Dandelion: An Online Testbed for LoRa Development

Zheng Wang, Zhenqiang Xu, Baishun Dong, Weimin Xu, Jing Yang
School of software
Tsinghua university
Beijing, China
zwang17, xu-zq17, dbs18, xwm18, jing-yan18@mails.tsinghua.edu.cn

***Abstract*—LoRa has been shown as a promising technology for connecting millions for devices in the era of the Internet of Things. LoRa can provide long-distance (up to several kilometers) and low power communication with a modest data rate. However, developing LoRa protocols and testing LoRa systems face practical challenges compared with traditional networks such as Wi-Fi. For example, it requires network deployment in a wide area and even frequently updating the node program in the network for testing. To address those challenges, this paper presents Dandelion[1], an online testbed for LoRa development. Dandelion provides an easy-to-use interface for developing and testing LoRa applications. Dandelion consists of multiple LoRa nodes, gateways, central controller and web user interfaces. In Dandelion, LoRa nodes are deployed in different environments, each attached to a Raspberry Pi based edge node with out-of-band communication capability to collect system information. Dandelion also provides web based user interface to conveniently program each node, view the status of each node, update the program, etc. This significantly reduces the overhead for LoRa network development. We believe Dandelion can significantly reduce the costs of deployment, maintenance and evaluation of LoRa network. We show two examples of using Dandelion for LoRa network deployment. First, we measure packet reception rate of the LoRa packet with different parameters. Second, we do an RSSI-distance measurement to test and verify existing RSSI-distance LoRa model based on Dandelion.**

***Index Terms*—testbed, LoRa, remote control and management, RSSI**

I. Introduction

The Internet of Things is changing the lifestyle and has a significant impact on our everyday life. The communication technology is the basis for the application of the Internet of things. Long Range communication (LoRa) is proposed to be an important communication for the Internet of things. LoRa provides long range and low power communication for Internet of Things devices. LoRaWAN, specified by the LoRa Alliance in January 2015, is a relatively new protocol in the family LoRa. Despite its young age, the adoption of LoRaWAN has grown rapidly. LoRa has attracted more and more research interest recently [?], [?], [?], [?], [?], [?], [11], [20]–[23].

However, comparing to other wireless technologies [1], [14], [15], experiments on LoRa are very difficult because of its large communication range and high cost in deployment. Developing LoRa network protocol is also very challenging compared with traditional networks such as WiFi. For example, to test the network performance, we usually need to deploy the LoRa nodes in a wide area up to several kilometers. This introduces a high overhead for network deployment. Moreover, to test different protocols for the LoRa nodes, we need to update the program for those nodes. Existing LoRa network has a very low data rate and cannot support over-the-air reprogramming. Therefore, it even requires collecting all the deployed nodes and then reprogramming those nodes. In a word, developing LoRa based network system is very costly in practice.

A common practice for developing and evaluating network is to setup a testbed first [2]–[5]. There are also some testbeds concerning on LoRa [6]–[10]. But they are not flexible and lacks fine grain management or remote reprogramming. In this paper, we present Dandelion, an online LoRa testbed deployed to facilitate LoRa network development. Dandelion consists of multiple LoRa nodes, gateways, a central controller and web user interfaces. Our node is based on the SX1276 chip from Semtech. In Dandelion, LoRa nodes are deployed in different environments, each connected to a raspberry based edge node with out-of-band communication capability to collect network information. The edge node can collect information from the connected LoRa node. Currently, each raspberry based edge node has mobile cellular communication to transmit LoRa node information back. Those edge nodes can also provide diagnostic information for each node especially when nodes encounter problems. Meanwhile, those edge nodes act as the daemon for each LoRa node. We can reprogram and control each LoRa node through the edge nodes. Dandelion also provides web based user interface through which users can program each node, view the status of each node, update the program, control each node, etc.

The design of Dandelion has the following three advantages:

- Dandelion is designed to reduce the costs of deployment, maintenance and evaluation of LoRa network. Users can focus on their program on LoRa nodes and do not need to spend a long time to deploy a network for testing. Dandelion avoids the costs and difficulties involved in setting-up and maintaining a large number of LoRa nodes.
- Dandelion provides remote LoRa node reprogramming. Nodes in Dandelion are distributed in different places and can be reprogrammed remotely.

[1]Dandelion is running at http://thulpwan.net

978-1-7281-5212-7/19/$31.00 ©2019 IEEE
DOI 10.1109/MSN48538.2019.00089

- Dandelion provides easy-to-use user interface and makes a rational resource allocation. It effectively helps researchers to do the LoRa experiments.

We show two examples of using Dandelion for LoRa network deployment. First, we measure packet reception rate of the LoRa packet in different parameters. Second, we do a RSSI-distance measurement to test and verify existing RSSI-distance LoRa model based on Dandelion.

The rest of the paper is organized as follows. In Sec. II, we present a detailed description of Dandelion and then show the Implementation of Dandelion. Sec. III shows a case study of using Dandelion. Sec. IV compares Dandelion against other testbeds. Sec. V concludes this work.

II. System Design

A. Architecture

Fig. 1 shows the architecture of Dandelion. From the perspective of the composition of hardware and software, Dandelion includes edge nodes, gateways, cloud servers and web interfaces. The network mainly consists of two parts. One is a traditional LoRaWAN network, which can be used for data collection, upload, forwarding, and storage. The other is the control platform part. This part of the testbed mainly provides functions like reprogramming of the nodes, reconfiguration of the gateway, resource scheduling on the server, and user management.

1) LoRa component: The LoRa component includes LoRa nodes, gateways [12] and servers [13]. The hardware of our gateways and nodes is shown in Fig. 2. The gateway combines the SX1301 chip and a Raspberry Pi. The node consists of the SX1276 chip and the STM32L053 chip. At present, we have deployed gateways at the top of three different buildings. In Fig. 1, we use blue arrow to indicate the communication flow of LoRaWAN which works with a star topology. Specifically, multiple LoRa nodes connect to one or multiple gateways using LoRa signals while gateways transmit the collected data to the LoRa backend servers via traditional wireless technology like TCP/IP. Furthermore, each LoRa node is connected to a raspberry Pi-based edge node. The edge nodes connect to the central server via an out-of-band communication, such as cellular networks. Therefore, the central server can send commands to the edge nodes thus controlling these LoRa nodes. The central server can also collect data from the edge nodes. To reduce deployment overhead further, we explore an alternative approach which uses LoRa's own signals as the control signal and reprogram the LoRa nodes. It should be noted that both of the control and data collection parts are separated from the LoRa network. Therefore, even if the LoRa network encounters some problems, the developer can still control the network and collect data for diagnosis.

2) Control platform: Two sets of control schemes are designed here to deal with different application scenarios.

At the bottom left of the figure, the central server receives the instructions from the Web server and then passes the instructions to the indoor nodes. Since the edge nodes and the LoRa node are connected by a USB interface, the control instructions are further transmitted to the LoRa node, thus achieving the purpose of the control node.

3) User management: The user management system is used to provide external interface services. In the test platform, the web page is the client system of the testbed. All users are required to register for authentication. The online web interface provides function like restarting nodes, downloading data, viewing logs and burning functions for remote nodes. The online website also provides remote configuration functions for some gateways, which is convenient for users to modify the receiving frequency of the gateway. The online website also provides the ability to view data packets in real-time.

B. Implementation

The framework used in front-end system development is based on Ant design. The implementation is based on React for the web interfaces. The back-end implementation uses Java's Spring, SpringMVC, and MyBatis frameworks, the SSM framework. The building project uses Maven tools to integrate Spring, SpringMVC, and Mybatis via pom.xml. The following briefly introduces the role of three parts. The central server is written by node.js, with the socket.io package. The serial communication in raspberry Pi edge nodes is also written by node.js. The code in the LoRa node is written in C and we provide samples for users to download. The program for the LoRa node is the same as other normal programs on the LoRa node.

The Raspberry Pi based edge nodes are supplied by USB-hub power. The LoRa node uses the LoRa chip is SX1276/77/78/79, and the gateway uses chip SX1301 and Ublox8 GPS chip. The LoRa nodes and edge nodes are deployed in different apartments and buildings, and the gateways are deployed at the top of several high buildings.

C. Remote programming

It is necessary for Dandelion to reprogram the outdoor node, our testbed designs an incremental programming method. In general, there exists a high similarity between the file of the updated node and the new file. Thus incremental updates are usually used [16], [17]. Our incremental method is based on R2 algorithm [18], [19]. LoRa is supposed to be a wide range communication technology. For most case, there is usually no commonly used network signals such as Wi-Fi. We design an energy-efficient remote reprogramming protocol for LoRa in the low-bandwidth situation. When the node receives the message from the port, it enters the CLASS C mode to continuously receive the data packet from the server. When the relocation table is updated, a new entry is calculated according to the difference and the old relocation entry and is merged with the directly added relocation entry to obtain a new relocation table. The user program is updated according to the editing script, and the data is copied from the old program or the update file according to the ADD or COPY instruction, and a new program is generated according to the relocation table to modify the address of the partial location. With the help

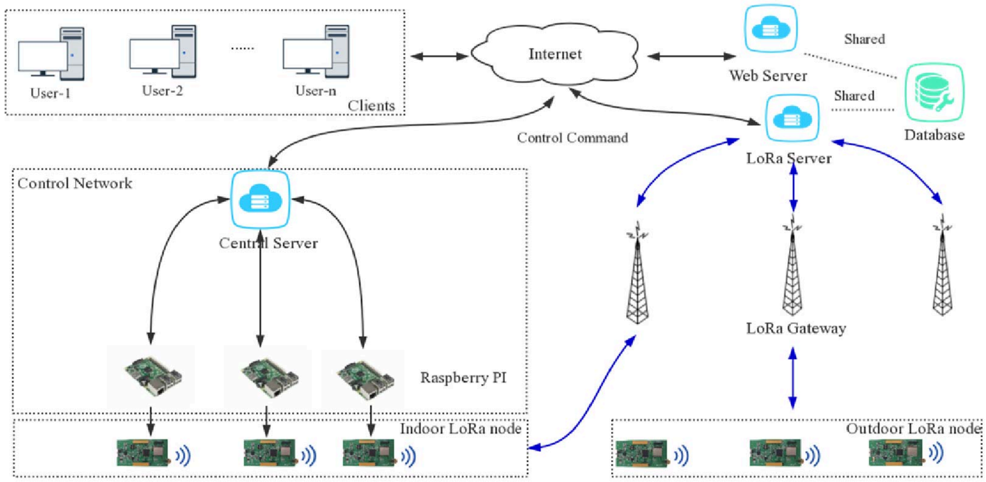

Fig. 1. The architecture of testbed, including web server, control server and raspberry based edge node.

Fig. 2. Hardware of our nodes and gateways. We have deployed gateways in three different places.

of incremental updates, we could reprogram outdoor nodes efficiently.

III. Use case: measurement of LoRa

In this section, we show two examples using Dandelion. We first place several LoRa nodes to measurement the packet loss of LoRa network. Then we collect the data and evaluate the correlation between RSSI and distance in the LoRa network using the existing RSSI-distance model.

A. Packet reception rate

As shown in the Fig. 3, we put the LoRa node in different scenarios. Point 4 is 500 meters away from the gateway and there are many trees nearby. Point 1, point 2 and point 3 are about 1000 meters away from the gateway. There are many obstacles near point 1. The obstacles near point 2 and point 3 are less than other points. Point 5 is about 1200 meters far from the gateway and is blocked by many trees and buildings. Table I shows the measurement result.

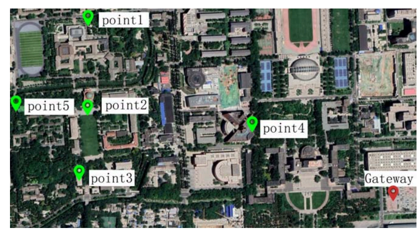

Fig. 3. Packet reception rate test.

TABLE I
Packet reception rate in different SF and situation

	SF7	SF8	SF9	SF10
point 1	0.09	0.12	0.47	0.54
point 2	0.44	0.98	0.97	0.98
point 3	0.86	1.0	0.98	0.99
point 4	0.4	0.67	0.79	1.0
point 5	0.28	0.66	0.79	0.33

We can see that low SF reduces the packet reception rate of LoRa packet, especially SF7 which only supports very short distance. The trees and buildings also affect PRR in some cases. For point 5 in SF10, we get an abnormal result. This may be due to some other factors that affect the experiment result.

B. RSSI of LoRa

RSSI (Received Signal Strength Indication) is related to the distance between the signal source and the receiver by the strength of the received signal.

We collect data from LoRaWAN deployed on campus and draw a coverage map. Moreover, we measure and compare the prediction results of different models for the LoRa link.

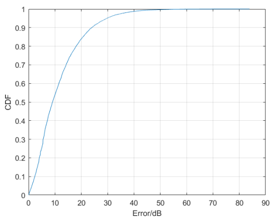

Fig. 4. Bor model

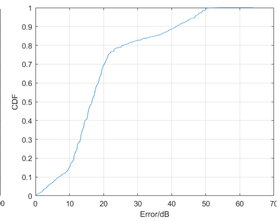

Fig. 5. Free space model

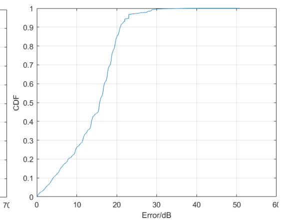

Fig. 6. Suburban model

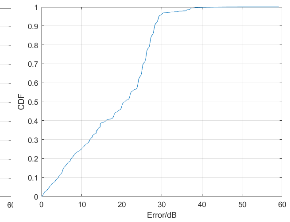

Fig. 7. Urban model

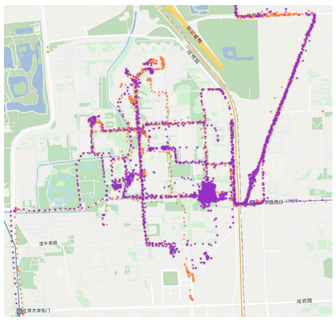

Fig. 8. It's a Coverage map of our testbed in our campus. The purple and orange points means different groups of data.

We compare the results of different models for LoRa link evaluation and analysis of the results.

Our goal is to measure the link quality for LoRa with different environments and different gateways on campus. Then we compare different existing link quality models.

C. Modeling

We mainly focus on three different models, i.e. free-space model, log-normal shadowing model and Okumura-Hata model based on the collected data.

1) Free space model: The free space model means the ideal propagation conditions that do not have any blocking and multi-path. The free-space propagation model is the simplest model of radio wave propagation and the loss of radio waves is only related to the propagation distance and the frequency of the radio wave. The classic FSPL(free space path loss) is given by the equation:

$$P^{rx} = P^{tx} + G^{tx} + G^{rx} – FSPL \quad (1)$$

$$FSPL = 20log10(d) + 20log10(f) – 27.55 \quad (2)$$

P^{rx} represent expected received power (RSSI), P^{tx} represent transmission power, G^{tx} and G^{rx} are transmitting and receiving antenna gains. FSPL represents a free-space path loss. For open spaces, the most accurate link model is given by the equation.

2) Bor model: Bor builds on top of a more realistic model called the log-normal shadowing model [24]. to analyze LoRa links. This is also known as the log-normal shadowing model. The shadow fading is determined by the environment in which the receiver is located, such as mountains and buildings nearby. By expressing this shadow effect in a model, a log-normal distribution is a suitable description. The Bor model is given by the equation:

$$PL[dB] = PL(d_0) + 10 * n * log10(d/d_0) + X_\sigma \quad (3)$$

$$P^{rx} = P^{tx} + G^{tx} + G^{rx} – PL[dB] \quad (4)$$

where d represents the distance from the transmitter. PL(d_0) means path loss at d_0, n is the path loss exponent of the environment, and σ is the standard deviation of a zero-mean Gaussian random variable X.

3) Okumura-Hata model: This basic modelis then adapted with a correction factor a(h_m) depending on the environment where the communication link is placed. Considering that we have two land-cover macro-classes, we use the two equations (urban and suburban) that best match the Okumura-Hata set [25].

$$L_U[dB] = 69.55 + 26.16lg(f) – 13.82lg(h_b) – a(h_m) +$$

$$(44.9 – 6.55lg(h_b))lg(d)$$

For urban model:

$$a(h_m)[dB] = (1.1lg(f) - 0.7)h_m - (1.56lg(f) - 0.8)$$

For suburban model:

$$L_{SU}[dB] = L_U - 2(lg(f/28))^2 - 5.4$$

D. Test setup

As stated earlier, we have deployed three gateways in different buildings in our university. All gateways are connected with Dandelion. In this experiment, we choose two of the gateways to collect data. The total measurement area can be seen in Fig. 8. The two different colors indicate data collected by the two gateways. The spread factor of the LoRa packet is from SF7 to SF12. The CR is 4/5 and tx power is 20 dBm and bandwidth is 125kHz.

We use our Dandelion to burn several LoRa nodes and give the node to some students. Then students hold the LoRa node and travel around the campus. Finally, we collect the data in our cloud server database.

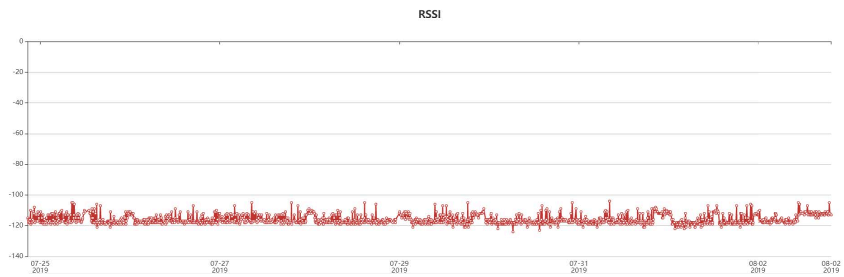

Fig. 9. Single node outside data in one month.

E. Measurement result

In this section, we present the measurement result. Fig. 9 presents the web page that shows the SNR of data for a single node in one month. Fig. 10 shows all data we collect by the two gateways. Different color indicates different RSSI values. From color red to color green, the signal strength is going to be stronger. We can see that the green region is the place where we deploy the gateway. The size of our dataset is 31263. The result is easy to understand as RSSI declines like the free space situation.

F. Analysis

The 50% error of the Bor model is 9dB, as shown in Fig. 4. It is about 6% deviation of the average RSSI and it is a weak acceptable result. From Fig. 5 to 7, we can see that the free space, Ohata suburban model and urban model are inaccurate. The 50% of error is 16, 16 and 20 dB.

The real RSSI data and fitted lines drawn by these models are shown in Fig. 11. The free space model is far from real data. Bor model can better predict the correlation between RSSI and distance. Hata model is not suited for any environments, for different devices and bandwidth settings. In our scene, the Bor model is the best. The environment in the urban settings is so complex that RSSI is not a good index to judge distance.

Fig. 10. Gateway 1 and 2 measurement of RSSI. Changing from red to green means that the signal strength gradually becomes stronger.

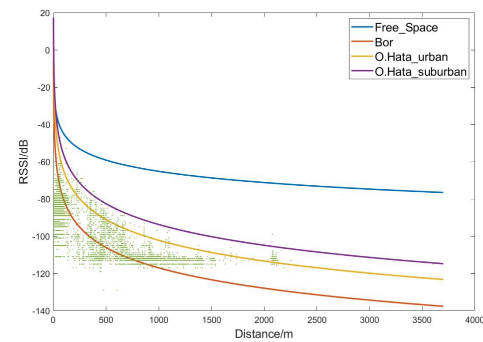

Fig. 11. RSSI-Distance with different model. The green points is the real data.

IV. Related work

In this section, we compare Dandelion with the existing testbed.

OpenChirp [10] is a LoRa related platform. It is used in a low-power WAN management framework that provides data context, storage, visualization and access control over the Web. OpenChirp is currently supporting Carnegie Mellon University's LoRaWAN network, which also spans a large area. Users can easily add devices or LoRa gateways to the managed network, or they can generate their independent instances.

AliOS Things [26] providesThings provides big data storage and search, data visualization. The AliOS Things uses the uDevice center which is a flexible IoT multi-device platform. The platform moves the IoT equipment online so that it becomes IoT ECS. And uDC provides abundant information for the multi-device complex network development. Users only need to download the software to manage the network. It provides visualization of programming node, and Dandelion provides, too. Our system adopts the same idea that users do not need to see the real hardware.

Indriya [3] is deployed at the National University of Singapore. It uses TelosB devices and it is built on an active-USB infrastructure. It is also a large-scale testbed and has been used by over 100 users.

The MoteLab [2] is deployed at Harvard University. MoteLab is a set of software tools for managing a testbed of ethernet-connected sensor network nodes. It contains a central server that handles scheduling, reprogramming nodes, logging data and providing a web interface to users. We take MoteLabs as a reference that we build a similar central server that provides reprogramming nodes and downloading data.

V. Conclusion

In this paper, we introduce Dandelion, a LoRa experimental testbed, which works in a large-scale and realistic environment. It effectively helps researchers to do the LoRa experiment in the real world. We demonstrate how to use the testbed to measure LoRa network performance and validate popular channel models for LoRa. We argue that our contribution is key for future LoRa development and deployment.

References

[1] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11 b mesh network. In *ACM SIGCOMM Computer Communication Review*, volume 34, pages 121–132. ACM, 2004.

[2] Geoffrey Werner-Allen, Patrick Swieskowski, and Matt Welsh. Motelab: A wireless sensor network testbed. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 68. IEEE Press, 2005.

[3] Manjunath Doddavenkatappa, Mun Choon Chan, and Akkihebbal L Ananda. Indriya: A low-cost, 3d wireless sensor network testbed. In *International conference on testbeds and research infrastructures*, pages 302–316. Springer, 2011.

[4] Asif M Yousuf, Edward M Rochester, and Majid Ghaderi. A low-cost lorawan testbed for iot: Implementation and measurements. In *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pages 361–366. IEEE, 2018.

[5] Anne-Sophie Tonneau, Nathalie Mitton, and Julien Vandaele. A survey on (mobile) wireless sensor network experimentation testbeds. In *2014 IEEE International Conference on Distributed Computing in Sensor Systems*, pages 263–268. IEEE, 2014.

[6] Jorge Navarro-Ortiz, Juan J Ramos-Munoz, Juan M Lopez-Soler, Cristina Cervello-Pastor, and Marisa Catalan. A lorawan testbed design for supporting critical situations: prototype and evaluation. *Wireless Communications and Mobile Computing*, 2019, 2019.

[7] Jaco M Marais, Reza Malekian, and Adnan M Abu-Mahfouz. Lora and lorawan testbeds: A review. In *2017 Ieee Africon*, pages 1496–1501. IEEE, 2017.

[8] Roman Trüb, Reto Da Forno, Tonia Gsell, Jan Beutel, and Lothar Thiele. Demo abstract: A testbed for long-range lora communication. In *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 342–343. IEEE, 2019.

[9] Pierre Neumann, Julien Montavont, and Thomas Noël. Indoor deployment of low-power wide area networks (lpwan): A lorawan case study. In *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8. IEEE, 2016.

[10] CMU. OpenChirp. https://openchirp.io.

[11] Kai-Hsiang Ke, Qi-Wen Liang, Guan-Jie Zeng, Jun-Han Lin, and Huang-Chen Lee. Demo abstract: A lora wireless mesh networking module for campus-scale monitoring. In *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 259–260. IEEE, 2017.

[12] Risinghf corporation. http://www.risinghf.com.

[13] LoRaServer. https://www.loraserver.io.

[14] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, Maria Kazandjieva, David Moss, and Philip Levis. Ctp: An efficient, robust, and reliable collection tree protocol for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 10(1):16, 2013.

[15] Carlos Cordeiro and Kiran Challapali. C-mac: A cognitive mac protocol for multi-channel wireless networks. In *2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 147–157. IEEE, 2007.

[16] Rajesh Krishna Panta, Saurabh Bagchi, and Samuel P Midkiff. Efficient incremental code update for sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 7(4):30, 2011.

[17] Wei Dong, Yunhao Liu, Chun Chen, Lin Gu, and Xiaofan Wu. Elon: Enabling efficient and long-term reprogramming for wireless sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 13(4):77, 2014.

[18] Wei Dong, Yunhao Liu, Chun Chen, Jiajun Bu, Chao Huang, and Zhiwei Zhao. R2: Incremental reprogramming using relocatable code in networked embedded systems. *IEEE Transactions on Computers*, 62(9):1837–1849, 2012.

[19] Sandeep Kulkarni and Limin Wang. Energy-efficient multihop reprogramming for sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 5(2):16, 2009.

[20] Marco Cattani, Carlo Boano, and Kay Römer. An experimental evaluation of the reliability of lora long-range low-power wireless communication. *Journal of Sensor and Actuator Networks*, 6(2):7, 2017.

[21] Juha Petajajarvi, Konstantin Mikhaylov, Antti Roivainen, Tuomo Hanninen, and Marko Pettissalo. On the coverage of lpwans: range evaluation and channel attenuation model for lora technology. In *2015 14th International Conference on ITS Telecommunications (ITST)*, pages 55–59. IEEE, 2015.

[22] Dmitry Bankov, Evgeny Khorov, and Andrey Lyakhov. On the limits of lorawan channel access. In *2016 International Conference on Engineering and Telecommunication (EnT)*, pages 10–14. IEEE, 2016.

[23] Peter J Radcliffe, Karina Gomez Chavez, Paul Beckett, Justin Spangaro, and Conrad Jakob. Usability of lorawan technology in a central business district. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pages 1–5. IEEE, 2017.

[24] Martin C Bor, John Vidler, and Utz Roedig. Lora for the internet of things. In *EWSN*, volume 16, pages 361–366, 2016.

[25] Masaharu Hata. Empirical formula for propagation loss in land mobile radio services. *IEEE transactions on Vehicular Technology*, 29(3):317–325, 1980.

[26] Aliosthings. https://aliosthings.io.