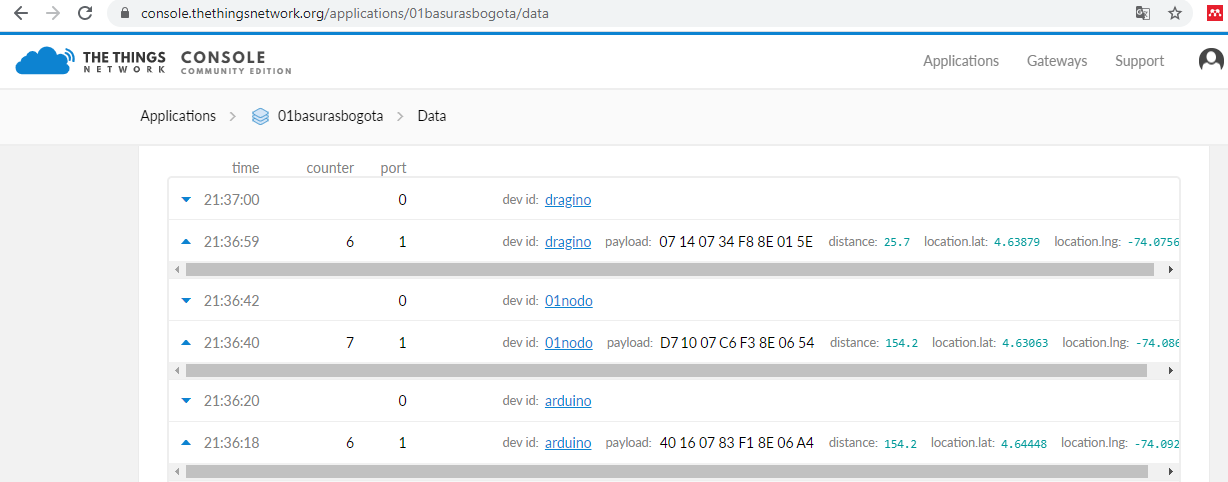
Other important thing to configure is the integrations this defines how to process data after it is received and validated in the server, we decide to use a MQTT broker because the server implements a Mosquito[[3]](#footnote-4) instance and create a topic with the name of the application configured, this way whenever new data coming from measure units reaches the server and it is validated according to the validate and decode functions the decoded payload is publish in the topic in the broker. As soon as the IoT server was configured, it was started along with the gateway so both remain up and running.

Following step was to extract data from the broker thus an MQTT client was implemented and ran in the backend server, we used a java implementation offered by TTN the repository is open and it is available in Github[[4]](#footnote-5) the code has a listener that continuously demands for information to the broker, when new information is obtained we use a function that process these data, the function search the id code of the dumpster that sends the information, then that data gets stored in a postgres data base.

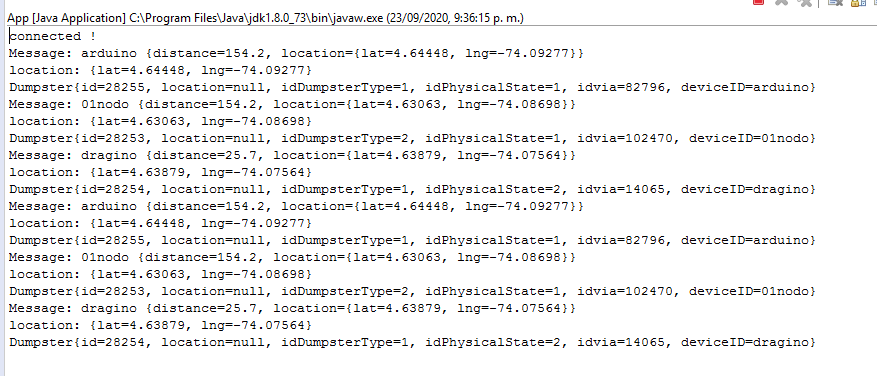
The whole data flow once integration of components was completed is show in figures 9, 10, 11 and 12



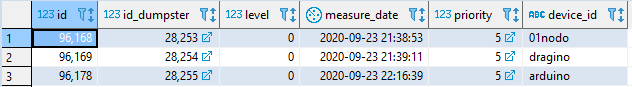
**Figure 9**: log of messages passing through the gateway.



**Figure 10**: messages received and validated in the IoT server.

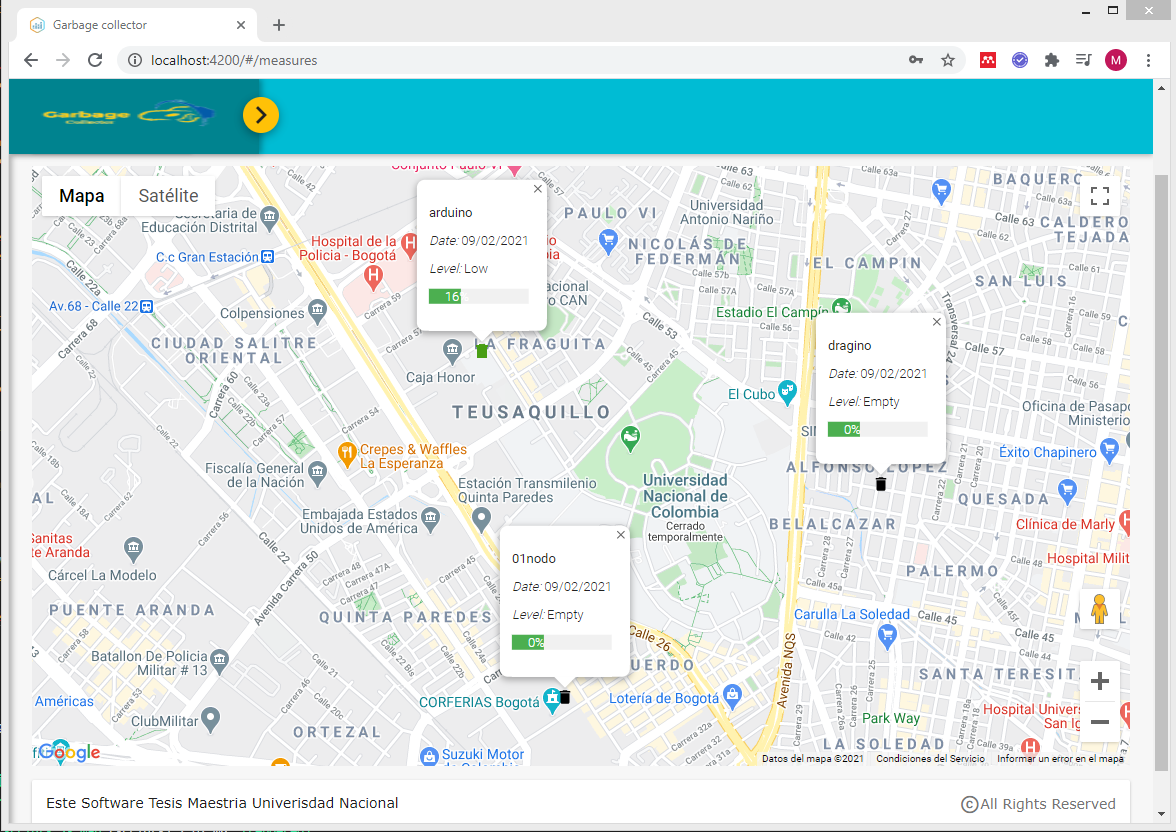


**Figure 11**: data collected by the MQTT client subscribed to the TTN application topic.



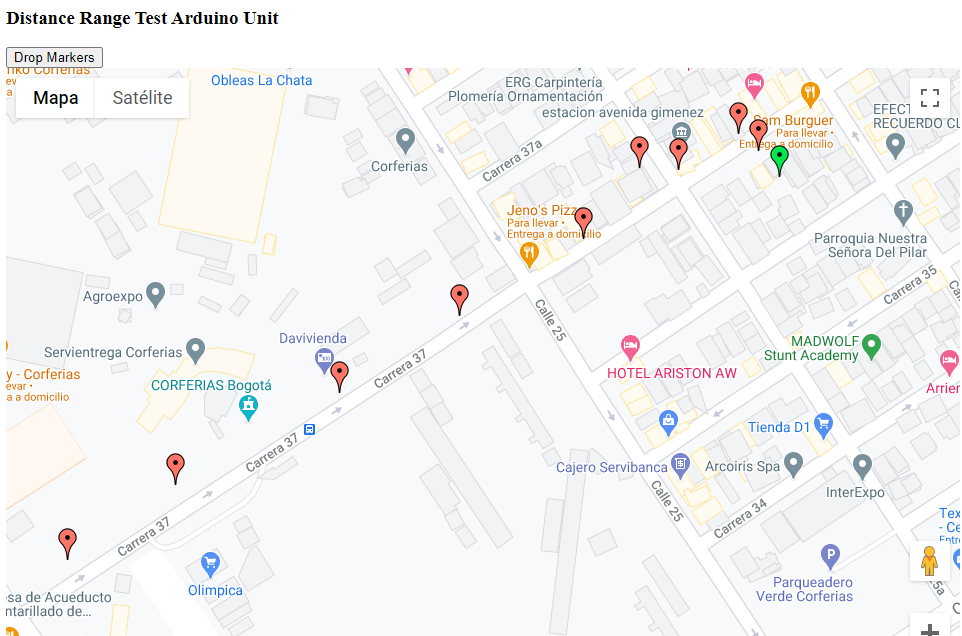
**Figure 12**: data storage in database after being retrieved by MQTT client.

The data remains available so when any application needs it, it can be directly query to the database. A simple web application was implements to visualize the data collected by measure units see figure 13.



**Figure 13**: web application displays data reported by measure units.

The first result to highlight was that information flows along LoRa network as expected; we carry on the research executing a couple of tests to verify the real range of distance reached by transmission modules we selected. So started the gateway and place it in a fixed location, then devices were configured to send a message each minute, these data was stored in the database then data was represented in a map and longest reached distance was measured. First test results are showed in figure 14



**Figure 14**: Results of Arduino unit sending data.

green marker represents gateway location and red markers are location retrieved by the GPS in the measure unit, sended and stored in the database, we select green marker and the furthest red market locations and use Google maps to measure distance between them, the longest distance was about 460 meters. This test lead us to realize that signal did not reached the gateway when there was an obstacle e.g. a wall that interrupts direct sight line, so we conclude that to have an effective transmission there must be a direct line of sight between both devices. Hence for the second test, we decide to find a high-level location to place the gateway then we start walking away from the gateway in a low-density building area so the direct line of sight was easier to achieve.

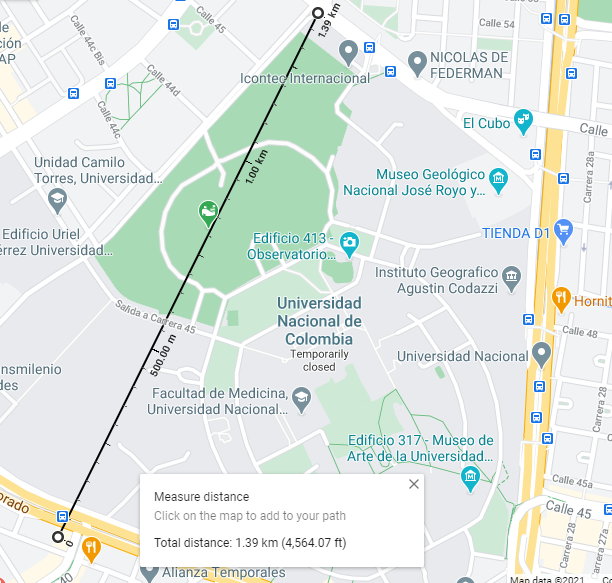


**Figure 15**: Dragino Unit executing data transmission test.

Figure 15 shows Dragino device used to execute the test; gateway was located in the 13th floor of the building show in the upper right hand side of the picture. These test showed longer distances than the first one 1.39 km to be precise (see figure 17).



**Figure 16**: Results of Dragino unit sending data.



**Figure 17**: measurement of maximum range reached with Dragino unit.

3.3. Data Simulation

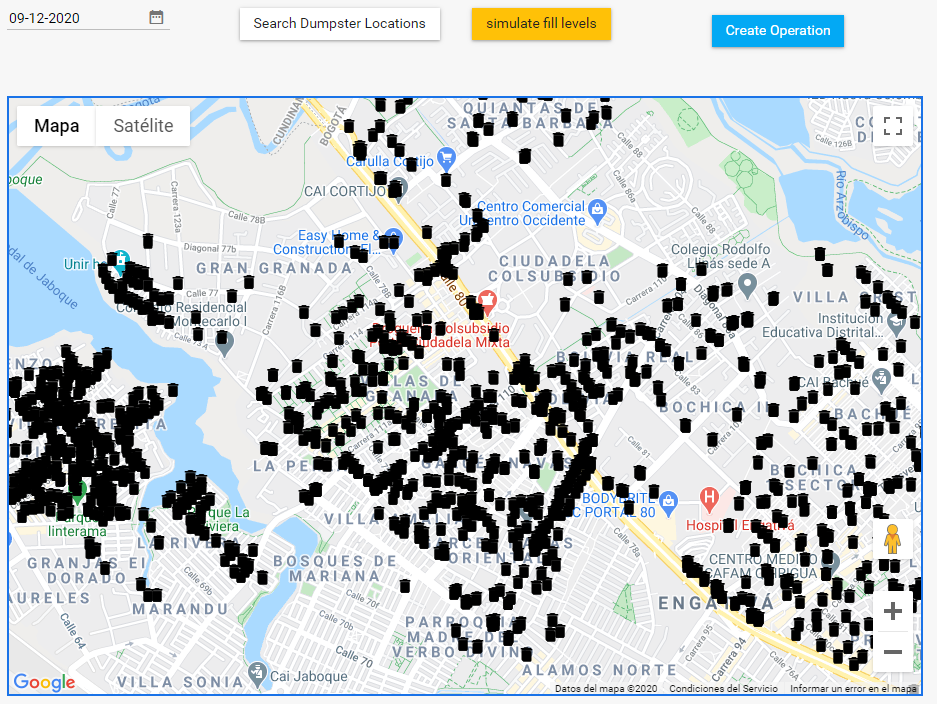
As we only provide from 3 real measure units it is impossible to generated significative plans and routes for collecting RSU other that just visiting 3 locations of our 3 measure units. So for these reason it was required to simulate a bigger amount of units reporting fulfillment levels and evaluate route calculation in conditions closer to the reality.

For the locations of the container, we defined a set of rules:

1. a container should be located on a street were a truck can reach it
2. containers are located always in pairs 1 for general purpose waste and 1 for recyclable material,
3. Each pair of container should be separated by at least one street length distance, these is to say 100 meters.

Third rule was thought to find a distribution of containers in a long area avoiding that many container are crowded in the same area. To implement first rule it was necessary to have geographical information of street configuration in the study region, therefore we download geographical data form “Unidad Administrativa Especial de Catastro Distrital Bogota” IDECA. The first set of geographical data contains the street axis lines that shapes the ways mesh of Bogota, and the second that contains the membership of blocks and neighborhood to Bogota districts.

Downloaded information was treated and organized, simulation was divided in 2 phases: the first one was to put representations of empty containers in the street were a truck could reach them and find an even distribution along the study area. To do so we used the street locations and we find a point contained in the street coordinate, the generated containers where represented in a map as show in figure 14.

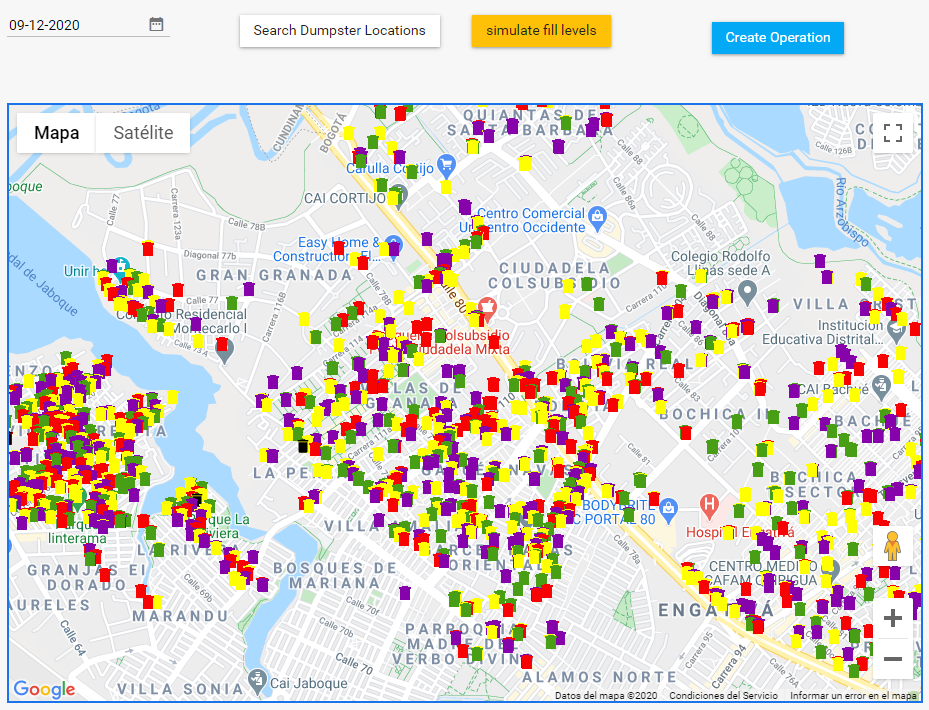


**Figure 18**: phase one container location simulation.

The second phase was to update the level of containers using random numbers between 0 and 100 representing the percentage of fulfilment, these numbers have the same mean and standard deviation found in a dataset of waste collecting routes from Austin Texas downloaded from Kaggle[[5]](#footnote-6), the colour of each container was updated as well according to convention show in table 7, The result is show in figure 15.

|  |  |  |
| --- | --- | --- |
| **level** | **Priority** | **Color** |
| Less than 10% | empty | black |
| Between 11% and 30% | low | green |
| Between 31% and 55% | medium | yellow |
| Between 56% and 80% | high | purple |
| Greater than 81% | full | red |

**Table 7**: colour code for container priority representation

**Figure 19**: phase two container filling level simulation.

We consider some facts so that the operation would be more efficient according to the variables defined in formula 1, we consider that if the truck must go across a street to collect a the content of a high level container, it must collect other containers in the same street, because when it collects only the container that reports the high level on the next days it should return to the same street to collect other containers increasing the travel distance.

It is required to reduce streets distance between containers as sometimes containers may seem spatially closed, but streets to go from one container to the other are not, so in many cases we must collect first containers that are spatially further but whose travel length is less.

Having these consideration in to account it result easier to prioritize the streets where containers are located, rather than containers, then we establish the street priority based on the rules defined in table

|  |  |  |
| --- | --- | --- |
| Street Priority | dumpster level reported | Description |
| 1 | full > 0 | there is at least a full container, or the container has not being collected in more than 3 days |
| 2 | high > 0 | there is at least a high level container, or the container has not being collected in more than 2 days |
| 3 | Medium > 0 | there are only containers with medium and low level |
| 4 | low > 0 | there are only containers with low level |

**Table 8**: rules to define street priority

With clear information of containers that has to be collected and streets prioritized we are almost ready to start planning the collection operation, but route calculation can be computational expensive and time consuming, so thinking over that for routing vehicles we will use an heuristic search base on tree spanning, and as we know the approximate volume of RSU that should be collected and the capacity of trucks in the fleet, we can estimate the number of truck required for the operation, even more, knowing the location of containers we can group the ones that are geographically close in such a way that those containers can be served by the same truck.

As group strategy we use a K-mean clustering this clustering method is fully explained in (Nazeer & Sebastian, 2009). But the fundamental behind this method is:

1. To define a feature reference and a number of groups.
2. Initial groups centroids are defined randomly, then elements are assigned equitably to the groups, one element is assigned to a group whose centroid is closer with respect to the value of the element itself.
3. the group centroids are recalculated extracting the means of the elements assigned to that group, and the elements are relocated in new groups according to the new group centres
4. The operation is iterative and it repeats until there is no reduction in global distances

Global distances is calculated as the sum of the distances of each elements to the centroid of the group where it was assigned, by reducing distances in each group global distance is also reduced.

For the clustering we implement a function that sums the volume of RSU located in the streets that have priority 1 or 2, using this volume we estimate K as the number of trucks required for the operation, that number matches the number of groups. So after the grouping, each truck could be assigned exactly to one group, thus we include a constraint to avoid that a groups of streets accumulates more volume than the one the truck could carries. Then we extract the centroid of those streets in terms of Latitude, Longitude and we use these values as reference features. The result of streets clustering is show in figure 20.

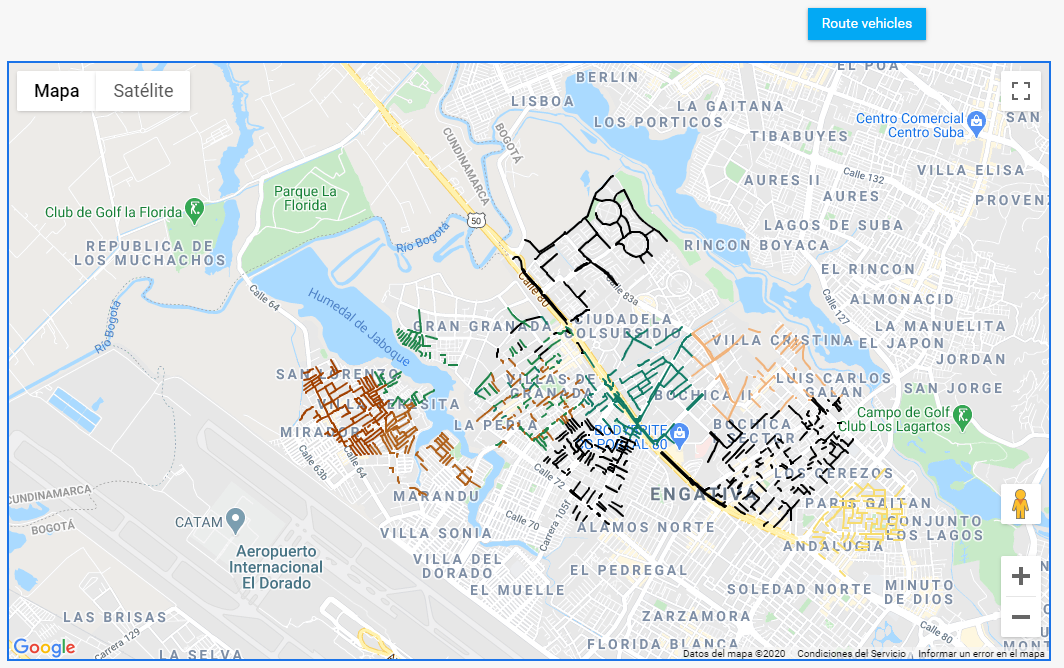


Figure 20 prioritized streets using K=9.

* 1. Route Calculation and Delivery

Once streets were prioritized and grouped the next step was to find a collection sequence that reduces travel distance in each group, taking advantage of available open services we decide to use one applied to route vehicles ORS[[6]](#footnote-7), the service uses a routing strategy called tree spanning which consist in finding the way of connecting nodes going from the closer one to the furthest or from inside to outside, the routing cost for 2 vertices is given as the sum of the cost of the edges in the unique tree path between them. (Wu et al., 2000) the strategy was originally proposed for communication routing, nonetheless considering the streets distance or time as a factor of the route cost, it can be applied to vehicle routing. The service performs some modifications to the initial solution trying to reduce distances applying operators, some of them are, relocation, interchange, inversion and alteration the solution strategy has being described in detail by (Zhou, et al. 2018) the service developers have publish another video explaining service solution strategy, functioning and use, this video can be accessed in (Coupey, J. 2018).

As ORS service is offered through and API getting a key to use it was compulsory, the service we needed was optimization, these requires to specify the places to visit and the truck features in a structure manner in JSON format which is documented in an open repository in Github[[7]](#footnote-8)

k-means grouping is a significant progress in 2 ways: first doing the assignation so we know in advance which truck will serve a group of containers and second the centroid of each group is handy as start location for each truck, so when we executed a routing task with the service we attach exactly one truck, then we transform each container location that belongs to the group into a job using the format the API expects. As a response the service return another JSON containing a lists of trucks with an array of container locations with high priority that the truck must visit, with an specific order that reduces the travel distance, a request example is shown in figure 21, then solution is stored in the database avoiding reprocessing every time an application demands routes, so the routes information is available as soon as external applications require it saving time, but also considering the service limitation of use per day.

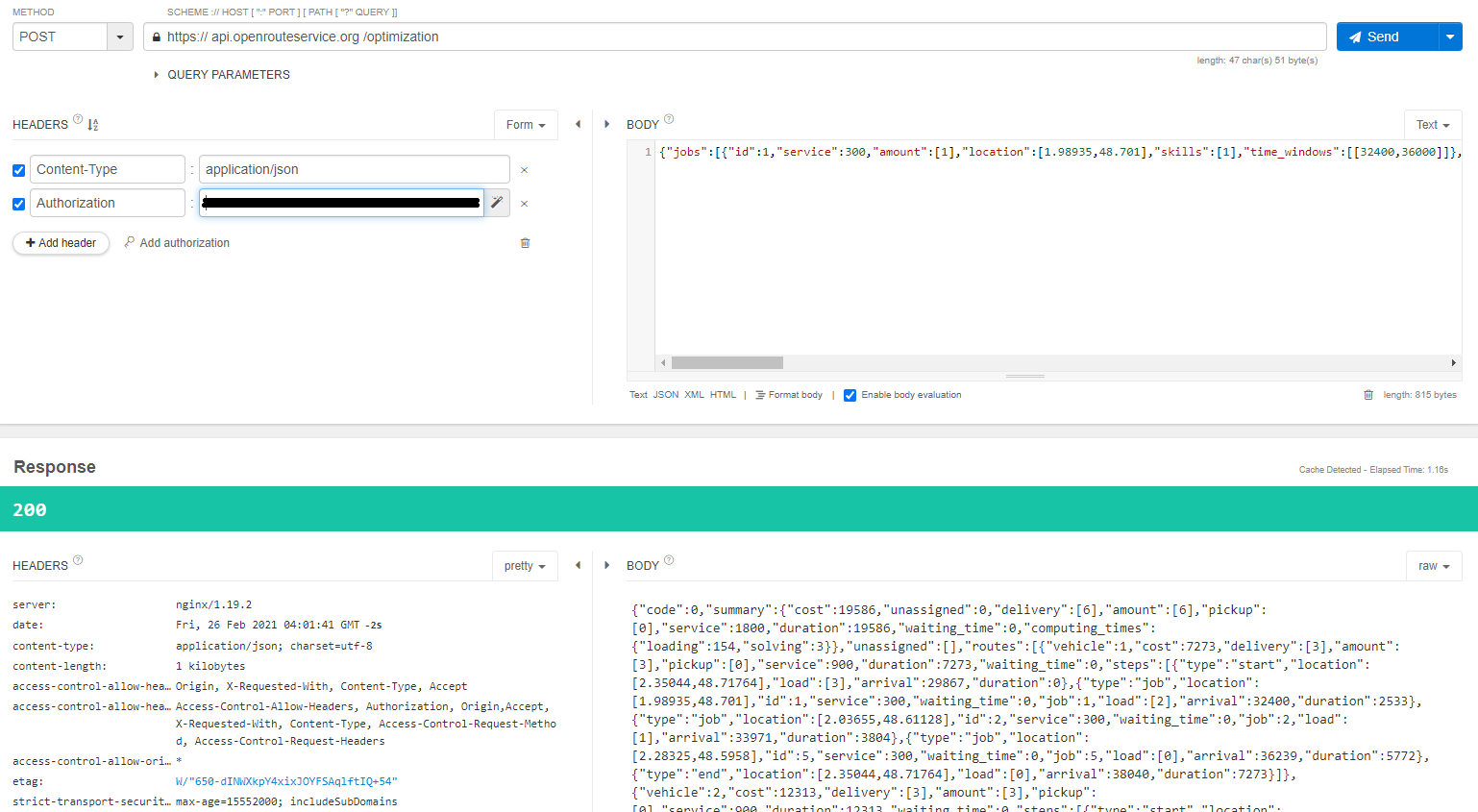


Figure 21 ORS API route request, optimization service

Up to this point we have the routes in geographical coordinates, now we need and effective way to display information to users in an easy to read and understand way. Considering that it is likely that persons who are going to use this information are truck drivers while searching dumpsters, it would be comfortable to visualize information in a portable device with internet connection, so a mobile application could fit these requirements.

A mobile application was coded in Flutter a recent Google framework for mobile application development it has an easy integration to Google services, such as Google Maps, we use a REST architecture for the application simply exposing services to handle request made by client applications, this is a simple way to transfer information in JSON format: some of the services exposed were authentication, generated routes in a specific date.

The list of routes is query via get verb using the date as parameter, the response is a list of routes is display in a component called list view see figure 22.

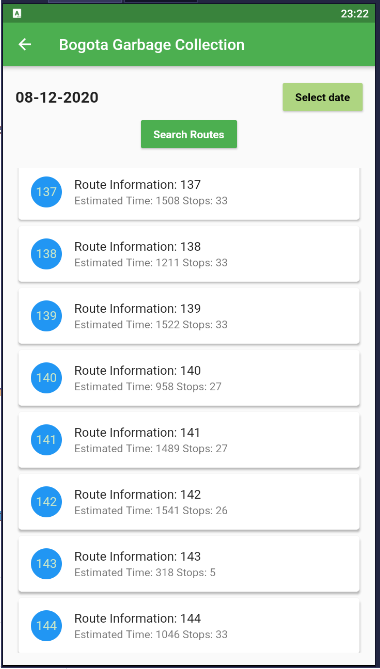


Figure 22 list of routes planned for 08 December 2020

By tapping in one element of the list, the plication will display the locations of containers that must be served for the current route see figure 23



Figure 23 initial view of container when a route is selected

When start route button is pressed the application will consume Google Directions API to find the fastest way to arrive from the driver current location to the first container in the route, when the first container is reached and served the next container button is enabled to find the path from last visited location to the next container in the route, this operation is repeated for all container in the route, a progress bar indicates collection progress it is updated when a new container is reached.

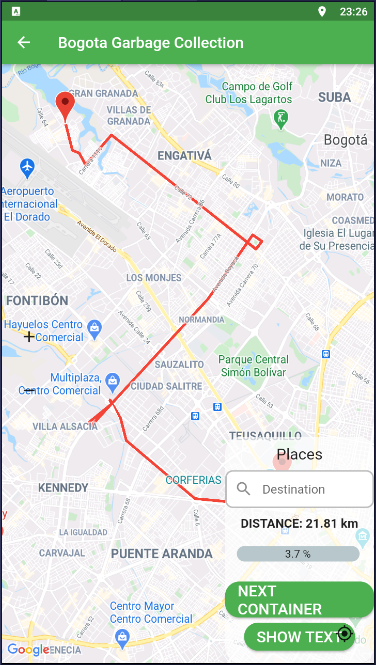


Figure 24 route from driver location to the first container

1. Discussion

To evaluate results we would prefer to analyse them separated by layers. In regard to the data collection process denominated the physical layer. The measurement units extract precise fulfilment levels and communicate them in efficiently to the server, MQTT broker acts as bridge so that data can be stored in a data base where applications can easily access it. In that sense we consider that even though there are some elements that can be enhanced, system physical part is doing pretty good.

The logical part of the system is an approach of how is possible to profit and handle collected data, but possibly there are more precise and efficient ways to treat this data; clustering containers needs to consider terrain characteristics, implementing other routing strategies or using private routing services should be validated and compared to the obtained results to select the most suited for collecting operation.

Although simulated data does not match the actual waste generation in Bogotá and it was used for academic purposes only, therefore this study does not compare the obtained routes to the routes are currently being applied to collect waste in Bogotá city, simulated data is valuable as it could be used as reference to evaluate routing algorithms performance.

There are still elements that can significantly improve the system here we explain some of the: using LoRa as transmission technology needs some considerations first is that since LoRa uses a star gateway centred topology, it is required to find an efficient gateway distribution to maximize coverage region reducing number of gateways, nonetheless a 50 % of coverage overlapping is highly recommended so that each measure node is to be served by at least by 2 gateways, this generates a fail tolerance in case any gateway is temporally out of service, another gateway in the network might serve as back up for the nodes allowing them to continue delivering measured an keeping them communicated with the server.

Gateways should be placed in altitude above building average height, trying to generate a direct line of sight from it to measurement units so messages are deliver and recived in the gateway, notwithstanding there possibly changing transition parameters as Spreading Factor and Bandwidth might upgrade signal power so when signal gets obstructed by obstacles as wall the gateway can still receive the signal anyway to conclude this point more research is required.

When grouping containers factors, other than geographical proximity need to be reviewed, for instance hydric resources as wetlands and rivers. In this work, we realized that clustering aggregates containers that were relatively close geographically, but some of them were separated by a river, so aggregating those in the same group imply to circle the river until the next bridge, which increases travel distance and affects negatively the route efficiency when evaluated using formula 1. To solve the problem a clustering strategy with spatial boundaries should be implemented in such a way that elements would be assigned to groups within the hydric boundaries and not overstep those limits.

Before implementing the proposed collection strategy a relocation of containers is required, it is necessary to conciliate a place where containers will be reached easily for citizens and recycling enterprises staff, but situated in easy access ways minimizing detours, or narrow ways were truck transit is difficult and could retard the operation.

Measurement units need to be improved on two facts: first they need to manage energy efficiently, for that it is required to adapt low consumption switches that turn on the devices exclusively when they will extract measures possibly 2 or 3 times a day, the other time they need to remain in sleep mode. Photovoltaic cells might be a good energy source alternative. Second, an external case need to be designed to protect circuits from weather conditions and manipulation damage.

Measurement units can perform very good if dispense with GPS sensor, it is not decisive as container locations can be indexed to the data schema when units are installed and identified by a unique id code, this reduces nodes cost in 33% approximately but also reduces energy consumption.

To increase operation profits routes should be separated; there must be at least two types of routes: one to collect general disposable waste and other for recyclable material. Last one is to be classified and manage properly avoiding it to get contaminated and come to an end in landfills where it becomes a source of contamination and damage for the environment.

The mobile application does not show exact alignment between the streets delivered by the maps provider and lines used to show routes, this happens because data used for simulation and routing uses a different geographical system reference than the maps provider. While data used for simulation and routing was coded using Magna Sirgas, official Bogota geographical reference system, maps provider uses the World geodesic system 84 WGS84, this causes that visually the paths shows a slight deviation, it is necessary to find a map provider that uses a reference system similar to Magna Sirgas.

6. Conclusions

A sensing system was implemented with LoRa as transmission technology, data flow was managed along the network components until data stored it in a database, collected data was processed to generate collection routes that priorized the collection of containers showing high levels, increasing collected volume and reducing travel distances, planned routes were displayed in a mobile application as support for collecting operations.

This research shows how updated data is an important asset that cities can take advantage of to better plan typical operations as RSU collection, the developed sensing system demonstrates technical viability to implement this collection schemas in real environment, showing that logistic process of RSU collection can by enhanced with communication technologies to reduce container saturation and overflow.

This work should be extended and generalized to fit general cities characteristics, so that cities have an open framework to manage their RSU in an efficient and sustainable way.

ACKNOWLEDGMENTS

Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here. Insert paragraph text here.

REFERENCES

[1] Nazeer, K. A & Sebastian, M.. (2009). Improving the Accuracy and Efficiency of the k-means Clustering Algorithm. Lecture Notes in Engineering and Computer Science. 2176.

[2]Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00

1. Full duplex is a communication mode that allows to emit and receive messages simultaneously. [↑](#footnote-ref-2)
2. <https://www.thethingsnetwork.org/> [↑](#footnote-ref-3)
3. An open source lightweight MQTT broker supported by Eclipse foundation [↑](#footnote-ref-4)
4. https://github.com/TheThingsNetwork/java-app-sdk [↑](#footnote-ref-5)
5. https://www.kaggle.com/jboysen/austin-waste [↑](#footnote-ref-6)
6. Open Route Service [↑](#footnote-ref-7)
7. <https://github.com/VROOM-Project/vroom/blob/master/docs/API.md> [↑](#footnote-ref-8)