Adding voice messages to a low-cost long-range data messaging system

Kiyoshy Nakamura
Universitat Politécnica de Valéncia
Valencia, Spain
minapin@posgrado.upv.es

Pietro Manzoni
Universitat Politécnica de Valéncia
Valencia, Spain
pmanzoni@disca.upv.es

Marco Zennaro
ICTP
Trieste, Italy
mzennaro@ictp.it

Juan-Carlos Cano
Universitat Politécnica de Valéncia
Valencia, Spain
jucano@disca.upv.es

Carlos T. Calafate
Universitat Politécnica de Valéncia
Valencia, Spain
calafate@disca.upv.es

ABSTRACT

Remote locations in rural areas can benefit from any system that would provide some form of connectivity. In a previous work we described a low cost architecture that, thanks to the use of the LoRa technology, allowed long links using low energy and a cheap infrastructure. In this work we extend those results by adding the possibility to include generic external sources of data using an MQTT based interface. More specifically, we describe a voice messaging system that allows users that cannot read or write to send voice based messages. We describe how the system was integrated into the existing platform and present some performance results. We consider that these results are promising and provide a tool that can offer a useful service at a low cost.

CCS CONCEPTS

• **Networks** → *Peer-to-peer protocols*; **Network architectures**; • **Human-centered computing** → **Accessibility systems and tools**; • **Information systems** → Collaborative and social computing systems and tools.

KEYWORDS

LoRa, MQTT, IoT, voice

ACM Reference Format:
Kiyoshy Nakamura, Pietro Manzoni, Marco Zennaro, Juan-Carlos Cano, and Carlos T. Calafate. 2020. Adding voice messages to a low-cost long-range data messaging system. In *6th EAI International Conference on Smart Objects and Technologies for Social Good (GoodTechs '20), September 14–16, 2020, Antwerp, Belgium.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3411170.3411238

GoodTechs '20, September 14–16, 2020, Antwerp, Belgium

ACM ISBN 978-1-4503-7559-7/20/09…$15.00
https://doi.org/10.1145/3411170.3411238

1 INTRODUCTION

While in developed countries most people are online, with close to 87 per cent of individuals using the Internet, in the least developed countries, on the other hand, only 19 per cent of individuals were online in 2019. Europe is the region with the highest Internet usage rates, whereas Africa the region with the lowest Internet usage rates. The global penetration rate increased from nearly 17 per cent in 2005 to over 53 per cent in 2019 [1].

A very active Internet Research Task Force (IRTF) group in this context, the "Global Access to the Internet for All Research Group (GAIA)" [2] focuses its activities to provide increased visibility and interest among the wider community on the challenges and opportunities to enable global Internet access, in terms of technology as well as in the social and economic drivers for its adoption.

In "off-line areas", even simple messaging services would be of great help for example to farmers wishing to know the price of a certain good they are interested to sell or buy before deciding whether a possibly long, expensive and exhausting trip is undertaken. For example, Martinez et al. offer [3] an analysis of the communication needs in rural primary health care in developing countries. They found that one very simple application originally implemented over HF voice-only radio communication proved quite successful: scheduling doctor's appointments for patients. It was found that patients in isolated areas had to spend significant time and resources to reach the nearest hospital, and often they could not be treated immediately, but given an appointment at a time that implied a second trip from home.

The general scope of this work is to offer solutions to provide connectivity in poorly or not-connected at all scenarios. More specifically, we focus on the the critical aspect of providing a solution for the inclusion of illiterate people, that is people that cannot read and write, in such types of systems. This goal is clearly related with the general goals of frugal innovation. Zeschky et al. [4] produced probably the most cited definition that states that frugal innovations as 'good-enough' and affordable products or services that suffice the needs of bottom-of-the pyramid. The major driving factors for these innovations were identified in the followings resource constraints: (a) less developed infrastructure, (b) institutional voids, (c) illiteracy, and (d) low purchasing power [5].

In this paper we extend the results of a previous work [6], where we presented a messaging system based on LoRa[7], by adding the possibility to include generic external sources of data using an MQTT based interface. There are in the literature other works that combine these technologies to provide connectivity related services, especially related to IoT, like in [8–14]. In [15] the authors proposes the design of LoRa-MQTT gateway device for supporting the sensor-to-cloud data transmission in smart aquaculture IoT application.

Our proposal combines a visual interface with an edge solution for data compression to include voice messages in the system. The objective is to allow users that cannot read or write to send voice messages to access remote information and services. We describe how the system was integrated in the existing platform and present some results on its performance. We consider that these results are promising and describe a tool that can provide a useful service at a low cost.

The paper is organized as follows. Section 2 describes the existing LoRa based messaging system while Section 3 details the integration of a MQTT proxy in the system. Section 4 presents the architecture of the voice recording source and Section 5 some evaluations results. Finally, Section 6 presents the conclusions.

2 THE LORA MESSAGING SYSTEM

The overall architecture of the messaging system is detailed in [6]. At the core there are dedicated devices, called *hubs*, that create the connectivity spot inside an area. The hubs have both a WiFi (IEEE 802.1b/g/), and a LoRa transceiver. The hubs work as standard WiFi access point to provide connectivity to close by devices. The interface with the messaging application is a web based system. The user can decide whether to send a text message to either a specific destination or to all reachable users, or to check for incoming messages stored in the hub. The hubs offer a REST interface to the connected devices to either send a message, or return previously received and locally stored ones.

Every user needs to "register" before exchanging any message. Registration is required to allow the system to localize end-points. When a user sends a message, the local hub "learns" that that user is connected through it, and creates an entry in a table. The first step is to discover where the destination user is located. To this end, the hub sends a broadcast message to all the surrounding devices and waits for the searched one to respond. A special *broadcast* user was included for messages that are to be delivered to all the registered users.

At this point, using a reliable unicast protocol, messages are transferred and stored in the destination hub. Once the user to whom the message is addressed to checks for available messages, he or she will receive the one stored in the local hub. The unicast protocol is based on a stop-and-wait ARQ approach with a dynamic and adaptive value for the re-transmission delay. The protocol ensures that information is not lost due to dropped packets and that packets are received in the correct order.

Finally, to better integrate our system with standard Internet applications, we designed a *gateway-hub* to link it with Telegram[1], a widely used messaging application, using "Bots". In short, if the gateway-hub receives via LoRa messages directed to a Telegram user, registered through the Bot, it forwards them to the user's phone via the Internet.

[1]https://telegram.org/

3 INTEGRATION OF EXTERNAL SOURCES

We integrated in the platform described before the data collection from external sources, that is inputting data from anything from a simple temperature sensor, to, as in this paper, a voice recorder. We defined a dedicated service called `MQTTproxy` that is attached to the system as a specific client. The general idea is that the external source has to pack the content it's providing as a structured piece of information, and send it to the `MQTTproxy` as a message. The `MQTTproxy`, will then (1) unpack the message, (2) build a proper MQTT message, and (3) publish it to the broker being used. The device that executes the `MQTTproxy` service has to be connected both to a hub using WiFi and must clearly have a connection to the used broker, either through the Internet or through a direct TCP/IP link.

The set-up required is the one shown in Figure 1. The external source is attached to the "Hub 1" through a WiFi link but also Bluetooth can be a option.

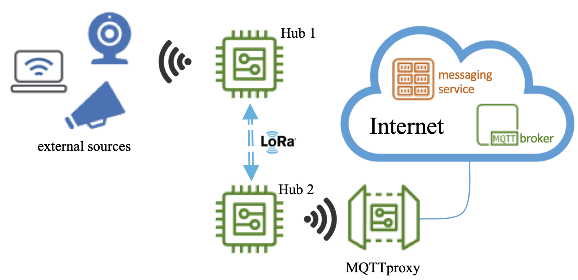


Figure 1: The connections structure among the external source and the MQTTproxy.

External sources are integrated in the platform and "talk" with hubs using a REST interface. The sequence is basically the same used by regular clients: there is first a registration phase followed by a "Push" phase. Figure 2 graphically describe this operations.

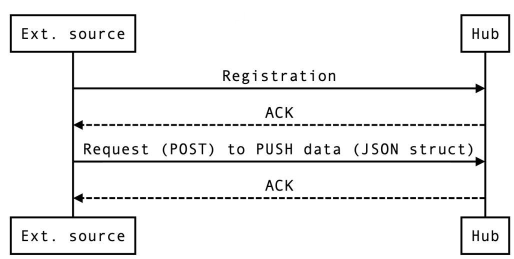


Figure 2: The data flow when an external source data is pushed to a hub.

The transferred data is structured as a JSON object with the form indicated in Listing 1.

The data contained in the JSON object can have a variable size limited only by the slow data rate that we can obtain from the LoRa

```
{
    'DEV_ID': 'voice_recorder',
    'QOS': 0,
    'TOPIC': 'rasp1095/voicemessage',
    'VALUE': {'sender': 'pietro',
              'receiver': 'miguel'
              'message': <mp3 file>}
}
```

Listing 1: The structure of the messages interchanged.

channel. Topics average length can range between 10 to 50 bytes, while values can be anything from a few bytes, to hundreds of kilobytes, like in our case with the voice recording source. In Section 5 we will present the results with messages of up to 100kbytes. We are aware of the limitations that LoRa imposes related to the use of the channel, with what is called "duty cycle"; messages of 100kB can violate this rule if the duty cycle is applied to slots of 1 hour, as it is suggested. In our case we assume that the areas where these channels will be used will not have a too high density of devices, thus limiting the possible impact of these long messages. There is anyway the need, as future work, to provide some form of sending scheduling to avoid possible interference with other LoRa sources.

The localization of the `MQTTproxy` is based on an "anycasting" approach. This means that there can be various `MQTTproxy`s available in the area covered by any hub. As for regular clients, the hub that received the JSON message will start the search for an `MQTTproxy` as if they were regular end users; if multiple replays are received the first one is selected. Other strategies could be adopted, based for example on the detected load of a certain `MQTTproxy` device. The hub will forward the JSON message sent to it to the selected `MQTTproxy` hub using the standard procedure used by the messaging system. The `MQTTproxy`, using the REST interface, will periodically probe the hub it is connected with to obtain the data. Figure 3 graphically describes this operation.

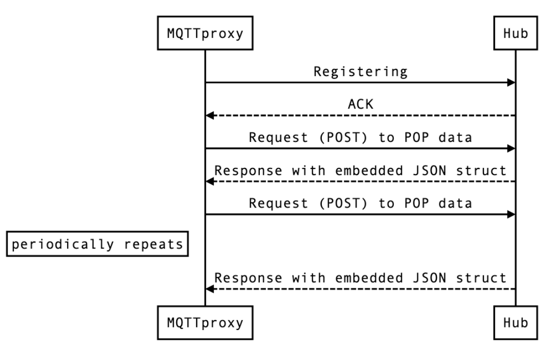


Figure 3: The MQTTproxy getting data from the hub.

Once the message is obtained, the `MQTTproxy` will extract the JSON and create a proper "publish" message to the connected broker.

Published messages will be received by a specific "messaging service" available in the cloud (see Figure 1) that will forward it to the destination through email or other means. The obtained reply will be handled the same way, by having the "messaging service" publish it to the broker and the `MQTTproxy` sending it back to the sender.

4 THE VOICE RECORDING EXTERNAL SOURCE

This section details the design and user interface of a voice recording external source. As explained previously the objective is to allow people that cannot read and write to send vocal messages. The aim was to obtain an interface that was easy to use and effective. The prototype of the source was implemented on a Raspberry Pi 3 Model B+ device with an add-on screen (Waveshare[2]). This 1.44-inch LCD screen has a joystick and three buttons that we used for the interaction of the user with the system.

As a first step, the user must identify himself (Figure 4). On the screen he will be able to see a series of photographs of users registered on that device, from which he will have to choose his own with the help of the joystick.

After the user identified himself, on the screen he will be able to see his photograph and some small icons that indicate the actions that each of the three buttons on one side of the screen perform (Figure 5). Basically, the actions are: (key 1) start recording, (key 2) stop recording, and (key 3) choose destination.

Pressing the 'key 1', the user will be able to record a voice clip of up to 50 seconds. This values was chosen based on various tests we did so to obtain after compression a content to be sent with a size at most equal to 100kB. We consider that 50 seconds is a reasonable compromise between the information that can be sent and the total time necessary to be sent. Section 5 shows some results in this sense.

If the message to be recorded does not require the whole 50 seconds, registration can be stopped by pressing 'key 2' (see Figure 6).

The voice clip generates a file with a .wav extension, when stored in the microSD memory of the Raspberry; a 50 seconds message typically has a size of 1 MB. Our messaging system allows us to send messages up to 100 Kb for this reason the .wav file must be compressed in order to be sent. To this end we use the FFmpeg[3], a framework that allows the manipulation of multimedia files, to compress the .wav files into .mp3 files. The resulting .mp3 file does not lose audio quality and its size decreases up to a factor of ten. Figure 7 allows to see the evolution of the compression time as a function of the recording time of the audio clip. As we can see, the compression time clearly grows as the recording time increases. Tests were performed with audio clips of 10 s, 20 s, 30 s, 40 s and 50 s and in the Table 1 we can see the size of the generated .wav files and the size of the .mp3 files when compressed.

Once the user has finished the recording process of the voice clip, they can listen to it. If the user decides to send their voice clip, they must press 'key 3' and it will indicate that they must choose a destination as shown in the Figure 8.

[2]https://www.waveshare.com/
[3]https://ffmpeg.org/

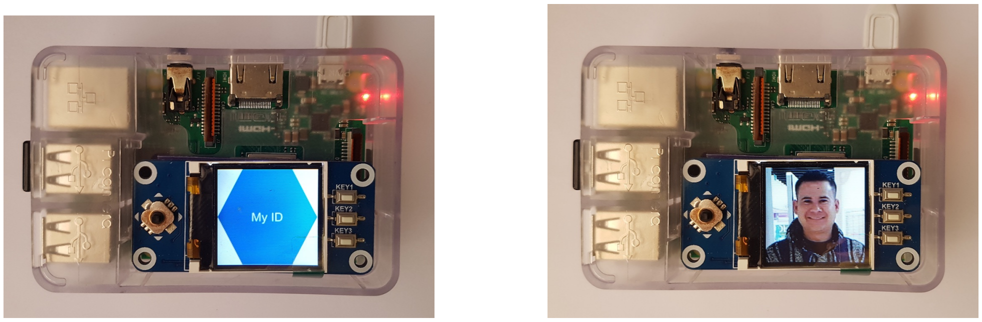


Figure 4: Screen sequence when identifying the sending user.

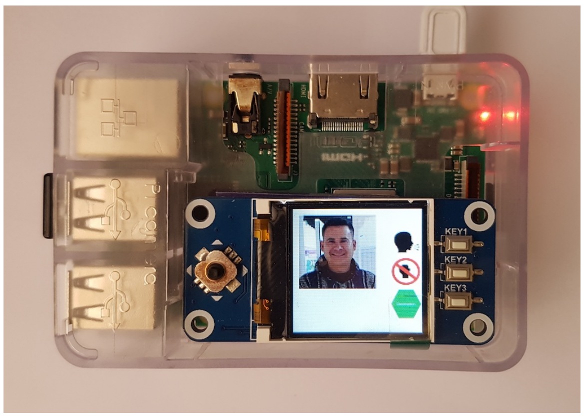


Figure 5: Screen showing the sender image and the action indicator icons.

Recording time (s)	.wav file size	.mp3 file size
10	220 Kb	24 Kb
20	436 Kb	40 Kb
30	652 Kb	60 Kb
40	868 Kb	80 Kb
50	1 Mb	100 Kb

Table 1: Size of .wav and .mp3 files

At this point the system already have all the required information, namely the source, the destination and the voice message, so that the JSON file can be prepared to be sent to the `MQTTproxy`.

5 EXPERIMENTAL RESULTS

This Section presents the performance results obtained with the proposed system by varying the distance between the hubs and the size of the sent messages. We considered the following distances between the two hubs: 1m, 750m, and 6000m. The 1 meter test were performed to have reference values to be compared with the results with longer distances. The 750m tests where performed in Valencia, in the "Ciudad de las Artes y las Ciencias" area, while the 6km test where performed between two viewpoints in Chiapas, México. These latter locations are in areas high enough not to have obstacles in between; Figure 9 shows the scenario where we the tests were done with a clear line of sight between the two points 6km away.

We measured the performance of the system using a metric called "successful transfer time (STT)". It measures the transfer time of a message from the point of view of the sender, and it is computed from the moment at which the first fragment of the message is sent, to the moment when the last ACK of the last fragment of the message is received. All the tests were performed using a Spreading factor of 7 (SF7) to minimize the time on air.

Bursts of 10 messages were sent to determine the stability of the system. The system performance was stable and almost identical to that of the tests at shorter distances. Re-transmissions were rare, even in the long-range experiments, having a negligible impact on the SST. We have to consider that delays are in the order of hundreds of seconds, and therefore a few more seconds do not affect the usability of the system; no effect was detected on the vocal messages delivery. Figure 10 shows the behavior of the STT versus distance between two nodes

An almost constant behavior can be observed in the results, although the STT clearly grows as the message size increases. The system is quite stable at increasing distance and very few re-transmissions were required during the experiments. For example, with messages of 100kBytes the maximum delay obtained was 457,56 sec, and the minimum 451,49 sec, about 7 minutes and a half.

Clearly, the worst aspect is the low throughput that we can obtain due to the use of LoRa. We consider, anyway, that this is compensated by the long range obtained and by the low energetic cost that these devices require, thus making this a frugal solution to a clear problem. As indicated in the previous Section we are aware of the limitations that LoRa imposes related to the use of the channel, with what is called "duty cycle"; messages of 100kB can violate this rule if the duty cycle is applied to slots of 1 hour, as it is suggested. In our case we suppose that the areas where these channels will be used will not have a too high density of devices, thus limiting the possible impact of these long messages. There is anyway the need, as future work, to provide some form of sending scheduling to avoid possible interference with other LoRa sources.

6 CONCLUSIONS

In this paper we presented an extension to the results of a previous work where we described a low cost architecture that, thanks to

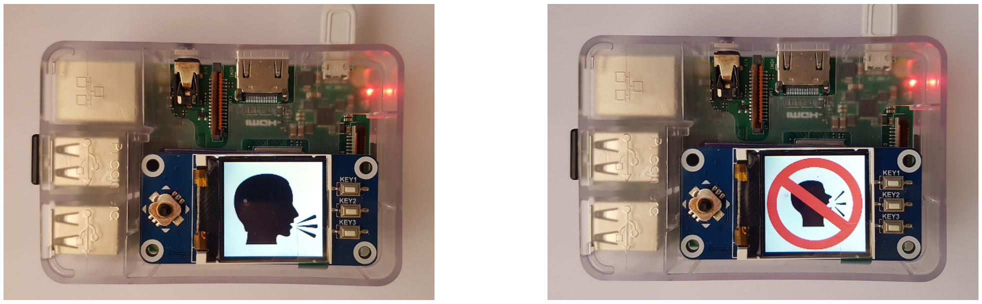


Figure 6: Image indicating that the recording process has started (left) and stopped (right).

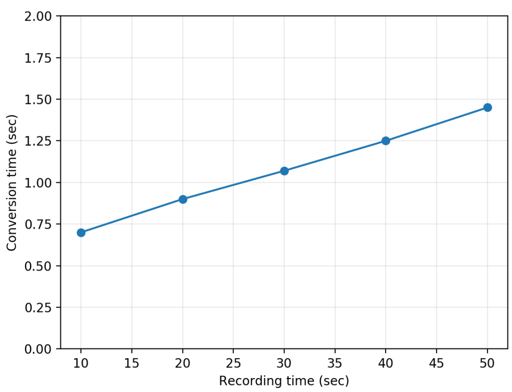


Figure 7: Compression time (median values).

the use of the LoRa technology, allowed long links with very low energy and infrastructure required. The goal was to provide remote locations in rural areas of a communication system that would provide some form of connectivity.

In this work we provided the possibility to include generic external sources of data using an MQTT based interface. More specifically, we described a voice messaging system that allows users that cannot read or write to send voice based messages. We described how the system was integrated in the existing platform and presented some results on its performance. We consider that these results are promising and describe a tool that can provide a useful service at a low cost.

As future work we aim to improve the user interface by having it tested by various users and to improve the data compression so to either allow for longer messages or to reduce the time required for their transfer.

7 ACKNOWLEDGMENT

This work was partially supported by the "Ministerio de Ciencia, Innovación y Universidades, Programa Estatal de Investigación, Desarrollo e Innovación Orientada a los Retos de la Sociedad, Proyectos I+D+I 2018", Spain, under Grant RTI2018-096384-B-I00.

REFERENCES

[1] ITU/UNESCO Broadband Commission for Sustainable Development, "The state of broadband 2019," On-line, ITU/UNESCO, Report, September 2019. [Online]. Available: "https://www.itu.int/dms_pub/itu-s/opb/pol/S-POL-BROADBAND.20-2019-PDF-E.pdf"

[2] J. Saldana, A. Arcia-Moret, B. Braem, E. Pietrosemoli, A. Sathiaseelan, and M. Zennaro, "Alternative Network Deployments: Taxonomy, Characterization, Technologies, and Architectures," Internet Requests for Comments, RFC Editor, RFC 7962, August 2016. [Online]. Available: https://www.rfc-editor.org/rfc/pdfrfc/rfc7962.txt.pdf

[3] A. Martinez, V. Villarroel, J. Seoane, and F. del Pozo, "Analysis of information and communication needs in rural primary health care in developing countries," *IEEE transactions on Information Technology in Biomedicine*, vol. 9, no. 1, pp. 66–72, 2005.

[4] M. Zeschky, B. Widenmayer, and O. Gassmann, "Frugal innovation in emerging markets," *Research-Technology Management*, vol. 54, no. 4, pp. 38–45, 2011. [Online]. Available: https://doi.org/10.5437/08956308X5404007

[5] T. Schuster and D. Holtbrügge, "Resource dependency, innovative strategies, and firm performance in bop markets," *Journal of Product Innovation Management*, vol. 31, no. S1, pp. 43–59, 2014. [Online]. Available: https://www.onlinelibrary.wiley.com/doi/abs/10.1111/jpim.12191

[6] A. M. Cardenas, M. K. Nakamura Pinto, E. Pietrosemoli, M. Zennaro, M. Rainone, and P. Manzoni, "A low-cost and low-power messaging system based on the lora wireless technology," *Mobile Networks and Applications*, Apr 2019. [Online]. Available: https://doi.org/10.1007/s11036-019-01235-5

[7] B. S. Chaudhari BS, Zennaro M, "Lpwan technologies: Emerging application characteristics, requirements, and design considerations," *Future Internet*, vol. 12, no. 3, 2020.

[8] A. S. Bharadwaj, R. Rego, and A. Chowdhury, "Iot based solid waste management system: A conceptual approach with an architectural solution as a smart city application," in *2016 IEEE Annual India Conference (INDICON)*, 2016, pp. 1–6.

[9] S. Spinsante, G. Ciattaglia, A. Del Campo, D. Perla, D. Pigini, G. Cancellieri, and E. Gambi, "A lora enabled building automation architecture based on mqtt," in *2017 AEIT International Annual Conference*, 2017, pp. 1–5.

[10] S. Penkov, A. Taneva, V. Kalkov, and S. Ahmed, "Industrial network design using low-power wide-area network," in *2017 4th International Conference on Systems and Informatics (ICSAI)*, 2017, pp. 40–44.

[11] M. Niswar, S. Wainalang, A. A. Ilham, Z. Zainuddin, Y. Fujaya, Z. Muslimin, A. W. Paundu, S. Kashihara, and D. Fall, "Iot-based water quality monitoring system for soft-shell crab farming," in *2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, 2018, pp. 6–9.

[12] A. Huang, M. Huang, Z. Shao, X. Zhang, D. Wu, and C. Cao, "A practical marine wireless sensor network monitoring system based on lora and mqtt," in *2019 IEEE 2nd International Conference on Electronics Technology (ICET)*, 2019, pp. 330–334.

[13] C. Paolini, H. Adigal, and M. Sarkar, "Upper bound on lora smart metering uplink rate," in *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, 2020, pp. 1–4.

[14] A. Lachtar, T. Val, and A. Kachouri, "Elderly monitoring system in a smart city environment using lora and mqtt," *IET Wireless Sensor Systems*, vol. 10, no. 2, pp. 70–77, 2020.

[15] A. Bhawiyuga, K. Amron, R. Primanandha, D. P. Kartikasari, H. Arijudin, and D. A. Prabandari, "Lora-mqtt gateway device for supporting sensor-to-cloud data transmission in smart aquaculture iot application," in *2019 International Conference on Sustainable Information Engineering and Technology (SIET)*, 2019, pp. 187–190.

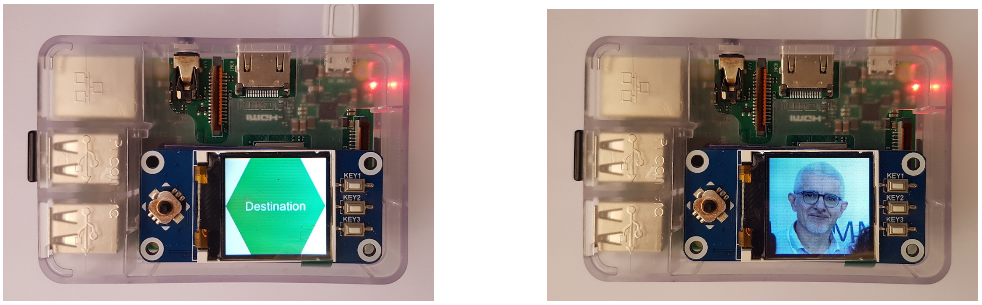
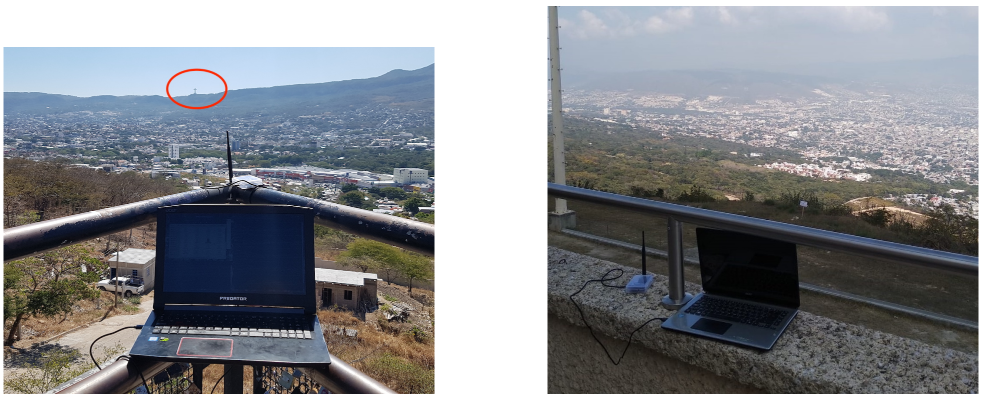


Figure 8: Screen sequence when identifying the destination user.



(a) Location 'Mirador los Amorosos'. **(b) Location 'Mirador del Glorioso Cristo de Chiapas'.**

Figure 9: Location of the 6 km experiments (Chiapas, México).

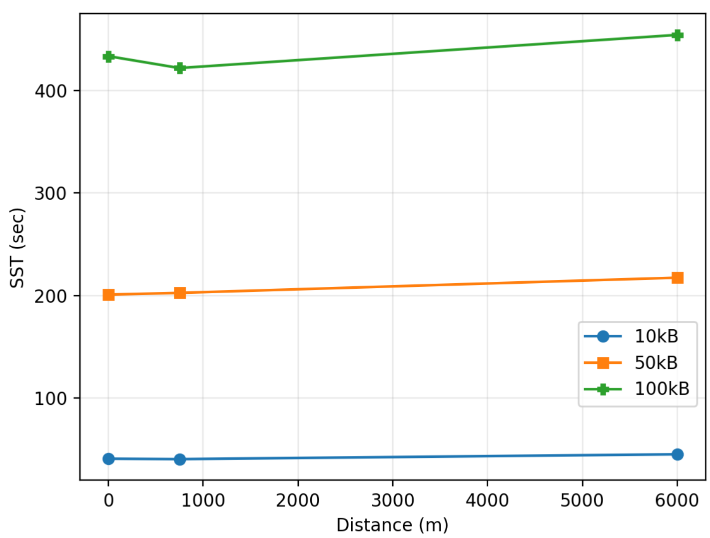


Figure 10: Behavior of the STT versus distance between two nodes.