LoRaWAN Multi-hop Uplink Extension

José Manuel da Custódia Dias

***Abstract*— The coverage of a LoRaWAN network in a city is greatly hampered by the harsh propagation environment. Sensors are sometimes placed under the ground or in places with strong electromagnetic attenuation. Also, for users who have a contract with a network operator, installing another gateway to improve coverage of blind spots is not an option. In other cases, there is no or very weak connectivity (e.g. basements). In the present work, we design and implement a multi-hop uplink solution integrated with the 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 Destination-Sequenced Distance Vector (DSDV) routing. The routing protocol was successfully implemented and was assessed using a linear and bottleneck topology, where the Packet Reception Rate (PRR) and throughput were measured. On the linear topology we observed that the throughput and PRR did not decrease considerably with the increase of hops. We also addressed the use of Carrier Activity Detection (CAD) to detect and avoid collisions by placing two transmitters transmitting randomly within different intervals with different collision probabilities. It was found that the CAD greatly improved the PRR for intervals in which collisions are certain or highly likely and hence, it can be successfully used to avoid collisions when both transmitters are at range from each other.**

***Index Terms*—LoRaWAN, multi-hop, routing, collisions, uplink, topology**

I. Introduction

In the Internet of Things (IoT), devices collect data, transmit it usually using radio links to base stations, which relay it to an Application Server in the Internet. Data is stored in the Cloud and processed using machine learning and data mining techniques to retrieve useful information. These devices collect their data through one or more connected sensors and are typically powered by a battery. There are several enabling technologies for IoT, which differ in power consumption, range, throughput, Quality of Service (QoS), scalability and cost. LoRa is one such technology operating in license-free sub-Ghz spectrum. The sub-Ghz spectrum has good propagation properties and longer range due to longer wavelength carriers than IEEE 802.11 or Bluetooth, but it is limited in European Union to a duty cycle of 1% in 868-868.6 MHz and 0.1% in 868.7-869.2 MHz [1]. As all devices are under the same restrictions, gateways need to be carefully designed to make an efficient use of the spectrum.

Gateways are complex and costly to maintain. They need to simultaneously listen in all channels and are permanently connected to the Internet. Its density can be reduced through the use of simple relay nodes in a multi-hop scheme. Cities may have dark spots, with weak or no coverage at all, in these scenarios a multi-hop solution is a simple, effective and cheap solution to improve coverage. Also, when a network user does not own the network infrastructure, deploying additional gateways is not an option. Moreover, multi-hop allows data to be collected from remote locations, such as villages or rural areas with no accessible gateways and where it is not economically viable to deploy and maintain a gateway. In addition, if a gateway fails, a multi-hop protocol could mitigate the network failure, as the messages would be forwarded by relay nodes to the nearest available gateway.

The main objective of the present work is to design, implement, test and assess a routing protocol between LoRa transceivers. A prototype shall be developed in order to evaluate the proposed mechanisms under relevant scenarios. This document is organized as follows: Section II provides a description of Low Power Wide Area Network (LPWAN) technologies, LoRa, LoRaWAN and ad hoc routing protocols. Section III describes an analysis to the most recent LoRa literature in various fields: scalability studies, collisions, empirical measurements and routing. Section IV describes the implemented communication protocol and mechanisms, and the prototype implementation. Sections V and VI contain, respectively, a set of empirical measurements and an assessment of the implemented solution.

II. Background

A. Low Power Wide Area Network

A LPWAN is characterized by long range, low power consumption, low cost and low throughput. End-nodes can be as far as dozens of kilometers, communicate up to dozens of kbits per second and with a battery lasting up to 10 years. In a typical architecture, end-nodes are usually battery powered devices with Radio Frequency (RF) modules which are placed in moving or static objects. Gateways are deployed kilometers apart depending on range, node density and physical propagation environment. End-nodes communicate using their RF modules to a gateway which relays the information to an application server.

1) LoRa Physical Layer: LoRa is a long-range wireless communication technology intended for battery powered applications where the battery lifetime and energy consumption is important. The physical layer implements a derivative of Chirp Spread Spectrum (CSS) scheme. In CSS, each data bit is modulated with a single or a sequences of chirps, which are signals whose frequency increases (up-chirp) or decreases (down-chirp) over time. Transmissions are distinguished by the time duration of each chirp reflected on the Spreading Factor (SF), i.e. higher SFs correspond to longer chirps and lower SFs to shorter chirps. Higher SFs increase resistance to natural interference, noise and jamming but decreases available bandwidth for data. According to [2], the definition of the SF for LoRa is $SF = \log_2(\frac{Chirp\ rate}{Symbol\ rate})$. Every symbol is encoded in 2^{SF} chirps that cover the available bandwidth (BW) (one

chirp per second per Hz). To improve robustness to errors, LoRa adds redundancy by employing a Cyclic Redundancy Check (CRC) to perform forward error detection and correction. The measure of redundancy is the Coding Rate (CR) defined as the portion of the bit stream which corresponds to effective data, i.e. non-redundant. The structure of a LoRa packet, which can be explicit or implicit, is showed on figure 1. It comprises a configurable preamble for synchronization of the receiver, which is a sequence of symbols (chirps) ranging from 6 to 65535, an optional header that can provide information about the payload length, the Forward Error Check (FEC) CR and if CRC is present. Implicit mode exists to reduce overhead by eliminating the header field. This is useful in situations where both sides have prior information about the CR and the CRC or when these parameters are fixed and manually configured at both sides.

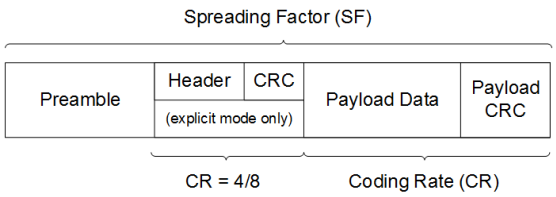

Fig. 1. Structure of a LoRaTM packet [3].

2) LoRaWAN: LoRaWAN is a network protocol specified from the data link layer above in the OSI[1] model, comprising the network, transport, session, presentation and application layer. Typically, a LoRaWAN network consists of a single hop star topology, in which gateways in the center relay messages between the end-devices and a Network Server. An end-device can reach one or more gateways using LoRa or Frequency Shift Keying (FSK) modulation. Gateways are connected to the Internet via Wifi, 3G or Ethernet and a message is delivered to the Network Server through IP [4]. Different end-devices and gateways can coexist in the same geographic area by isolating communications in frequency and also in virtual channels within the same frequency channel, as, due to spread spectrum, transmissions with different spreading factors are orthogonal. Each gateway is constantly listening on simultaneous channels and can successfully receive multiple packets as long as they are transmitted in different channels. To address applications with different needs of power consumption, LoRaWAN specifies three classes of end-devices based on their listening time, all of which with bidirectional communication [4]:

- Class A - An end-device only listens for Downlink (DL) traffic during two scheduled time intervals after an Uplink (UL) transmission takes place.
- Class B - An end-device will listen at scheduled times. A beacon is periodically sent by the gateway for synchronization.
- Class C - An end-device listens whenever it is not transmitting.

[1]Open Systems Interconnection

B. Adaptive Data Rate Scheme (ADR)

The LoRa network infrastructure may have control over end-nodes's RF power and spreading factor through an Adaptive Data Rate Scheme (ADR), which directly affects the data rate. Hence, when referring to a change in data rate, the spreading factor is implicitly changed as well. The ADR tries to assign the lowest possible spreading factor to each node, resulting in the lowest time on air of each transmission and thus, fastest data rate available. The node must verify that the gateway is receiving uplink packets. If not, the spreading factor in use is not appropriate for the quality of the radio link and thus needs to be increased to allow a longer range.

C. Routing

1) Routing Protocol for Low-power and Lossy Networks (RPL): Routing Protocol for Low-power and Lossy Networks (RPL) is a proactive distance vector routing protocol specified by the Internet Engineering Task Force (IETF) for Low-power and Lossy Network (LLN). A LLN is formed by energy, processing power and memory constrained nodes, in which the links connecting them are bidirectional, lossy, low rate and possibly unstable. Although RPL was designed for IPv6 over Low Power Wireless Person Area Networks (6LoWPAN), it can be adapted to a LoRa network given that the traffic is mainly oriented from the end-nodes towards the gateway.

2) Destination-Sequenced Distance Vector (DSDV): The DSDV routing protocol was proposed by C. Perkins and P.Bhagwat in [5]. In DSDV, each node in the network stores a routing table which contains the destinations reachable by that node, the metric associated with a certain path, the next node in that path and other fields. DSDV considers two types of routing advertisement packets based on the information they carry:

- Full dump - Contains all the entries in the routing table.
- Incremental - Only contains information that changed since the last Full dump packet.

Full dumps are to be sent sporadically. When changes are more frequent and there are too many incremental packets being sent, full dumps shall be scheduled more frequently. One important property of DSDV is that its routes are loop-free, which proof can be found in [5].

III. LoRa Related Work

A. Collisions

LoRa modulation is referred to have non destructive concurrent transmissions when using the same physical (frequency) and virtual (SF) channel. To address this topic, [6] set up an experiment with two LoRa transceivers configured with a difference of power of 1 dBm to differentiate a strong from a weak transmitter. Other parameters were the following: $SF = 12$, $BW = 125\ kHz$, $CR = 4/5$, payload was set to 10 bytes and the preamble to 12.25 symbols, which resulted in an airtime of 991 ms.

The experiment consisted on varying the offset time of the strong transmitter relative to the weak transmitter. The results are described below for a packet with 60.25 symbols:

- Both strong and weak transmissions were received successfully when the strong transmission takes place more than 57 symbols before than the weak transmission. In this case, the three last symbols of the strong transmission overlap the 3 first symbols of the preamble of the weak transmission.
- The strong transmission is received if it initiates less than 57 symbols earlier but not more than 3 symbols after the weak transmission. Within those intervals the weak transmission is not decoded.
- If the strong node initiates the transmission 3 symbols after the weak node, neither of the two is received up to 57 symbols. After that value, the strongest transmission is received but the weak transmission continues not to be received. This case corresponds to the overlapping of the strong transmission with the CRC of the weak transmission in the end of the packet. The weak transmission only starts to be received again when the strong transmission initiates approximately when the weak transmission has ended, which corresponds to more than 60 symbols given the packet length.

The experiment was repeated with both transmitters configured with the same power and in this case, either one is perceived as the strongest with similar results. The authors conclude that one of two concurrent transmissions is received with very high probability if the offset between the two transmissions is not greater than 3 symbols.

B. LoRaBlink

LoRaBlink is a multi-hop protocol for LoRa™transceivers proposed in [6] to address multi-hop under low density and low traffic volume of the network. It combines Medium Access Control (MAC) and routing, using beacons for time synchronization and in order to communicate the distance in number of hops to the gateway or sink. The first beacon is transmitted by the sink and then there is a flooding process in which each node propagates to its neighbors the number of hops it is away from the sink. As a time synchronization mechanism, a beacon indicates the start of an epoch. An epoch is constituted by N slots, in which the first N_B are beacon slots used to transmit hop distance beacons and N_D are data slots. N_B establishes the maximum number of hops. For instance, if a node received a beacon with a hop count of 4, it received it in the 4th slot and will convey (if this is the lowest value to date) its own hop count of 5 in the 5th slot. For the implementation, each node was configured with $SF = 12$ and $BW = 125$ kHz and to transmit a 10 byte packet. Such configuration implies a very low data rate, with packets being spaced in time, at least, 1 minute and 31 seconds based on an airtime of 925 ms and according to the duty cycle limitations. The authors decided for a longer period of, on average, 5 minutes. When a message with data is sent, all the neighbor nodes check its origin in terms of hop distance and if they are closer to the sink, they relay the message. Several transmissions may arrive at the gateway through different relaying paths, which introduces redundancy and therefore unnecessary power consumption.

C. Empirical Measurements

In [7], two sets of empirical measurements were conducted in the coastal city of Oulu, Finland, which is mainly flat and in which the highest buildings are 12 floors high. One of the sets was performed on water in a moving boat and the other on land in a moving car at the maximum spreading factor of 12 and a power of 14 dBm. On land, more than 80% of the packets were successfully delivered for distances less than 5 km and 60% for distances ranging between 5 to 10 km. On water, the success rate was approximately 70% for distances below 15 km.

In [8], the authors conducted two experiments within a building to study the packet loss and signal attenuation in the presence of reinforced concrete. The results showed that, even with the highest spreading factor and power being used, a LoRa gateway was not able to cover an entire reinforced concrete building, with the packet loss decreasing rapidly with each entrance.

D. Scalability

In [9], the authors conducted simulations using different combinations of frequency channels and spreading factors to see how the throughput and the Packet Error Rate (PER) vary with an increase in number of nodes with a single gateway. When communications were divided in both frequency and SF, $PER = 32\%$ for 1000 nodes. The simulation of a pure Aloha scheme (assuming colliding transmissions to be lost), resulted in $PER = 90\%$ for 1000 nodes, much worse than when considering collisions as non destructive under certain limits. The throughput was 6 times lower in the pure Aloha scheme.

IV. LoRaWAN Multi-hop Uplink Extension

The scope of the present work is to design, implement and test a routing protocol which allows typical LoRaWAN end nodes, named Leaf Nodes (LN), which are not in range of a gateway, to send their data to intermediate nodes located between them and the gateway, named Routing Nodes (RN). Each RN runs an instance of the routing protocol, exchanging routing information packets with neighbor RNs to find routes in order to forward packets from LNs to gateways. We assume RNs are not energy constrained - an energy analysis to assess its effectiveness outside this assumption is left for future work. The implemented protocol will have its performance evaluated in Section VII.

A. Routing Protocol

1) Overview: The protocol chosen to be implemented was a simplified version of DSDV. The differences stem from the need to optimize a protocol originally designed for mobile nodes to a LoRaWAN network. Firstly, routing nodes are not intended to be mobile, but static and extend the radio coverage of the network in specific spots. For this reason, there is no pointer for information regarding the stability of a certain route as in the original DSDV. As fluctuations are not expected in a LoRaWAN network, routes are considered stable when they

are advertised. Lastly, there is synchronization between nodes, contrary to the DSDV, which is set by a gateway. Gateway neighbors transmit their full dumps upon a beacon reception and not when a ticker expires. Other RNs will transmit their full dump upon reception of the first full dump from a neighbor within each beacon period as represented in the first and third sequence of arrows in Figure 2. Other full dumps may be received after a node transmits its own and, in this case, the information received after will be stored normally and sent in the full dump of the next beacon period.

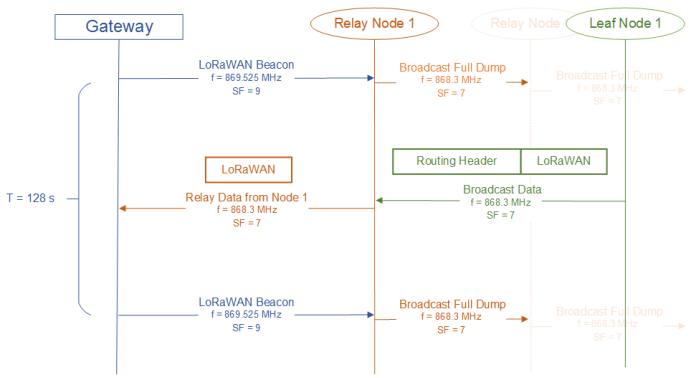

Fig. 2. Timing diagram of the implemented multi-hop solution. Different colors represent different types of nodes: gateway (blue), RN (orange) and LN (green).

LNs construct a LoRaWAN packet and append a routing overhead to forward the packet within the multi-hop extension. When the packet is received by a gateway neighbor, the overhead is dropped and only the LoRaWAN part is sent. A representation is given in the second sequence of arrows with origin in LN1 in Figure 2. The packet will appear in the Application Server as being sent by the node that sent it and its encryption is end-to-end as a LoRaWAN single hop network. The three colors in Figure 2 represent the three different components in the network. A gateway, in blue, sends its LoRaWAN beacon periodically, most commonly, each 128 seconds. RN1, in dark orange, is close to the gateway and will receive the beacon, update its routing table and transmit a full dump as soon as possible. RNs that did not receive the beacon, in light orange, receive the full dump from a neighbor RN (as RN1 for instance), update their routing table and transmit their full dump. LN1, in green, transmits data periodically. The data packet comprises the LoRaWAN part and the header until it reaches RN1, which contains a gateway in its routing table. RN1 removes the header and forwards only the LoRaWAN part so that the packet is correctly interpreted by the gateway.

Each RN stores a routing table containing a list of destinations, the metric to reach each destination, the next hop in the path, the sequence number issued by the destination and a time stamp to detect stale entries. Since there is only uplink traffic in the current implementation, a destination corresponds to a gateway, and only the gateway with the less number of hops (metric) is stored. The protocol is, however, prepared to support other destinations such as RNs or LNs, which allows downlink traffic in future work to be delivered from a gateway to a RN or LN along the correct path. When full dumps are received, the information is compared with the stored routing table, which is updated when there are new destinations, better metrics for the same sequence number or newer sequence numbers. RNs can be divided in those that are gateway neighbors and those that are not. Gateway neighbors schedule a short reception window right after receiving a beacon in which the channel parameters are changed to those used by the gateway to transmit the beacon. Other RNs also schedule the next instant from which they expected the first full dump of the next beacon period after which a RN should send its own full dump.

2) Packet Strucure: The packets exchanged within the multi-hop extension may be of two types: full dumps containing routing information or data packets containing application data. Nodes exchange full dumps in order to form routes, with each node storing only the next node along the path and its metric to a given destination as in a typical distance vector protocol. Sequence numbers issued by each destination are also included as they ensure loop freeness. The full dump packet structure is represented in Figure 3. The sender is

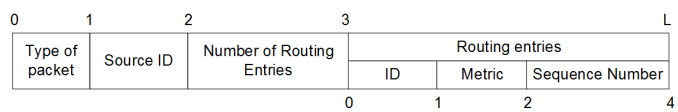

Fig. 3. Packet structure of a full dump containing routing information with field lengths in bytes.

identified in the Source ID field, which can only contain an ID from one of the receiving node's neighbor. When a destination is added to the routing table of the receiving node, the Source ID field is added as the next hop to reach that destination. The number of entries is included for the receiving node to know how to parse the entries. Each entry contains the ID of a destination, its metric and the sequence number generated by that destination. The list of routing entries represents a short version of the routing table of the sender node. The maximum length of a LoRa packet is 256 bytes which gives a maximum number of $\frac{256-3}{4} \approx 63$ destinations in a single packet. In the presence of more than 63 destinations, fragmentation is needed.

Fig. 4. Structure of a application data packet within the multi-hop network with field lengths in bytes.

A LN sends a data packet with the structure of Figure 4. The Unicast ID field identifies the next hop node, i.e. to whom the packet shall be forward to. This ensures packets will only travel to one path towards their destination when reaching a RN. When a LN transmits a packet, this field is assigned a broadcast value. The Destination ID generally corresponds to the ID of the gateway, i.e. 0. The remaining part, LoRaWAN, contains the data payload and the fields specified in Section II. Activation by Personalization (ABP) must be used since the current implementation does not support downlink.

B. Components

1) Parsing of received packet: When a packet is received, it is verified to be one of the following: a full dump, a data packet or a beacon. The different cases are explained below individually.

- Beacon from a gateway. The parameters contained in a full dump, namely the destination, sequence number and metric are extracted from a LoRaWAN beacon packet.
- Full dump. The sender's ID and and type of packet are extracted from the expected fields according to Figure 3. The type is now checked to contain an 'R'. Next, the packet is iterated to extract the several advertised destinations.
- Application data packet. If the packet contains application data, the type field should contain a 'M' character. The two IDs are extracted and, firstly, the Unicast ID is compared with the ID of the RN that received it. If there is a match, the RN looks for the next node in the path. If the next hop is a gateway, the three bytes of overhead are eliminated, and if not, the Unicast ID field is updated to the next hop.

2) Full Dump: Full dumps are transmitted right after receiving a beacon or the first full dump within a beacon period. The routing table is iterated and all entries are put in a full dump packet to be broadcasted to neighbors. While iterating, the program first checks if the current entry is stale by comparing its Install Time with the current value of the timer. If the difference is greater than a certain number of beacon periods, the entry is considered stale. If the entry is stale, the sequence number is incremented to indicate an unreachable destination, the row is added to the current full dump packet and then removed from the table. If the sequence number is even, that destination was received as unreachable from a full dump packet in the previous beacon period. As such, it was not transmitted yet and was maintained in the routing table to be transmitted in the next beacon period. A destination in this condition is added to the packet and removed from the routing table.

3) Packet Transmission: Every transmission starts with a CAD analysis on the medium to detect a valid LoRa preamble. If a preamble is not detected, it assumes the medium is free and transmits. On the other hand, if a preamble is detected, the transmitter will try to avoid a collision. The transmitter then follows an exponential backoff, i.e. wait a random amount of time until checking the medium again and doubling the upper bound of the waiting time when preambles are consecutively detected.

C. Design Choices

This section provides a description of the choices taken when developing certain features of the protocol together with the respective justifications and alternative options.

1) Carrier Activity Detection: According to the datasheet of the transceiver, the current consumption in CAD mode, approximately 8.2 mA, is slightly lower than receiving mode, 10.3 mA. The difference comes from the fact that the CAD mode is divided in two phases: receiver and processing [3]. Thus, one idea could be to use the CAD to continuously check the medium until a valid preamble is found instead of being in continuous receive mode. However, when the CAD finishes, an interrupt is fired and the CAD does not repeat even if no CAD was detected. As each CAD action needs to be sent individually, one after the other, this would imply that the MCU could not sleep by having to call the CAD requests.

2) Frequency Channels: RNs transmit in one of the frequency channels between 868.0 and 868.6 MHz (allowed to a maximum of 1% duty cycle). The frequency in which the nodes exchange the full and incremental dumps and the application data messages is constant and should be defined when routing nodes are flashed. A RN which is neighbor of a gateway and encapsulates the packets in LoRaWAN, transmits to the gateway in a random channel in the set 868.1, 868.3, 868.5 MHz to comply with the LoRaWAN specification.

3) Downlink: Downlink is not supported in the presented work. LNs do not advertise themselves to RNs. Routing nodes can only know about a LN's existence by storing the Node Address field contained in a LoRaWAN application data packet when received with a broadcast identification in the Unicast field, which means that RN is the first hop along the path. Another approach would be to store the node IDs of RNs that forward the application data packet inside the packet itself. When the packet reached a neighbor of a gateway, the path would be extracted and stored in a cache. When receiving a downlink packet, the RN would forward it along the inverse path. If we consider that the RNs are subject to duty cycle constraints, it becomes evident that sharing resources between uplink and downlink turns it difficult to guarantee downlink packet delivery to LNs unless they are put in continuous reception. Otherwise, even if there was some sort of synchronization mechanism between a RN and a LN, it would be hard to guarantee packet delivery in specific time instants.

4) Beacon Timing: We initially thought of implementing an exponential timer between beacons similar to RPL due to the static profile of a LoRaWAN network. The advantage would be the reduction of overhead and consequently higher data throughput with the consequence of faulty nodes being detected and advertised with a higher delay. The drawback was that the gateway's packet forwarder had to have its beacon timer modified from a constant to a dynamic period, which does not correspond to the LoRaWAN specification and counters the notion of extension. Another solution for the beacon was to make it a downlink packet sent from the application server as an alternative to avoid modifying the gateway. However, there are two main drawbacks related with such option. Firstly, independent multi-hop networks would need separate downlink messages, which negatively affects the available downlink throughput to single hop nodes. Furthermore, there is no guarantee of message delivery within a certain time. As such, a RN at range of more than one gateway could receive the same sequence number at different instants, which could originate fluctuations on routes and unnecessary advertisements.

D. Prototype Description

The routing protocol described previously was implemented and tested in a prototype system. This section covers the choice of hardware and software to integrate the protocol in the several components of an LPWAN: the nodes, the gateway, the Network Server and the Application Server.

1) Nodes: The selected LoRa™radio module was the HopeRF RFM96 868/915Mhz RF transceiver module, which communicates with de Microcontroller (MCU) through Serial Peripheral Interface (SPI). There are several parameters, which can be configured by modifying the module's registers such as the SF, CR and the Bandwidth (BW). Further information can be found in [10]. The STM32L432KC board, which includes an embedded ARM 32-bit Cortex-M4 CPU, was selected. The selection was based on its cheap cost, ultra low power operating modes, user-friendly Software Development Kit (SDK) and Application Programming Interface (API)s, compatibility with the shields for Arduino Nano and extent of open source code and libraries. To increase diversity, LNs are composed by the LoRa module and an Nucleo Board or an Arduino Pro Mini, depending on the testbed. There are two nodes with the Arduino Pro Mini and four nodes with the Nucleo Board.

2) Gateway: The LoRa concentrator is the iC880A-SPI board manufactured by IMST, which supports up to 8 frequency channels at all spreading factors. This board has an integrated Semtech SX1301 as a demodulator and two SX1257 transceivers. The concentrator was assembled to a Raspberry Pi 3 with the help of an interface board and a GPS was also added to obtain a time reference. Ultimately, the antenna used was $\frac{1}{2}\lambda$ and 2 dBi gain. The Raspberry was running the Raspbian Jessie Lite operating system. A detailed guide[2] from the TTN Zurich was used to configure the packet forwarder.

3) Network Server: The Things Network (TTN) is a crowd-sourced and opensource LoRaWAN network with an incorporated Network Server. To communicate from an end node to the Network Server, the device must be registered in the backend. An application was created in the TTN console, and a device was registered in it with a given EUI and an application Key was generated for that particular device. Any gateway registered in TTN can receive messages from the device and forward them to the application. However, a gateway was also registered in the TTN console with its unique identifier, the EUI. When the packet forwarder starts to run in the gateway it is possible to confirm the status of the gateway as active in the backend.

4) Application Server: TTN dashboard does not persist data and only displays incoming packets in real time. Past events are eliminated and thus, there is a need of using one of the available types of integration to store data from tests for later analysis. The software programming tool Node-Red was chosen for such task, which allows the creation of an MQTT broker that subscribes to the specified Application EUI and one or more Device Addresses. Different functions, according to the test, were then written in JavaScript to parse the JSON packets and put the extracted data in a CSV file.

[2]https://github.com/ttn-zh/ic880a-gateway/wiki

V. Empirical LoRa coverage measurements

The city of Almada is constituted by trees, buildings with different heights and a considerable amount of changes in elevation ranging between 5 and 115 m. The gateway was positioned on the 7th and last floor of a building, located in a place with an elevation between 50 and 70 m, one of the highest points in the city. A mobile node constituted by an Arduino Pro Mini and a RFM95 LoRa transceiver configured with a transmit power of 14 dBm, $BW = 125$ kHz, $SF = 7$ and $CR = 4/5$, was carried in a car and driven through the city at speeds not exceeding 50 km/h. Figure 5 shows the map of the city where the data was collected. The convergence point

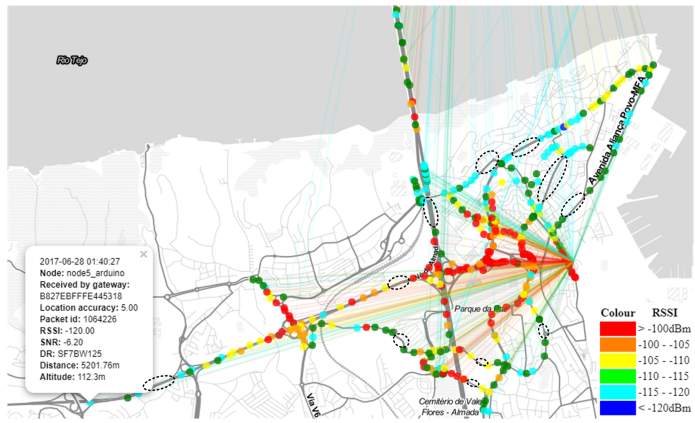

Fig. 5. Map generated by TTN Mapper with the measurement location points.

represents the location of the gateway, the colors indicate the RSSI in the gateway and the black dotted ellipses represent dark spots, where more than one transmission was lost. As the RSSI before these zones did not decrease significantly, it was not possible to predict the the subsequent loss. Lines that start in the points, but do not converge only to a single point in the figure, correspond to transmissions that were received by other public TTN gateways located in Lisbon.

VI. Evaluation of the Multi-hop Routing Protocol

LoRa and other technologies that operate in the unlicensed spectrum have duty cycle regulations that limit the maximum data rate available per node. This upper limit is much lower than the maximum data rate imposed by the modulation and coding employed. In a multi-hop network where nodes with the same restrictions forward packets from other nodes, it is critical to identify what is possible when designing under such restrictions. Equation 1, which complete expression can be found at [3], was used to calculate the time on air throughout the current section.

$$\begin{aligned} ToA &= T_{preamble} + T_{packet} \\ &= T_{symbol}(L_{preamble} + L_{PHDR} + L_{PHDR\_CRC} + \\ &+ L_{PHDR\_CRC} + L_{PHY\_payload} + L_{PHY\_CRC}) \end{aligned} \quad (1)$$

A. CAD

Two experiments were conducted to evaluate if CAD can be used to successfully avoid collisions. The first experiment is used as reference and comprises one transmitter without CAD and one receiver. The second experiment comprises two transmitters with CAD and one receiver. The CAD experiment consisted on having two synchronized transmitters which would transmit randomly within a given interval of time. The time intervals were varied from 8 ms to 128 ms in the set 8, 16, 32, 64, 128 with 2000 transmissions in each. The transmitted packets had a payload of 20 bytes, with explicit header, $CR = 4/5$, $BW = 250$ kHz and a programmed preamble of 8 symbols. The resulting air time is $ToA \approx 28.29\ ms$.

Comparing the packet ToA with the testing intervals one may observe the following:

- For 8 and 16 ms, collisions are certain
- For 32 ms, collisions are highly probable
- For 64 and 128 ms, collisions are less probable

The time between transmission cycles, 1 second, needs to include a minimum number of backoff times in case of consecutive carrier detections. Since the same code runs on both transmitters and they start at the same time, synchronization is expected to be kept. The PRR for the experiments with CAD (triangles) and without CAD (circles) is represented in Figure 6. The colors represent the different transmits intervals. As

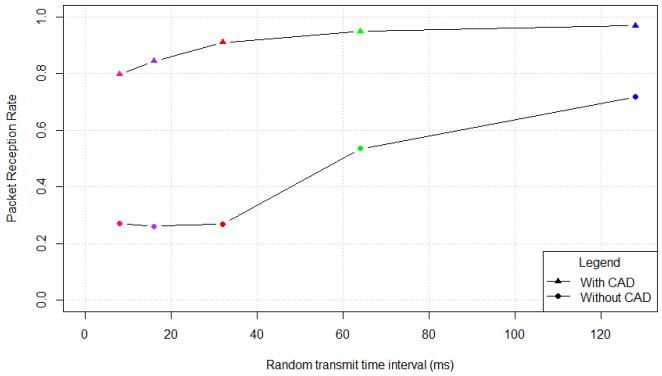

Fig. 6. PRR for the different transmit time intervals tested.

expected, in the experiment without CAD, the PRR decreased with the shortening of the interval, due to an increase in the probability of collisions. Also when shortening the interval, the distance between points for the same intervals, with and without CAD, is higher. The results also show that the PRR stays approximately constant at 27% for intervals where collisions are highly likely and certain. For every interval, being collisions probable or certain, the CAD is observed to greatly increase the number of received packets when compared with the case where no CAD was used. The obtained numeric results are presented in Table I.

In [6], two transmitters with different transmit power were used. In the present experiment, nodes were configured with the same transmission power but were placed at two different distances from the receiver, which leads to a weak transmission

TABLE I: PRR in percentage for situations with and without CAD for different intervals and 2000 transmitted packets.

Time interval (ms)	PRR Without CAD	PRR (With CAD)
8	27.1%	79.6%
16	26.0%	84.3%
32	26.8%	90.1%
64	53.4%	94.8%
128	71.6%	96.8%

(more distanced from the receiver) and a strong transmission (less distanced from the receiver). It was observed that for intervals where collisions are almost certain, for the case without CAD, there is a concentration of RSSI in the higher (less negative, around -37 dBm) values. This indicates that only the strongest transmission is received, which corresponds to the capture effect. When CAD is used, the distribution of RSSI flattens and distributes both around -37 dBm (strong transmission) and -53 dBm (weak transmission) for the 8 ms interval, which is due to the fact that the transmissions begin to alternate. When the weak transmitter is transmitting, the strong transmission will detect it and wait, allowing the weak transmission to be received and demodulated correctly.

B. Beacon Reception Window Accuracy

The time a RN stays in reception mode in the beacon channel, may affect the performance of the multi-hop network, since nodes may lose application packets if the beacon reception interval is too long. To evaluate how shrinkable this interval can be, an experiment was set up comprising 4 independent tests. Each test consisted in configuring a single RN with a different beacon period (64, 128, 256, 512). In each test we measured the time interval between the start of the reception window and the actual reception of the beacon.

The results show that the Nucleo's clock drifts relative to the gateway's clock and that the delay increases with the beacon's period. We computed a linear regression over 4 points given by the mean of time until reception for each beacons' period. The distance between minimum and maximum samples in each case was not greater than 6ms. The resulting expression with the corresponding error of estimate σ_{est} is displayed on Equation 2 plus a reference time t_0 equal to 700 ms. This time corresponds to the instant at which the reception window started, measured from the end of the beacon period, T_{beacon}. For instance, in each of the four experiments, the window started 700 ms before T_{beacon}.

$$t_{RXstart} \approx -0.575 \times T_{beacon} + 684 + t_0\ [ms],\ \sigma_{est} \approx 0.133\ ms \tag{2}$$

$t_{RX\ start}$ represents the start of the next beacon reception window relative to the beacon period. For instance, for a beacon period, T_{beacon}, of 256 seconds, $t_{RXstart\ 256} \approx 536.5 - 700 \approx -163.5\ ms$, which indicates that the next reception window should start at instant $256 + t_{RX\ start} \times 10^3$ seconds. By inspecting the slope of Equation 2, we conclude that the drift increases approximately 0.6 ms per additional second in the beacon period.

VII. Multi-hop Performance Evaluation

The current section presents a study on the performance of the implemented routing protocol. The Packet Reception Rate (PRR) and Throughput were the indicators used to assess such performance. In each of the following experiments, we logged each transmission in each RN and also on the Application Server. In each RN, a counter for the number of received packets from each LN was maintained and, in the Application Server, each received packet, containing the packet counter, node identification, time and other parameters, was stored in a *.csv file with a Node-Red script. This section describes the results obtained for the following scenarios:

- Single hop;
- Two hops with a simple forwarder - no routing protocol;
- Linear topology with routing protocol, from one to five hops;
- Bottleneck topology with routing protocol, two hops.

A total length of 31 bytes was chosen: 15 bytes of application data, 3 bytes of routing overhead and 13 bytes of LoRaWAN overhead (similar size to Sigfox). The rest of the configured parameters were $f = 250$kHz, $CR = 4/5$, a programmable preamble of 8 symbols and the use of explicit header. On all experiments, a total of 500 packets were transmitted.

A. Single Hop - LoRaWAN Benchmark

For this first experiment, we assessed the performance of one node reaching a gateway through single hop such as in a typical LoRaWAN network. The maximum theoretical throughput is limited by the technology, by the duty cycle and by the LoRaWAN overhead. The packet time on air is $ToA_{ref} = 33.41$ ms. If we consider the duty cycle limitation, we get the maximum theoretical throughput of 35.92 bit/s for a 15 byte data packet. The resulting experimental throughput, calculated based on the time between the arrival of the first and the last packet, Δt, and on the total number of packets received N, resulted in Equation 3. The obtained PRR was 99.4% for 500 transmitted packets, which is higher than the value given on the datasheet (99%).

$$Th_{ref} = \frac{L \times 8 \times N}{\Delta t} = \frac{15 \times 8 \times 497}{1685} \approx 35.39\ bit/s. \quad (3)$$

The small difference between the two comes from the loss of three packets, which is not considered in the theoretical equation. This result may be used as a reference for the tests that follow.

B. Two Hops without Routing Protocol

The second experiment consisted on evaluating the throughput and PRR when one node forwards LoRaWAN packets from the LN to the gateway. A diagram representing the present experiment is given in Figure 7. The LN1 (green) will send LoRaWAN packets with no overhead, which will be received by the forwarder (orange) and retransmitted to the gateway. We do not name the forwarder node a RN given that this node does not run the implemented routing protocol. Instead, the forwarder solely transmits what it receives. The resulting throughput, as calculated according to Equation 3,

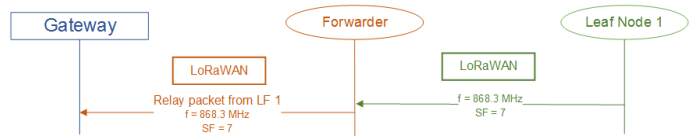

Fig. 7. Diagram representing the experiment with a simple forwarder between the LN and the gateway and no used routing overhead.

and the PRR, both calculated at the Application Server, were, respectively, 34.73 bit/s and 97.6%. These values are slightly worse than the single hop case. The datasheet lists a PRR of 99% for the used radio module, which indicates that more losses are expected with the introduction of one more hop.

C. Linear Topology

In this subsection, we analyze the routing protocol overhead and the impact of increasing the number of hops in a linear topology, from two hops to five hops, as represented in Figure 8. The figure represents the four conducted tests, in which the number of RNs between the gateway and the LN5 was increased by one. The numbers in brackets correspond to the number of the test.

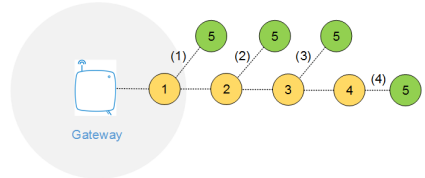

Fig. 8. Linear topology test configuration. In each test the number of RNs was increased by 1.

The first experiment was similar to that of the previous subsection, with the difference that the forwarder node had the routing protocol implemented, constituting a RN. In this case, the forwarder node waits for a beacon and sends its full dump. Also, each packet originated from the LN to the RN has three bytes appended to the LoRaWAN packet. Hence, the throughput and PRR are expected to be decrease.

The subsequent experiments consisted of increasing the number of RNs between the LN and the gateway, one by one. RNs that were in the middle were configured to only fire the reception interrupt if a routing packet came from an adjacent RN neighbor or, in case of a data packet, if the Unicast ID field had their ID. For the RN adjacent to the LN, the previous Unicast ID condition was changed to the Broadcast ID. This way, we are sure that the path taken by a data packet passes through all the nodes.

We first let the topology be constructed by switching the nodes on and waiting for the first beacon to be received. The first beacon will trigger the series of full dumps within the multi-hop topology. The obtained results for the PRR are presented in Table II and the throughput, again calculated from Equation 3, is presented in Table II.

We observe that from the simple forwarder with no protocol to the RN with implemented protocol, the throughput decreases 6.5%. This may have two reasons. The first comes

TABLE II: PRR and throughput for 4 experiments with a linear topology with different number of RNs.

RNs	App. Server	RN1	RN2	RN3	RN4	Throughput (bit/s)
1	98.2%	99.4%	-	-	-	32.48
2	94.8%	95.0%	96.8%	-	-	31.36
3	93.4%	94.2%	96.2%	96.8%	-	30.87
4	91.6%	94.2%	94.4%	96.2%	98.0%	30.26

from the fact that the three bytes of overhead appended to the packet by the LN increase the packet air time, also increasing the denominator of Equation 3 without increasing the numerator, which only takes into consideration the application payload. The second reason may be the packet loss, since the RN is not only receiving data packets from the LN but also switching frequency at a period of 128 seconds to received the beacon. During this interval, the RN cannot receive packets originated by the LN. Moreover, full dumps have priority over data packets and, if the queue is full, a packet must be dropped in order to transmit the full dump every 128 seconds. However, as shown in Table II, the PRR was not lower than the case with the simple forwarder and thus we assume that the difference between the two is inside the confidence interval. Regarding the results with different number of RNs, we found that the PRR at the Application Server decreases on average 2.2% and the throughput 2.3% with each new hop.

D. Bottleneck Topology

The bottleneck topology is constituted by a RN within the range of the gateway and a set of neighbor LNs. This case is particularly important in a duty cycled network, where a RN is under the same restrictions as the other nodes. Consider the experiment's topology depicted in Figure 9. If each of

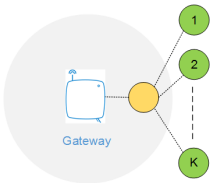

Fig. 9. Bottleneck topology with one RN and K LNs.

the K nodes transmits at the maximum rate of one packet per time interval of $ToA \times 100$, the queue of the RN, in yellow, increases in every such interval, $ToA \times 100$, by $(K-1)$ packets. This results from the fact that the RN receives the K packets, one from each LN, minus the packet that it forwards to the gateway. Unless every LN only transmits a packet every $ToA \times 100 \times K$ seconds, the RN will suffer congestion and will eventually run out of memory and start to discard packets.

We have set up an experiment constituted by three LNs and one RN, in which we maintained the same packet parameters but configured each LN to transmit every $ToA \times 100 \times 3 = 10.79$ seconds, where the packet air time is $ToA = 35.97\ ms$. Also, the transmit power was decreased to the minimum (0 dB). All LNs were configured to be in reception mode when started. Upon reception of the first packet (the full dump from the RN), each LN started a periodic timer at the same time to be synchronized with the other two LNs. Each transmission occurs in a random interval within the first 6.4 seconds of each interval of 10 seconds. The LNs and the RN were positioned at a distance of 5 cm from each other and at the same distance from the RN, which introduces the possibility of collisions. The obtained results for the PRR and throughput are presented in Table III.

TABLE III: PRR and throughput for the experiment with a bottleneck topology composed by three LNs and one RN.

	Application Server			RN1		
	LN1	LN2	LN3	LN1	LN2	LN3
PRR	87.2%	87.6%	88.6%	92.8%	93.6%	94.4%
		88.7%			93.6%	
Throughput		29.56 (bit/s)			-	

The obtained throughput at the Application Server for the ensemble of three nodes is lower for the bottleneck topology than for the linear topology with one RN. The difference may be explained by the difference in the PRR, which is also lower. The difference of 9.5% in the PRR may be explained by concurrent transmissions from the LNs colliding at the RN. Despite the CAD mechanism being implemented in the LNs to avoid collisions, and although it dramatically improves the PRR as already seen, it is still not able to avoid all collisions. If a transmission is not detected by the CAD and the LN transmits, then both packets may interfere destructively and may not be received by the RN. For an interval of 128 ms the PRR was 96.6% in the CAD test. The interval in the current test is 6.4 seconds, which is fifty times larger than the 128 ms interval, but, on the other hand, it also has another transmitter added. The probability of collisions decreases as the interval increases, but it with one more transmitter. When observing the independent PRRs for each of the three LNs, we observe that the differences are at most 1.4%, which is not considered significant.

Another reason for such a decrease when compared with the linear topology is that packets from either of the LNs may arrive at the RN while the RN is transmitting, either a full dump, where there is no difference with the linear topology, or an application data packet. Note that in the linear topology this was unlikely to happen considering that there was no randomness associated with the transmissions of the LN. Hence, when a packet was received by a RN, it would be quickly processed and forwarded if the wait time for the previous packet had elapsed. As the RN removes the routing overhead, the wait time is lower for the RN than for the LN and thus, when the RN would be forwarding the packet from the LN, the same LN would be waiting to transmit the next packet. This stability is only compromised when a beacon is received and a full dump must be sent.

All the tests so far were conducted with a frequency channel separation between the gateway and the RN, and the RN and the other RNs or LNs. A LN transmits, for instance, in 868.5 MHz. The packet travels through the multi-hop network in the same frequency until reaching a RN that is a gateway neighbor.

The last RN in the path, transmits on a different frequency to prevent collisions at the gateway, due to the nodes being close to each other. We set up a last test to determine the effect of having the three LNs associated with the bottleneck possibly colliding with the transmission from the RN to the gateway since no frequency separation was used during this test. Note that when the RN performs a CAD and detects activity, it backs off, loosing the incoming packet from the LN and causing a delay which contributes to decrease the throughput. The results are presented in Table IV.

TABLE IV: PRR for the experiment with a bottleneck topology composed by three LN and one RN with no frequency separation between links.

	Application Server			RN1		
	LN1	LN2	LN3	LN1	LN2	LN3
PRR	89.0%	92.0%	81.2%	90.4%	93.6%	84.4%
	87.4%			89.5%		
Throughput	29.16 (bit/s)			-		

The throughput and the PRR were lower than the previous, confirming the expected lower performance of the not using frequency separation. The difference in the PRR between the three LNs, measured at RN1, was at most 10.8% (between LN2 and LN3). Such difference started on link LN→RN and was not significantly changed in the link RN→Gateway. This could be attributed, for instance, to the mean of RSSI values at the RN be the highest for LN2 and the minimum for LN3. As we keep track of the receptions at the RN, it was possible to verify that the RSSI values of LN2 were agglomerated in the highest values while the RSSI measurements of LN3 were agglomerated in the lowest values.

VIII. Conclusions

The long range of LoRa may still propagate with flaws in harsh environments. In one of our tests, we proved the existence of dark spots in a city with different building heights and changes in elevation, with a gateway placed in one of the highest points in the city. On dark spots there was loss of consecutive packets. In most of the cases, the receptions immediately before and after a dark spot did not indicate a RSSI close to the sensitivity threshold for that spreading factor. Thus, one could not predict the loss of packets that followed. Smart meters that use a wireless link and are located under the ground or in locations with strong electromagnetic attenuation may have the same or worse behavior. For the previous scenarios, a multi-hop solution is an economic way of extending a gateway coverage. Furthermore, a user may not have access to the network infrastructure, e.g. to the gateways.

We proposed a routing solution to be integrated with the already existent LoRaWAN network for uplink transmissions. The protocol was assessed using the PRR and throughput and was tested in linear and bottleneck topologies. The overall results of the multi-hop experiments show, firstly, that a routing solution complying in the LoRaWAN specification is feasible. Secondly, the obtained values for throughput are satisfactory for most of the applications in IoT, in which uplink packets contain mostly sensor readings. Also, for non critical applications, a PRR of 90% may still be acceptable without worsening data analysis and machine learning. Each transmission is preceded by a CAD analysis to avoid that the packet to be transmitted collides with other transmission. This feature was tested separately in an experience that forced collisions, and it was concluded to greatly improve the PRR from 26.5% to 81.9%. The beacon reception window for the used hardware, the Nucleo board, needs to be adapted to the beacon period in use. A delay of 0.6 ms per second increased in the beacon period was observed.

We conclude that a LoRaWAN multi-hop uplink extension is feasible even with the duty cycle restriction of the 868 MHz band. This solution delivers an improved range at the cost of a reduction in the throughput and PRR, which benefits applications with sporadic traffic, and in which the loss of a few packets is not critical.

A. Future Work

There exist some points that can be explored to continue and possibly upgrade what was accomplished in the present work:

- The RNs were assumed to be powered by an external power source. An analysis on the energy consumption on these nodes may possibly extend the application's scenarios to remote areas where an AC plug is not present.
- Eventually add downlink capability in order to allow LNs to be connected to actuators.
- Integrate a Transmit Power Control (TPC) mechanism.
- Consider new metrics other than the number of hops to chose routes.
- Combine several packets that arrive at an RN and retransmit a single aggregated packet, which contains those packets.

References

[1] Electronic Communications Committee (EEC). ERC Recommendation 70-03, 2016. Accessed on January 7th, 2017.

[2] Semtech Corporation. *AN1200.22, "LoRa Modulation Basics"*, 2015.

[3] Semtech Corporation. *SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver*, 2015. Datasheet.

[4] N. Sornin (Semtech), M. Luis (Semtech), T. Eirich (IBM), T. Kramp (IBM), and O.Hersent (Actility). *LoRaWAN Specification*. LoRa Alliance, July 2016.

[5] Charles E. Perkins and Pravin Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *SIGCOMM Comput. Commun. Rev.*, 24(4):234–244, October 1994.

[6] Martin Bor, John Edward Vidler, and Utz Roedig. *LoRa for the Internet of Things*, pages 361–366. Junction Publishing, 2 2016.

[7] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. 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, Dec 2015.

[8] L. Gregora, L. Vojtech, and M. Neruda. Indoor signal propagation of LoRa technology. In *2016 17th International Conference on Mechatronics - Mechatronika (ME)*, pages 1–4, Dec 2016.

[9] Jetmir Haxhibeqiri, Floris Van den Abeele, Ingrid Moerman, and Jeroen Hoebeke. Lora scalability: A simulation model based on interference measurements. *Sensors*, 17(6), 2017.

[10] HopeRF. *RFM95/96/97/98(W) - Low Power Long Range Transceiver Module V1.0*. Datasheet.