A 2-hop LoRa Approach Based on Smart and Transparent Relay-Device

Mamour Diop[1,2], Congduc Pham[1], and Ousmane Thiaré[2](✉)

[1] University of Pau, Pau, France
{mamour.diop,congduc.pham}@univ-pau.fr
[2] Gaston Berger University, Saint-Louis, Senegal
{mamour.diop,ousmane.thiare}@ugb.edu.sn

Abstract. LoRa is designed for long-range communication where devices are directly connected to the gateway, which removes typically the need of constructing and maintaining a complex multi-hop network. Nonetheless, even with the advantage of penetration of walls, the range may not sometimes be sufficient. This article describes a 2-hop LoRa approach to reduce both packet losses and transmission cost. To that aim, we introduce a smart, transparent and battery-operated relay-device that can be added after a deployment campaign to seamlessly provide an extra hop between the remote devices and the gateway. Field tests were conducted to assess relays' ability to automatically synchronize to the network without advertising their presence.

Keywords: LoRa · Low-power IoT · Low-cost IoT · Multihop · Rural area

1 Introduction

Recently, Low-Power Wide Area Networks (LPWAN) play a key role in the IoT maturation process. This is a broad term for a variety of technologies enabling power efficient wireless communication over very long distances. For instance, technologies based on ultra-narrow band modulation (UNB) – e.g. SigFox™ – or Chirp Spread Spectrum modulation (CSS) – e.g. LoRa™ [1] – have become de facto standards in the IoT ecosystem. Most of LPWAN technologies can achieve more than 20 km in line of sight (LOS) condition and they definitely provide a better connectivity answer for IoT by avoiding complex and costly relay nodes to be deployed and maintained.

In the context of the H2020 WAZIUP project, we developed a low-cost IoT generic platform using LoRa technologies to enable the deployment of smarter rural applications in developing countries [2–4]. From this generic platform, significant real-world deployments have already been realized in Senegal (Cattle Rustling), Ghana (Fish Farming, AGRI-Weather) and Pakistan (AGRI-Soil with multi-level soil moisture for crop irrigation). The feedback we have with these

G. Mendy et al. (Eds.): AFRICOMM 2018, LNICST 275, pp. 24–34, 2019.
https://doi.org/10.1007/978-3-030-16042-5_3

rural deployment experiences is that even with the longer range offered by LoRa, we encountered in many of these deployment campaigns connectivity issues with the gateway: there is no or very weak connectivity. In fact a clear LOS communication is hardly the case. Reasons are numerous, for instance there are constraints on gateway and gateway's antenna placement – e.g. in the farm office with power supply and wired Internet access – and some devices can become very isolated from the vast majority of deployed devices even when device's antenna can be placed higher than the device itself. Regarding the transmission power, there are also limitations in many countries. But it is not always desirable to use higher transmission power levels as it would result in severe energy consumption and reduced battery lifetime.

In this paper, we investigate a 2-hop LoRa approach to extend the coverage area and solve these connectivity issues of real-world IoT deployment in rural environments. Very importantly, the main objective is to design a smart, transparent and battery-operated intermediate node – *relay-device* – that can be added after a deployment campaign to seamlessly provide an extra hop between the remote devices and the gateway. That can significantly improve reliability of data transmission in non-line of sight (NLOS) scenarios. The remainder of the paper is organized as follows. Section 2 provides an analysis to the LoRa technology multi-hop schemes proposed in the literature. Section 3 describes the proposed approach based on low-power relay nodes. Performance evaluation and measurement results are discussed in Sect. 4. We conclude in Sect. 5.

2 Related Work

In a multi-hop network, every node can communicate with the other nodes. They provide routing for each other so that two nodes physically far away from each other can communicate using nodes between them. Multi-hop alternatives for the uplink in LPWANs technologies – particularly in LoRaWAN – have not yet been profoundly explored in networks operating at sub-1 GHz. In exploring the limits of LoRaWAN, the authors in [5] addressed the use of TDMA and multi-hop solutions in order to reduce both the number of collisions and the needed transmission power. From there, an extension of LoRaWAN protocol enabling relay-based communication to extend coverage area without the need of gateways and increase the performances of end-devices, is designed in [6]. LoRaBlink [7] is a protocol on top of LoRa's physical layer designed to support reliable and energy efficient multi-hop communications. Time synchronization is used to define slotted channel access. While downlink messages are distributed through flooding, nodes use a directed flooding approach for uplink communications.

In [8], the author analyzed the impact of introducing a forwarder node between an end device and a gateway to improve the range and quality of LoraWAN communications. As the forwarder aims to reduce the power consumption on end-nodes, the work mainly focused on an energy analysis. However, the device receive window must be increase to manage downlink packets. In [9], authors investigated the combination of LoRa and concurrent transmission

(CT) – a recently proposed multi-hop protocol that can significantly improve the network efficiency – to realize a reliable CT-based LoRa multi-hop network. On the one hand, the long transmission range of LoRa ensures the indoor coverage, reduces the number of redundant relay nodes, and keeps the transmission power small. On the other hand, the CT protocol helps to realize a simple but efficient one-to-any fast packet broadcast by introducing the synchronized packet collisions. [10] proposed a multi-hop uplink solution compatible with LoRaWAN specification, which can act as an extension to already deployed gateways. End nodes transmit data messages to intermediate nodes, which relay them to gateways by choosing routes based on a simplified version of DSDV routing.

These works propose centralized approaches controlled by the gateway, the network server or the initiator, which sets up both the relays and the devices through MAC commands. In addition, the synchronization mechanism requires message exchanges. In most of these works, end-devices act as relay depending on the needs. [10] introduces routing nodes (RNs) for relaying uplink packets from leaf nodes. However, RN are assumed not energy constrained. The purpose of this work is not to use the multi-hop concept to propose a new LPWAN protocol or an extension of LoRaWAN, to solve the aforementioned problems. In rural applications context for developing countries, gateways cannot act as relays as in [11] where more gateways are deployed to ensure multi-hop communication. This would lead to additional deployment cost since a gateway (a) is considered to be appropriately placed close to an unlimited power source, (b) requires an IP connection to operate and (c), is the most expensive component, even in our low-cost context. End-devices also don't act as relays because they run very specific sensing template code and must be placed according to sensing needs.

3 Smart 2-hop Relaying Mode

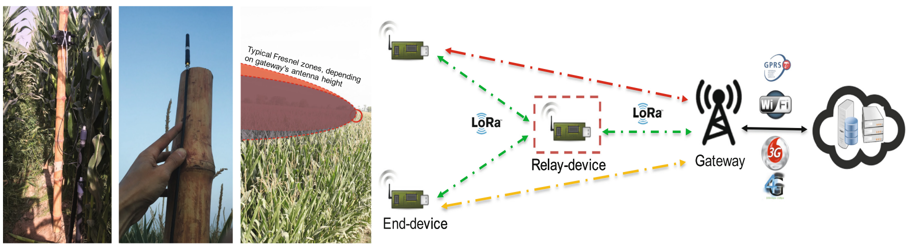
3.1 Principle

Our 2-hop LoRa relay approach consists, in a post deployment addition, of an extra hop between some end-devices and the gateway in NLOS scenarios as illustrated in crop fields for the Nestlé WaterSense project (the left part of Fig. 1). We propose to have *relay-devices* which are special low-power nodes different from the end-devices. However, similar to the end-devices, relay-devices are built from the generic hardware IoT platform but their unique feature is to extend the network coverage by performing data receive and forward operations. It does not take part in any data sensing, data processing nor aggregation tasks. One of the major considerations of a relay-device should be its appropriate location to cover areas where connectivity is either lost or unstable after the network deployment. We designed the relay-device with the following requirements:

- *Low power*: relay-devices are battery-powered and therefore energy constrained. Their hardware should be very similar to those used by low-cost end-devices (i.e. Arduino Pro Mini). Being battery-operated they must not listen continuously, which basically would make them gateways and this is not what we wanted.

- *Smart*: relay-devices must be designed to remain in low-power mode most of the time. Obviously, they have to wake-up at appropriate moments to catch uplink transmissions from specific devices in order to perform the relay operation. This is the major consideration of this work since missing uplink packets would make the network less reliable than it was. Therefore, a relay-device must be able to switch from sleep to active mode by smartly analyzing the uplink pattern from end-devices.
- *Transparent*: relay-device nodes must be transparent to the rest of the network: (*a*) no change in hardware or software for end-devices or gateway to support the new 2-hop approach; (*b*) no additional signaling traffic between relay-devices and end-devices or gateway. Therefore, end-devices should not be aware of the 2-hop relay mode, nor to perform any discovery and binding process to a nearby relay-device. A relay-device also does not need to exchange parameters with the gateway for advertising its presence. And, on the gateway side, no scheduling mechanism for end-devices and relay-devices is required. The presence of a relay-device should not be detected although it is possible to indicate its presence with a specific flag in the packet header if it is desirable for the gateway (or network server) to have this information. Our approach is not centralized, neither at gateway nor network server as in related works. Furthermore, withdrawal or failure of a relay-device leaves the network as functional as before its integration in the network.

Figure 1 (right part) depicts our proposed architecture for providing a transparent 2-hop LoRa connectivity. The red link means no direct connectivity while the orange link means unstable connectivity. The green links means high quality, stable connectivity. The main advantage of our smart relaying mode is related to the relay-devices' ability to adapt in complete autonomy and transparency to their deployment environment. This is realized with an autonomous and asymmetric synchronization approach. It does not require any time synchronization between the nodes, e.g. end-devices behavior remain unchanged as indicated previously. It is asymmetric in the sense that the synchronization work is done by the relay-device: only the relay-device has to learn wakeup periods of the end-devices.



Deployment with non-LOS conditions   *Long-range 2-hop connectivity architecture*

Fig. 1. Long-range 2-hop connectivity architecture (Color figure online)

3.2 Implementation

In a typical telemetry LoRa network, end-devices periodically measure environmental parameters and transmit data packets mostly at regular intervals, being most of the time in deep sleep mode where they are unable to send nor receive packets. We assume here that end-devices wake-up at least once every 60 min – from their local time as there is no synchronicity between end-devices. When inserted in an existing LoRa network, relay-devices are responsible for forwarding data packets from end-devices with no prior knowledge of how end-devices will wake up. Once deployed, a relay-device discovers end-devices in its vicinity and will build a wake-up table. When powered-on a relay-device first runs an observation phase and then a data forwarding phase.

Observation Phase. This phase consists in observing network traffic for a specified duration. At start-up a relay-device usually does not know when it will receive an uplink packet, so it needs to be in receive mode during all the observation duration. This observation duration must be long enough to catch the various uplink packets from end-devices. Assuming that end-devices wake-up at least once every 60 min, an observation duration longer than 60 min is sufficient. The Arduino Pro Mini running at 3.3 V consumes about 15 mA in continuous receive mode, so 60 min of observation has little impact on the battery lifetime as this process is only performed on startup.

Algorithm 1. Observation stage

```
Input:
  obs_duration: appropriate time interval for observing the network traffic in order
  to synchronize with it.
  Bound_Devices: an array of bounded devices to the relay device, sorted in receiving
  packet order.
while obs_duration ≥ current_time do
  pkt ← Receive_Pkt()
  if pkt ≠ Null then
    if pkt.Src ≠ Gateway_id then
      //it's an uplink data
      if not is_bound(pkt.Src) then
        //it's the first reception for this device
        Record_Device(Bound_Devices, pkt.Src, timestamp)
      else
        /*send back downlink message to pkt.Src device, if present. Then delete
        it */
        Send_Back_Downlink_Msg(Bound_Devices, pkt.Src)

        /*update pkt.Src device in Bound_Devices */
        Update_Device(Bound_Devices, pkt.Src, timestamp)
      end if
      /*forward the received packet to the gateway by keeping the original packet
      header */
      Forward_Data(pkt)

    else
      //it's a downlink message
      if is_bound(pkt.Src) then
        Store_Msg(Bound_Devices, pkt.Src, pkt.Msg)
      else
        Record_Device_Msg(Bound_Devices, pkt.Src, timestamp, pkt.Msg)
      end if
    end if
  end if
end while
```

In the observation phase a relay-device receiving an uplink packet from an end-device (a) sends to the device any cached downlink packet; (b) records relevant information of the uplink packet such as the source address, the timestamp, etc.; (c) forwards the packet to the gateway by keeping the original packet header. Note that packet forwarding from a relay-device to the gateway during this phase can also allow for transmission quality comparison if the original packet also reach the gateway. Note that the relay-device can also receive downlink packets from the gateway to specific devices. In this case, the relay-device stores this downlink packet and will forward it at the next uplink transmission from the corresponding end-device. The reason to do so, instead of directly forwarding the downlink packet to the device is because the device receive window probably does not take into account the additional delay introduced by the relay-device. This process, detailed in Algorithm 1, is repeated throughout the observation duration. When the observation phase is over the relay-device switches to the data forwarding phase.

Data Forwarding Phase. With the collected information during the observation phase, the relay-device is now able to determine wakeup time of the end-devices in its vicinity. It can determine its own activity schedule in each round to wakeup at appropriate moment to forward uplink packets and remain in low-power mode the rest of the time. Algorithm 2 shows how the sleep period is computed. In the data forwarding phase the relay-device determines the wakeup time T (using *sleep_period*) to wake up to catch the next uplink packet from device i. The relay-device will actually wake up at $T - T_{guard}$ in order to compensate for clock drift. T_{guard} must be kept small to reduce energy consumption. Once awake, the relay-device enters in receive mode waiting for the next uplink packet until time $T + T_{guard}$. When receiving the uplink packet it simply forwards the packet to the gateway. If it receives a downlink message in the receive window, it stores the message until the next upstream transmission from the corresponding end-device as explained previously for the observation phase. Note that upon reception of the uplink packet from device i the relay-device updates the wakeup time of device i accordingly to take into account any clock drift.

Algorithm 2. Computing sleep period

Input:
Bound_Devices: *an array of bound devices to relay device, sorted in receiving packet order.*

Output:
sleep_period

```
1: min_time ← Bound_Devices[0].timestamp + Bound_Devices[0].reception_interval
2: for i = 1 to Bound_Devices.size() do
3:     tmp_time ← Bound_Devices[i].timestamp + Bound_Devices[i].reception_interval
4:     min_time ← min(min_time, tmp_time)
5: end for
6: sleep_period ← min_time − current_time
7: return sleep_period
```

4 Performance Evaluation

We performed field tests to assess the performance of the proposed 2-hop approach for increasing network reliability. The university campus with many vegetation and sparse buildings has been our rural environment of deployment. We deployed at first a network consisting of 3 soil humidity end-devices (ED_1, ED_2, ED_3) and one gateway (GW), as shown in the part (a) of Fig. 2. GW was placed in the car park of the Faculty of Science and Technology, two meters above the ground. End-devices have different transmission intervals: ED_1 sends a data packet every 3 min, ED_2 every 5 min, ED_3 every 7 min. LoRa parameters of the experiments were chosen as follows: spreading factor of 12, bandwidth of 125 kHz and coding rate of 4/5, which is the usual setting that provides the longest range. The transmission power for all tests has been set to 14dBm and all measurements were done in NLOS conditions. Two relevant metrics have been identified: packet error rate and power consumption.

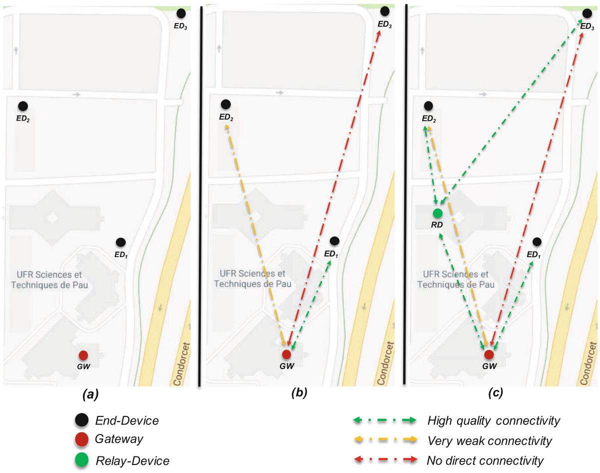


Fig. 2. Deployment scenarios

4.1 Network Reliability

In our first set of experiments, we adopted the standard LoRa one-hop communication scheme. In order to determine the network reliability, we simply measured the number of correctly received packets by the gateway GW. Results, as shown in Fig. 3 (left part), indicate connectivity state of end-devices compared to GW: high quality for ED_1, very weak for ED_2, no direct for ED_3. Part (b) of Fig. 2 illustrates this network connectivity state. In order to assess our 2-hop approach, we introduced a relay-device (RD) in the network so as to obtain stable connectivity between the isolated nodes (ED_2, and ED_3) and RD, but also between RD and the gateway. The part (c) of Fig. 2 shows this deployment scenario. We first tested the network reliability by adopting our 2-hop approach with the relay-device in continuously listening mode waiting for uplink packets. As expected, results validated that adding an extra hop between isolated end-devices and gateway can significantly increase the link reliability in high packet error rate conditions. Indeed, all packets sent by end-devices have been correctly received on the gateway side. That means packets from ED_2 and ED_3 were catched by RD and forwarded to the gateway. Finally, we conducted tests to assess the relay-device's ability to automatically synchronize with the rest of the network by fully embracing our smart and transparent 2-hop approach.

As the relay-device should wakeup in advance (by T_{guard}) to safely catch the next uplink packet, the value for T_{guard} is critical: a too small value may make the relay-device miss the uplink packet while a too large value would consume more energy. By varying T_{guard} from 0 s to 5 s we measured the ratio of correctly received packets at the gateway, i.e. uplink packets catched and forwarded by RD to GW. The observation phase is set to 15 min and at least two packets per end-devices are expected to be catched. We ran each test during 1 h: the first 15mn for the observation phase and the remaining time for the data forwarding phase. Results are shown in Fig. 3 (right part).

Of course, during the observation phase all packets sent by the end-devices are correctly received and forwarded to the gateway. As shown in Fig. 3 (right part), there is a total desynchronization of the relay-device with the rest of the network when $T_{guard} \leq 2$ s. This is mainly due to the fact that the wakeup of the relay-device takes some time. When $T_{guard} \in [3, 4]$, synchronization is partial: at least 50% of packets can be correctly received but not more than 70%. This is due to a small clock drift. When $T_{guard} = 5$ s, it is possible to obtain 100% of correctly received packets.

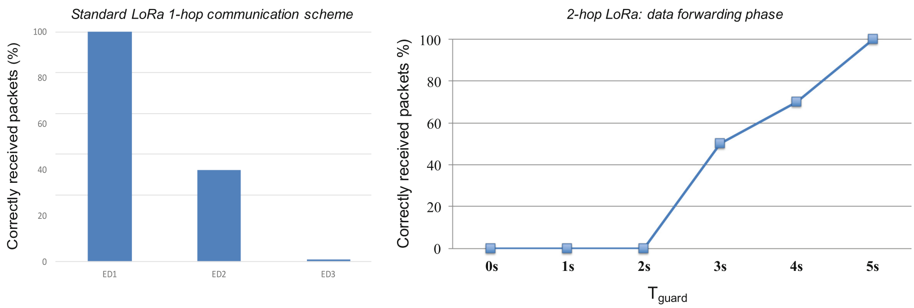


Fig. 3. Correctly received packets

4.2 Discussion on Radio Duty-Cycle

In Europe, electromagnetic transmissions in the unlicensed EU 863-870 MHz Industrial-Scientific-Medical (ISM) band used by Semtech's LoRa technology falls into the Short Range Devices (SRD) category. The ETSI EN300-220-1 document [12] specifies for Europe various requirements for SRD devices, especially those on radio activity. Basically, a transmitter is constrained to 1% duty-cycle (i.e. 36 s/h) in the general case. This duty cycle limit applies to the total transmission time, even if the transmitter can change to another channel. Obviously, a relay-device that has to forward uplink packets from n end-devices will have to transmit at least n packets/hour. Assuming that each transmission takes about 1.5 s (approximatively the time-on-air of a 20-byte payload packet – header included) then a relay-device can relay 24 packets/hour which is quite sufficient in most of the cases.

4.3 Discussion on Energy Consumption

The Arduino Pro Mini (in its 3.3 V and 8 MHz) with the LoRa module draws about 40 mA when active (taking a measure) and transmitting. The whole process takes about 2 s. In deep sleep mode, the board draws 5uA. Therefore an end-device that sends 1 measure every hour consumes in the average $(2 * 40\,\text{mA} + 3598 * 0.005\,\text{mA})/3600 = 0.0272\,\text{mA}$. We have real devices running on AA batteries that have been functioning for more than 2 years at time of writing.

Observation Stage Consumption. In the observation phase, a relay-device must remain in continuous receive mode for a specified duration D_{obs}. The Arduino Pro Mini running at 3.3 V consumes about 15 mA in receive mode. Then, it has to forward the packet which has an energy consumption similar to the transmission from an end-device, i.e. 40 mA during 2 s. At the relay-device level, managing 3 isolated end-devices by relaying for example 3 packets for a duration $D_{obs} = 60$ min, consumes in average $((3 * 2\,\text{s}) * 40\,\text{mA} + (3600\,\text{s} - 3 * 2\,\text{s}) * 15\,\text{mA})/3600\,\text{s} = 15.04\,\text{mA}$. The left part of Fig. 4 shows the average consumption of a relay-device that relays n packets for a duration of 1 h, 2 h and 3 h. Results shows that 1 h of observation has little impact on the battery lifetime even by relaying the maximum number of packets per hour regarding radio duty-cycle ($n = 24$): 15.33 mA, 6.8 days over more than one year of operation.

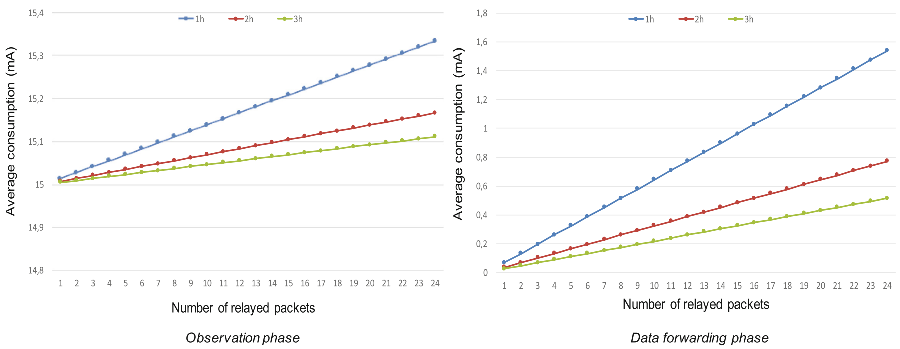


Fig. 4. Average consumption

Data Forwarding Stage Consumption. Regarding the relay-device, it has to wakeup and forward uplink packets. For each wakeup, there will be a continuous receive during $2 * T_{guard} = 10s$ at the maximum, then it has to forward the packet during 2 s. Therefore, for each uplink packet, the relay-device consumes in the average $(10\,\text{s} * 15\,\text{mA} + 2\,\text{s} * 40\,\text{mA})/12\,\text{s} = 19.16\,\text{mA}$. If we assume that a relay-device is used to relay a very small number of isolated end-devices, e.g. 3 end-devices, then the number of wakeup can be limited. For instance, with 3 end-devices, the relay-device has to wakeup 3 times per hour resulting in an average consumption of $(3 * 12\,\text{s} * 19.16\,\text{mA} + (3600\,\text{s} - 3 * 12\,\text{s}) * 0.005\,\text{mA})/3600\,\text{s} = 0.196\,\text{mA}$ which still allows for more than a year of operation. As illustrated in the right part of Fig. 4, results are even better when the relay-device has to wakeup 3 times every 2 h (more than 2 years of operation) or every 3 h (more than 4 years of operation). While maintaining at least one year of operation, a relay-device can relay 4 packets if it has to wakeup every hour, 8 packets every 2 h and 13 packets every 3 h.

5 Conclusion

We described in this article a 2-hop LoRa approach to increase reliability in real-world deployment scenarios. We proposed a smart, transparent and low-power relay-device that can be added seamlessly into an existing LoRa network, between some end-devices and the gateway. Both end-devices and gateway are unchanged and can work with or without the relay-device. The experimental tests demonstrate the effectiveness of our approach, especially validating the relay-device's ability to synchronize in an automatic and asymmetric way with the rest of the network. Using low-cost hardware for the relay-device, the experimental tests also show that a safety wakeup of 5 s prior to the expected time of receiving an uplink packet is sufficient to significantly increase the network reliability.

Acknowledgments. This work was supported by WAZIUP project which was funded by EU Horizon 2020 research and innovation program under grant agreement No 687607.

References

1. Semtech: LoRa modulation basics. Rev.2 (2015)
2. Pham, C., Rahim, A., Cousin, P.: Low-cost, long-range open IoT for smarter rural African Villages. In: Proceedings of ISC2, Italy (2016)
3. Pham, C., Dupont, C.: IoT, an affordable technology to empower African addressing needs in Africa. In: Proceedings of the IST-Africa, Namibia (2017)
4. Pham, C., Rahim, A., Cousin, P.: WAZIUP: a low-cost infrastructure for deploying IoT in developing countries. In: Proceedings of AFRICOMM, Ouagadougou (2016)
5. Adelantado, F., Vilajosana, X., Tuset, P., Martinez, B., Melia-Segui, J.: Understanding the limits of LoRaWAN. IEEE Commun. Mag. **55**, 34–40 (2017)

6. Sanfratello, A.: Enabling relay-based communication in LoRa networks for the Internet of Things. Master thesis. University of Pisa, Italy (2016)
7. Bor, M., Vidler, J.-E., Roedig, U.: LoRa for the internet of things. In: Proceedings of EWSN 2016, Graz, Austria, 361–366 (2016)
8. De Velde, B.-V.: Multi-hop lorawan: including a forwarding node (2017)
9. Liao, C.-H., Zhu, G., Kuwabara, D., Suzuki, M.: Multi-Hop LoRa networks enabled by concurrent transmission. IEEE Access **5**, 21430–21446 (2017)
10. Dias, J.: LoRaWAN multi-hop uplink extension. Comput. Sci. **130**, 424–431 (2018)
11. Lundell, D.: Ad-hoc network possibilities inside LoRaWAN. Thesis. Lund U (2017)
12. ETSI: Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1 000 MHz frequency range with power levels ranging up to 500 mW; Part 1: Technical characteristics and test methods (2012)