Document downloaded from:

http://hdl.handle.net/10251/179806

This paper must be cited as:

Nakamura, K.; Manzoni, P.; Zennaro, M.; Cano, J.; Tavares De Araujo Cesariny Calafate, CM. (2020). Integrating an MQTT Proxy in a LoRa-Based Messaging System for Generic Sensor Data Collection. Springer Nature. 282-294. https://doi.org/10.1007/978-3-030-61746-2_21

The final publication is available at

https://doi.org/10.1007/978-3-030-61746-2_21

Copyright Springer Nature

Additional Information

Integrating an MQTT proxy for sensor data collection in a LoRa-based messaging system

Kiyoshy Nakamura, Pietro Manzoni
Universitat Politécnica de Valéncia
Valencia, Spain
minapin@posgrado.upv.es
pmanzoni@disca.upv.es

Marco Zennaro
ICTP
Trieste, Italy
mzennaro@ictp.it

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

***Abstract*—In this paper we combine smart computing with the frugal innovation approach to propose a "resourceful" solution, in terms of minimal utilization of resources and sustainability, to build a basic support for an IoT system in rural areas.**

Our system includes an MQTT proxy to integrate low-cost and low-power devices in a messaging system based on LoRa technology. LoRa technology provides long wireless links that can be used to connect villages and towns. Through a REST-based interface, we show how our platform can be used to distribute sensor information from rural communities. Moreover, MQTT allows these data to be provided to external "data lakes" so that they can be used for tasks such as reporting, visualization, advanced analytic, and machine learning.

Finally, we demonstrate through experimental evaluation that this solution provides stable data transfers over links of various kilometers with a minimal utilization of resources.

I. Introduction

According to ITU's (International Telecommunication Union) latest statistics [1], the world has reached an important milestone regarding connected people. At the end of 2018, 51.2% of the global population, or 3.9 billion people, had access to the Internet; in developing countries, growth has shown an increase from 7.7% in 2005 to 45.3% at the end of 2018. Despite this, and basically due to economical reasons, there are still large areas of poorly or not-connected at all zones, not only in developing countries, but also in the countryside of Europe and the USA [2].

Community networks are a successful attempt to solve this situation. They are built by citizens and organizations who join together and share resources and efforts to build network infrastructures. The employed technologies span a very wide range, from low-cost, off-the-shelf wireless (e.g., WiFi) routers to expensive optical fiber equipment. The methods of participation, organization, and funding are very diverse. For example, some networks are freely accessible, others are cooperative based, and some are run by federations of small ISPs. A few examples are: Broadband for Rural North (B4RN)[1], in Lancashire, UK, the Nepal Wireless Networking Project (NWNP)[2], or the GUIFINET in Spain [3].

The examples above are just a few representing the area of ICT for development (ICT4D). This area aims at providing sustainable solutions for ubiquitous connectivity. A very active Internet Research Task Force (IRTF) group in this context, the "Global Access to the Internet for All Research Group (GAIA)" [4] 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.

Messaging systems like WhatsApp, Telegram, or the "classical" SMS (short message service) have been and are among the most widely used applications for mobile devices worldwide since they offer an open communication channel among people or the members of a community. Clearly, the possibility to offer this service between villages, and between villages and main cities is highly demanded either for purely personal use or for commercial purposes in rural area, too.

For example, Martinez et al. offer [5] 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.

To specifically address these rural scenarios, in this work we combine ICT4D, smart computing and the growing approach of frugal innovation [6], [7]. This latter term comes from the initial focus on *Bottom of the Pyramid (BOP)* customers in emerging markets, and the process is fueled by the growing traction towards resourceful and minimalist consumption in advanced economies, also encouraging the applicability and potential exploration of 'frugality'. This concept is nowadays moving away from the basic idea of "cheap" innovations to become a more "resourceful" approach in terms of minimal utilization of resources such as water, electricity, time etc. and sustainable innovations, where sustainable means developed to be long–lasting and environmentally responsible. [8]

With these ideas in mind, in a previous work [9] we defined the architecture for a messaging system that combined very cheap and flexible devices and the LoRa technology to

[1] http://b4rn.org.uk/

[2] http://www.nepalwireless.net/

establish links that can cover wide areas with an easy-to-use interface. A simple messaging application like ours could serve, for example in the case study above, to arrange for a specific date in which the patient is guaranteed treatment, thus saving time and resources.

We centered our solution on the use of LoRa, a long range, low power wireless platform that has become a main technology for Internet of Things (IoT) networks worldwide. In mountainous regions one can leverage the terrain topography to accomplish line-of-sight transmissions at very long distances.

In this work we focused on integrating in this platform the support for ordinary sensing applications, based on any type of sensors like environmental sensors for weather forecasting, chemical sensors, level sensors, images sensors, and so on. Locating these devices close to our system's "hubs", enables them to send data across long distances and to be integrated in a Publish–Subscribe (Pub/Sub) system based on MQTT. MQTT [10] is a machine-to-machine (M2M) connectivity protocol currently widely used in the Internet of Things world. It was designed as an extremely lightweight publish/subscribe messaging transport and it is therefore extremely useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium.

Summing up, in this paper, we propose a system that can collect data from remote sensors, transmit this data over long distances using LoRa, and automatically integrate it into a pub/sub system based on MQTT.

The paper is organized as follows. Section II presents the overall architecture and functioning of the messaging system. Section III describes the integration of the MQTT proxy in the messaging system. Section IV shows the performance evaluation of a prototype of our proposal. Finally, Section V presents the conclusions.

II. The LoRa messaging system

Figure 1 presents the overall architecture proposed in [9]. At the core there are dedicated devices, called *hubs*, that create the connectivity spot inside an area. The hubs must have both a WiFi (IEEE 802.1b/g/), and a LoRa transceiver.

The hubs work as standard WiFi access point to provide connectivity to closeby 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 retransmission delay. The protocol ensures that information is not lost due to dropped packets and that packets are received in the correct order.

The packet header structure is shown in Figure 2. We must point out that the maximum application payload depends on the selected data rate. For example, assuming the European 863-870MHz band, the maximum packet size used was set according to the spreading factor used, i.e. 25 bytes for SF12, and 200 bytes for SF7, with a fixed 24 bytes header.

Finally, to better integrate our system with standard Internet applications, we designed a *gateway-hub* to link it with Telegram[3], a widely used messaging application, using "Bots". In short, the gateway-hub receives via LoRa messages directed to a Telegram user, registered through the Bot, and forwards them to the user's phone via the Internet. The gateway-hub of our prototype was implemented using a Raspberry Pi board.

More details of the messaging system can be found here [9].

III. Integration of the MQTT proxy

Data sensor collection is integrated within our platform through a dedicated service that is attached to the system as a specific client. The general idea is that sensors will collect the data, pack them as a structured piece of information, and send them to this service as a message. The receiving device, where the service is executing, will then (1) unpack the message, (2) build a proper MQTT message, and (3) publish it to the broker being used. The set-up required is the one shown in Figure 3. The sensor is attached to the hub using the WiFi link, as any other client. The device that will execute the *MQTTproxy* service has also to be connected 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.

Sensors are integrated in this platform using the REST interface. The sequence is basically the same used by regular clients: there is a first register phase followed by a "Push" phase. Figure 4 graphically describe this operation.

The transferred data is stored in a JSON object with the structure indicated in Listing1.

```
{
    'DEV_ID': 'lopy_374534',
    'QOS': 0,
    'TOPIC': 'sensor1/luminosity',
    'VALUE': 25
}
```

Listing 1: JSON structure of the sent message.

[3] https://telegram.org/

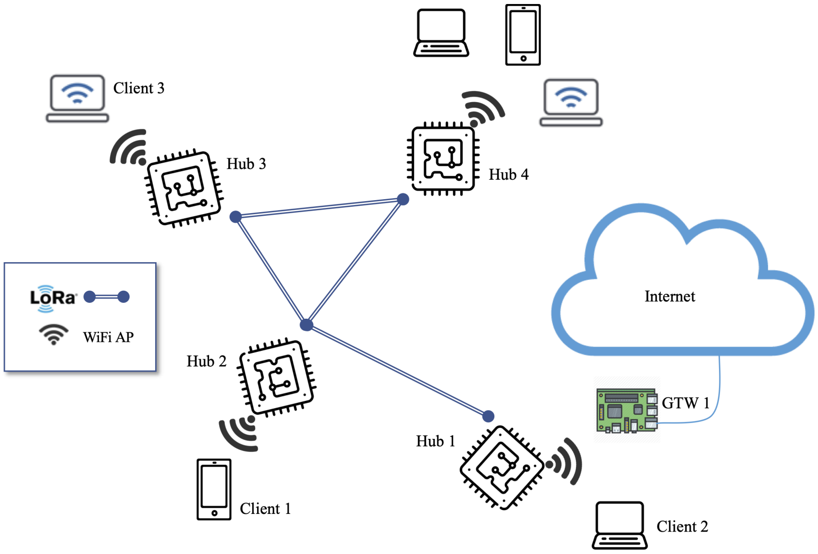
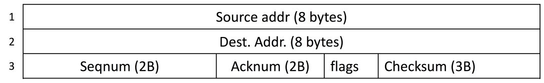


Fig. 1: Overall structure of the messaging platform.



1	Source addr (8 bytes)			
2	Dest. Addr. (8 bytes)			
3	Seqnum (2B)	Acknum (2B)	flags	Checksum (3B)

Fig. 2: Structure of the header of a packet used by our stop-and-wait ARQ.

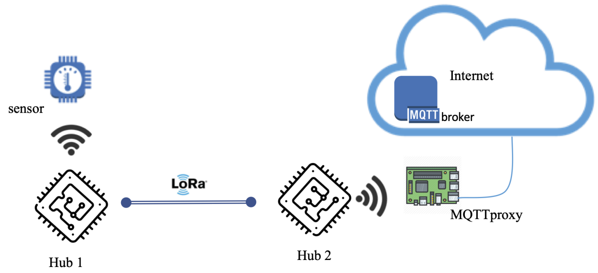


Fig. 3: The required set-up for the sensors and the MQTTproxy integration.

The data contained in the JSON object can have a variable size limited only by the slow data rate that we can obtain from a LoRa 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 with photo-pictures (e.g., taken by a camera supervising a crop) or a short video or audio registration. Multimedia content has to be previously encoded with anything as simple as the *Base64* algorithm. In the evaluation Section we will present the results with messages of up to 100kbytes to include all these cases.

The provision of the MQTTproxy is based on an "any-casting" approach. This means that there can be various MQTTproxys 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 packetize 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 interrogate the hub it is connected with to obtain the data. Figure 5 graphically describes this operation.

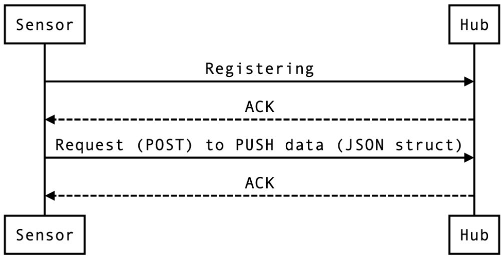


Fig. 4: The data flow when a sensor data is pushed to a hub.

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

IV. Experimental Results

This Section presents the results obtained with the proposed system by varying the distance between the hubs and the size of the sent messages; moreover, we compared the performance while using two different spreading factors, namely SF7 and SF12.

The devices we used as hubs were LoPy4 by Pycom[4].

[4] https://pycom.io/

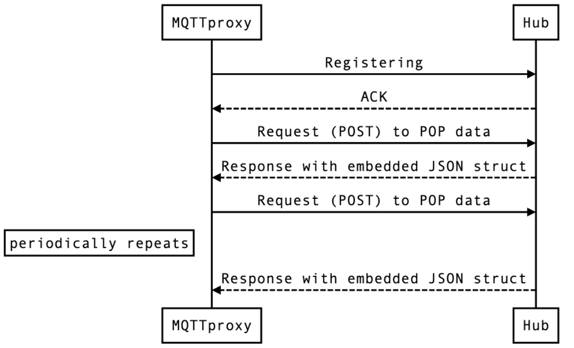

MQTTproxy
Hub
Registering
ACK
Request (POST) to POP data
Response with embedded JSON struct
Request (POST) to POP data
periodically repeats
Response with embedded JSON struct
MQTTproxy
Hub

Fig. 5: The MQTTproxy getting data from the hub.

The LoPy4 are a quadruple bearer MicroPython enabled development board with: IEEE 802.11b/g/n, Bluetooth v4.2 BR/EDR and BLE, LoRa (Semtech SX1276), and Sigfox with an Espressif ESP32 chipset (Xtensa dual–core 32–bit LX6), 520KB + 4MB of RAM, and 8MB of external flash. It has a dual processor and two antenna connectors: one for the 868 MHz band used by LoRA and another one for the 2.4 GHz band used by WiFi and Bluetooth. This allows fitting the antenna best suited for the application. For instance, high-gain directional antennas can be used to connect a rural village to a city that could be at a very long distance, using very small power by leveraging the spread spectrum features of LoRa modulation. The network processor handles the WiFi connectivity and the IP stack, while the main processor is entirely free to run the user application.

A prototype of the complete hub, made of a LoPy node with an omnidirectional antenna, and a 5 W solar panel, is shown in Figure 6.

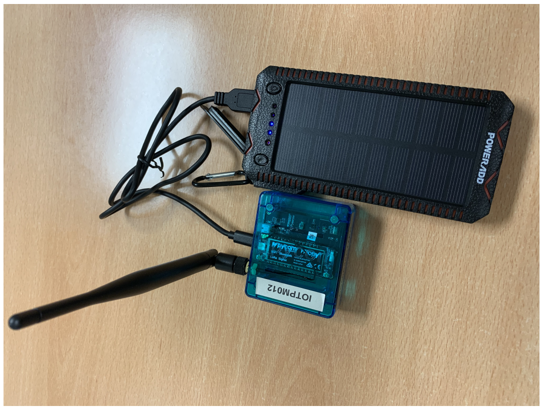


Fig. 6: A prototype of a complete hub.

To execute the MQTTproxy we used a Raspberry Pi 3 Model B+ that has a 64-bit quad core processor running at 1.4GHz, dual-band 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2/BLE, and an Ethernet port.

We considered the following distances between the two hubs: 1m, 100m, 750m, and 6000m. The 1 and 100 meters tests where performed in the facilities of the University. The 750m tests where performed in Valencia, in the "Ciudad de las Artes y las Ciencias" area, see Figure 7, 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 8 shows the scenario where we the tests were done; there is a clear line of sight between the two points that are 6km away, as shown in figure 8b.

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 both a Spreading factor of 7 (SF7) and a Spreading factor of 12 (SF12).

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. Retransmissions were rare events 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 message delivery.

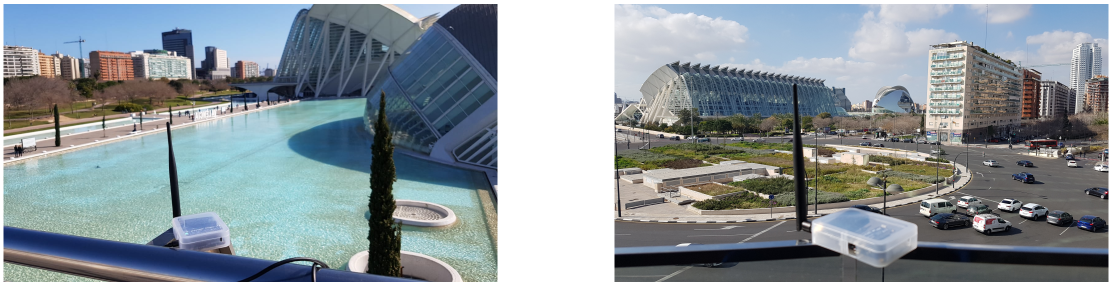
Figure 9 allows to better view the evolution of the STT as a function of the message size using a spreading factor SF7. As we can see, the STT clearly grows as the message size increases, while it is evident that the impact of distance is negligible. As expected, the overall throughput that LoRa offers us is quite low, in the order of 250 bps.

The following figures show the behavior of the STT by varying the distance between the two nodes. The average values are displayed. Figure 11 is obtained using a spreading factor SF7 and Figure 12 using a spreading factor SF12, in both cases messages of 1 kB, 10 kB, 50 kB, and 100 kB are sent. 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 retransmissions were required during the experiments.

From the above results we can conclude that our solution is an effective and stable solution to integrate data from long distances using LoRa and MQTT. For example, with messages of 100kBytes and SF7 the maximum delay obtained was 457,56 sec, and the minimum 451,49 sec; with SF12 the maximum delay obtained was 82,545 minutes, and the minimum 82,513 minutes. 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 requires, thus making this a frugal solution to a clear problem.

V. Conclusions

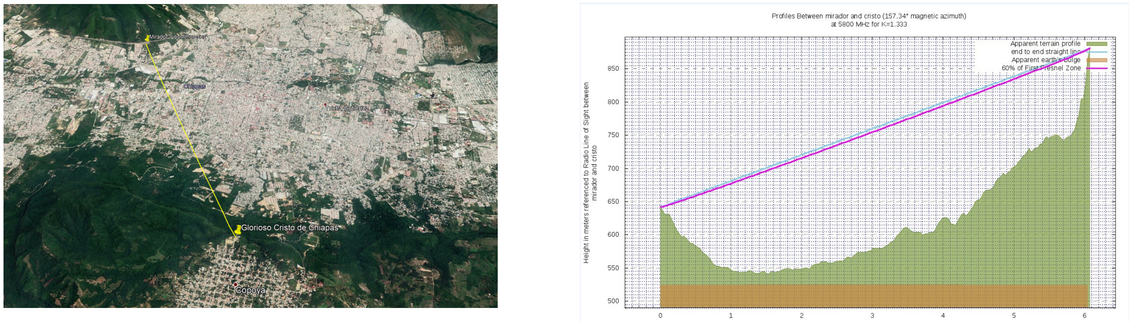
In this paper we presented a system that can collect data from remote sensors, transmit this data using LoRa to cover long distances, and automatically integrating it into a pub/sub



(a) View from the right-side hub.

(b) View from the left-side hub.

Fig. 7: Location in Valencia for the 750m tests.


Glorioso Cristo de Chiapas
Profiles Between mirador and cristo (157.34° magnetic azimuth)
at 5800 MHz for K=1.333
Apparent terrain profile
end to end straight line
Apparent earth's bulge
60% of First Fresnel Zone
Height in meters referenced to Radio Line of Sight between mirador and cristo
500
550
600
650
700
750
800
850
0
1
2
3
4
5
6
From mirador to cristo : 6.07 km

(a) Map view of the link trajectory. (Image obtained using Google Earth, ©2019 DigitalGlobe)

(b) Terrain profile between the two locations.

Fig. 8: The 6km tests location in Chiapas, México.

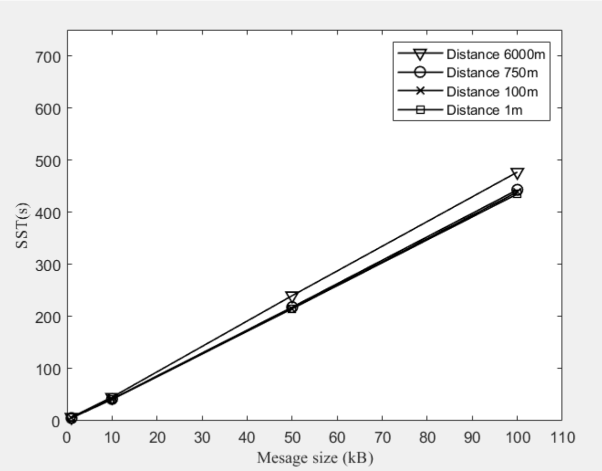


Fig. 9: Behavior of the STT when varying the message size (median values) using SF7

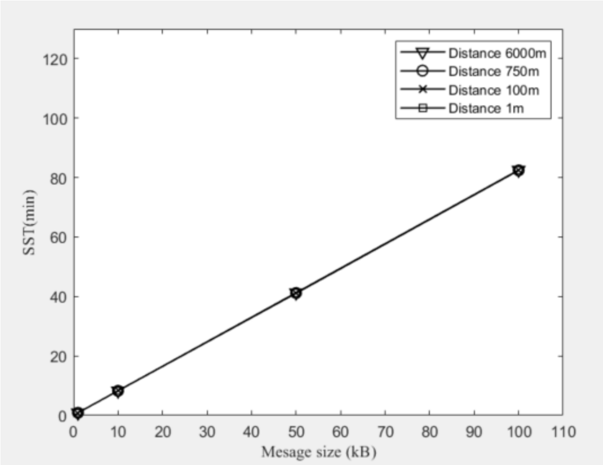


Fig. 10: Behavior of the STT when varying the message size (median values) using SF12

system based on MQTT. The goal is to provide support for IoT systems in rural areas connecting any type of sensor, like environmental sensors for weather forecasting, chemical sensor, level sensors, images sensors and so on. This system is embedded in a LoRa-based messaging system to conform a novel Frugal IoT platform.

We focused on the combination of ICT4D (ICT for development) with smart computing considered as how to design, build and use computing systems and technology for resource sustainability, where sustainable means developed to be long–lasting and environmentally responsible.

We evaluated our solution varying the size of the packets and the spreading factor, and compared the performance over various distances, showing that our solution is effective and stable, allowing us to integrate data from long distances using LoRa and MQTT. The current version of the used code is

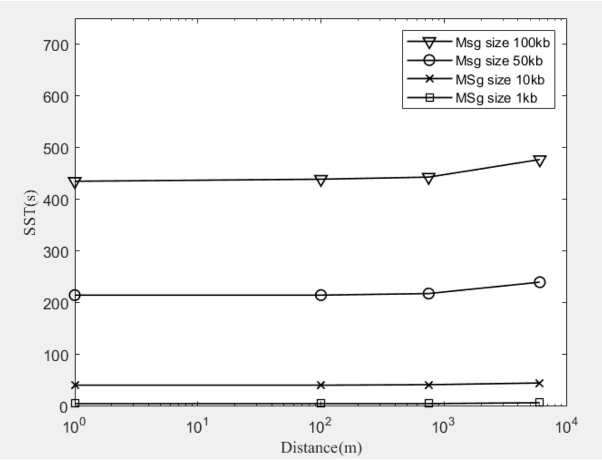


Fig. 11: STT versus distance between two nodes using SF7

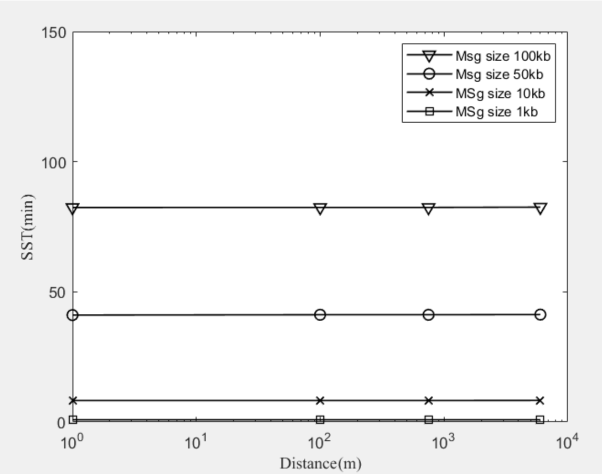


Fig. 12: STT versus distance between two nodes using SF12

available at: http://bit.ly/msnlora.

VI. 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. Crowcroft, A. Wolisz, and A. Sathiaseelan, “Towards an Affordable Internet Access for Everyone: The Quest for Enabling Universal Service Commitment (Dagstuhl Seminar 14471),” *Dagstuhl Reports*, vol. 4, no. 11, pp. 78–137, 2015. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2015/4971

[3] P. Micholia, M. Karaliopoulos, I. Koutsopoulos, L. Navarro, R. Baig Vias, D. Boucas, M. Michalis, and P. Antoniadis, “Community networks and sustainability: A survey of perceptions, practices, and proposed solutions,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3581–3606, Fourthquarter 2018.

[4] 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

[5] 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.

[6] N. Agarwal and A. Brem, “Frugal innovation-past, present, and future,” *IEEE Engineering Management Review*, vol. 45, no. 3, pp. 37–41, Third 2017.

[7] B. C. Rao, “Advances in science and technology through frugality,” *IEEE Engineering Management Review*, vol. 45, no. 1, pp. 32–38, First 2017.

[8] S. Ottosson, A. Moldavska, O. Ogorodnyk, and T. Skogsrød, “What is and how to develop sustainable innovation?” in *Leadership, Innovation and Entrepreneurship as Driving Forces of the Global Economy*, R. Benlamri and M. Sparer, Eds. Cham: Springer International Publishing, 2017, pp. 191–209.

[9] 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

[10] A. Banks, E. Briggs, K. Borgendale, and R. Gupta, “MQTT Version 5.0,” On-line, OASIS Standard, Tech. Rep., March 2019. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html