A Low Power LoRa-LoRaWan Relay Function with a Single Input, Single Output Device

Olivier Flauzac
University of Reims Champagne-Ardenne, CReSTIC
olivier.flauzac@univ-reims.fr

Joffrey Herard
University of Reims Champagne-Ardenne, CReSTIC
joffrey.herard@univ-reims.fr

Florent Nolot
University of Reims Champagne-Ardenne, CReSTIC
florent.nolot@univ-reims.fr

Philippe Cola
Bouygues Telecom
pcola@bouyguestelecom.fr

Abstract

Nowadays, more and more information is being collected, either by energy suppliers or by environmental organizations, remotely, through wireless networks. Unfortunately, some equipment is not always in areas properly covered by operators, such as metro tunnels, and is isolated. In this paper, we propose a solution for setting up equipment that will act as a relay between the isolated equipment and the operator's equipment. To do this, we use LoRa technology to collect information from the isolated equipment and the LoRaWAN[1] protocol to send the data to the operator's gateway. Our relay can switch between pure LoRa on the one hand and LoRaWAN on the other while providing a standby function to save energy. We also present the relaying and data collection protocol as well as the results of our experiments and some corrections made in Semtech's LoRaWAN library to achieve our objective. Our solution has been tested for several months with an industrial company and an IoT operator to ensure that the solution presented in this paper worked properly.

1 Introduction

Sensors networks are widely used to collect environmental data, to monitor infrastructures such as buildings, roads or bridges. That can be for example temperature, barometrical weight, that can be recovered. These networks can also be used to border intrusion detection and surveillance like in [7]. One of the most wireless sensor network (WSN) architecture used is Linear Sensor Networks (LSN) [13] as illustrated in Figure 1. This network topology puts sensors in a linear form. The Linear sensor networks have gained much attraction of the researchers due to their several positive aspects including easy deployment for linear structures and robustness in different environments. But with this architecture, if a node is out of the gateway's reach, we are no longer able to collect its data. We, therefore, propose in this article, to add a relay function to collect data of these nodes. The relay function is implemented on the same node that the classical node collects data.

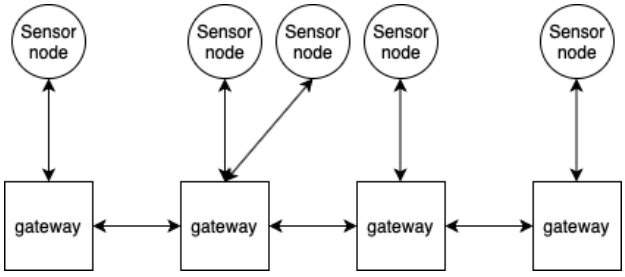

Figure 1. Classical Linear Sensor Network

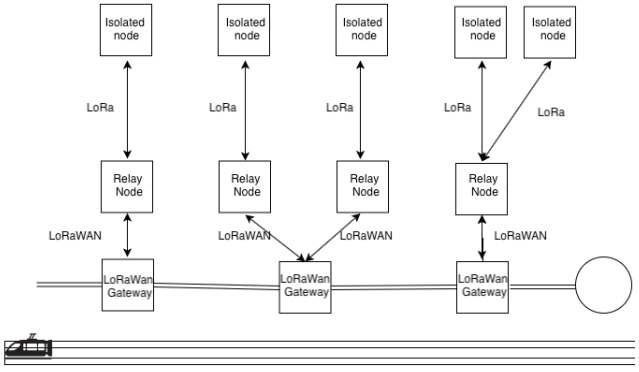

Figure 2. Linear sensor network example for railway

There are remote sensor arrange designs worked around an access point or specially appointed system. A specially

[1] Specification LoRAWAN : https://lora-alliance.org/resource-hub/lorawanr-specification-v103

International Conference on Embedded Wireless Systems and Networks (EWSN) 2020
17–19 February, Lyon, France
ISBN: 978-0-9949886-4-5

appointed system is a system made out of individual device discussing legitimately with one another. In any case, it is important to send information to a central network. The most commonly used protocol for collection is LoRaWAN in our architecture (i.e. Figure 3). We have therefore based in our article as a radio protocol the LoRa RAW and the LoRaWAN.

We base our investigation on Figure 3 engineering. We assume a LoRaWAN Gateway exists and it is constantly associated with the Network Server. A Relaying node can trade information with the LoRaWAN Gateway and an Isolated node that can just speak with the relaying node. An isolated device is characterized as a node that is unable to join a LoRaWAN Gateway. The main issue that shows up is the restricted inclusion zone of the LoRaWAN Gateway. In this way, all isolated nodes must send their information through the relaying node. We consider that the only information the isolated

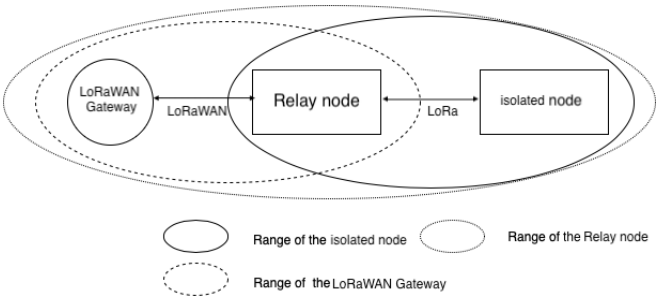

Figure 3. Case of a remote LoRa device

device has at its disposal is an identifier. In this paper, we propose a solution to gather data from a LoRa node through a LoRa / LoRaWAN relay node. This relay node can send all data through the LoRaWAN protocol to any LoRaWAN gateway. One of the main problems is to synchronize the isolated node and the relay node. This problem comes from wireless technology. Whenever there can be radio impedance because of synchronous correspondences between nodes. It is, accordingly, important to synchronize trades to keep away from this. Even though they exist an answer to this issue in some wireless technology, similar to Wifi a lot of different wireless technology like LoRa simply utilizes an ALOHA like the strategy used in LoRaWAN [1]. The main problem of our study is the need to synchronize a set of isolated nodes with relay nodes while respecting the duty cycle imposed by the ETSI EN300.220. Our article is constructed as follows: first of all, we have a state of the art, followed by the description of our LoRa-LoRaWAN relay protocol as well as the experimentation. This section will further develop the problems encountered in development and the solutions. We will then discuss the power consumption and power-saving aspect before concluding.

2 Related works

In the literature, there are several synchronization solutions for the ad-hoc network. In [4], the authors propose a novel solution based on distributed synchronization that enables the connectivity of massive devices without any external reference timing agent. A common beacon with the same chirp-like signature is broadcasted by every device and the collision with the others represents a compound signature that embeds the synchronization reference of the network. Beacon's collisions are exploited here to drive the IoT network to global time and carrier frequency synchronization. The potential problem of this solution is the cost of a broadcast. Moreover, the solution does not consider a multi-hop data access solution. However, the synchronization solution works while minimizing the signaling overhead by using the same signature in synchronization beacon to take benefit of collision as shown by the author. In [14], a clock synchronization architecture for IoT access is presented, with high accuracy. The authors also discuss the integration of the PTP (Precision Time Protocol) and their measurement of the time required for clock distribution to increase the accuracy. To synchronize an ad-hoc WSN, there are many methods based on slotting, which describe the transmission and reception optimization procedures. Especially used via ALOHA, the packet transmission times are synchronized between the different nodes of the network. This is defined in a second version of the ALOHA protocol in paper [8]. Also, in the event of a collision, the transmission is made after a random number of slots. The transmission always takes place at the beginning of the slot, i. e. the transmissions are synchronized with the system clock, and the ACK (acknowledgment of receipt) will be received at the end of the current slot. In [6], the authors provide a resource-efficient and stable mechanism to synchronize stations in a highly mobile ad hoc network, based on a slotted ALOHA medium access. The authors carried out several simulations in realistic traffic situations which is indicate good performance of the proposed scheme applied to a slotted TDMA protocol. There are also different ways of using ALOHA in literature. For example, the Spread Slotted ALOHA (SSA) could reduce the number of packet retransmissions and hence increase the number of information bits that can be transmitted over a channel under higher traffic conditions[9]. The system performance was evaluated according to the different amounts of error-correcting (ECC) and detecting (EDC) capability of the BCH code on an environment with noise (AWGN) and co-channel interference from other users trying to access the hub. However, this solution uses a Forward Error Correction code, which represents a drawback when the traffic condition is low. In the same way of avoiding collisions, [2] is based on the implementation of a Multi-Dimensional Slotted Aloha MAC Protocol for Low-Collision High-Throughput. The authors show in this work that their protocol which uses three-dimensional reservations, namely time, code and frequency, increases system throughput and reduces the collision rate. They propose ways of improving how to reduce collisions in noisy channels. Once again their work is based on ALOHA. The authors of [12] is based on a slotting logic for a TDMA based Ad Hoc network, with a decentralized way and in a wireless network to eliminate interference. The solution they propose is supported by simulations that show the convergence of their solution. A time slot synchronous strategy to help continuous arranged TDMA for a multi-hop ad hoc wireless system is described in [15] as the last articles cite before. In this one, the authors proposed a dis-

tributed synchronous method for ad hoc networks. Every hub redundantly remedies its nearby schedule vacancy reference by a modification, which is processed by its availability distinction between those of its neighbor hubs. The problem raised in our article is a problem for which industrialists have a great need. We can see several examples in the literature of real-time multi-hop solutions. For example, in this article [11], the authors used sensors with Bluetooth Low Energy (BLE), a short-range wireless protocol, as a radio protocol. This technology reduces power consumption and provides a configuration method for the BLE standard that guarantees bounded message latency on a star topology. Secondly, they propose a protocol working on top of BLE that allows for meshed topology while maintaining bounded latency. In [10], Multi-hop Real-time BLE (MRT-BLE) is proposed. It is a real-time protocol, developed on top of BLE, to build low-cost Industrial Wireless Sensor Network (IWSN) with mesh topology. The motivations of the authors are to provide real-time functionality into a Bluetooth mesh standard. To remove this limitation, they have implemented priorities. They studied what happens, in the worst case, during the construction of the network and make real measurements. In [3], the authors measure the impact of taking collisions into account in a dense network. They show that the superposition of signals provides remarkable improvements in synchronization compared to conventional non-collision based synchronization, even on network settings that degrades the performance of the synchronization. They talk about the most robust ways to avoid overcutting in a model where collisions are considered.

3 The LoRa-LoRaWAN relay protocol

We assume we have at least one isolated node, called LoRa Node in Figure 4, a relay node, and a LoRaWAN gateway like in Figure 3. The relay node has just to be compatible with the LoRaWAN protocol and can be reached by the LoRaWAN gateway of the provider. In our work, we respect the specification version 1.0.3. The isolated node has just to be able to communicate in LoRa and be in the range of the relay node. In the beginning, the relay node uses normal Join procedure define in the LoRaWAN protocol. This allows for generating the necessary session elements called NwkSkey and AppSkey. Then the relay node will listen during a defined RX window. During this period it can be synchronized via a chosen mechanism. Next, the relay node initializes a local synchronization system with the isolated node. In our solution, the communication is not synchronized. So, to receive the data from the isolated node, the relay node has to be in RX state and when the relay node sends data to the isolated node, it has to be in TX state. The isolated node and the relay node have to be synchronized with each other.

When the relay node sends the information up to the network server through the GW, we must be able to identify the real source of the data. Is the data from the isolated node or the relay? Either the data is sent with the identity of the isolated node and the feedback, from the provider is therefore carried out after a Join request procedure. Or the identity of the sender is included in the payload sent by the relay node as illustrated in Figure 4. In that case, we have not to make

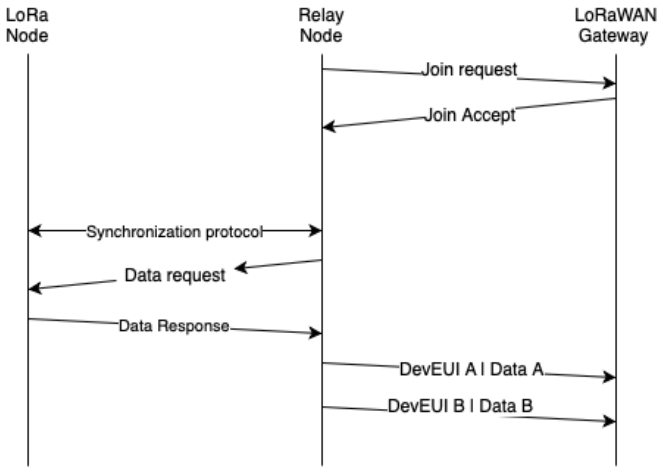

Figure 4. Global mecanism

a new join procedure. Moreover, that limits the calculation of session elements and encryption costs. Moreover, we can also talk about another problem, the aggregation. If the data of the isolated node can be aggregated with the data from the relay node then a message is saved. The different steps of our algorithm at the relay node are:

1. Initiate the Join Request procedure.
2. Listen and transmit in LoRa to synchronize with an isolated node.
3. Transmit the information collected by the isolated node to LoRaWAN.
4. Transmit the information collected by the relay node.
5. Start over at the next RX window.

For the isolated node that's possible, to sum up like this :

1. Detect its role: if it is isolated
2. Listen and transmit in LoRa to synchronize with a relay node.
3. Transmit the information to the relay when the node is asked.
4. Start over at the next RX window.

We will now move on to the development part and therefore practical.

4 Experimentation

To carry out the procedure in terms of programming, there are several things to review in the Semtech's LoRaWAN library. The first step does not change anything in a normal procedure. However, when changing the radio listening mode code, that deletes all session elements (i.e. NwkSKey, AppSkey) which will initialize a new Join request procedure and the power consumption is impacted. It is, therefore, necessary to save the session elements between Step 1 and Step 2 for the relay node, of the procedure described in Section 3. We will then see which method to use to avoid redoing the Join.

In Step 2, this is where the synchronization or pairing process takes place. This is to trigger a future RX window which

will allow collecting the information from the isolated node. Once the data from the isolated node is retrieved. The node will, therefore, perform steps 3 and 4. It is, therefore, necessary to restore the session elements previously mentioned. This is not enough. During our tests on the LoRaWAN platforms of The Things Network (TTN) and Spot [2], the first two messages from the Join request, the first uplink and downlink appear without any problem on TTN. But it only appeared on the Spot platform the Downlink message, without the content of the uplink repatriated. As shown in the following Figure 5 and 6.

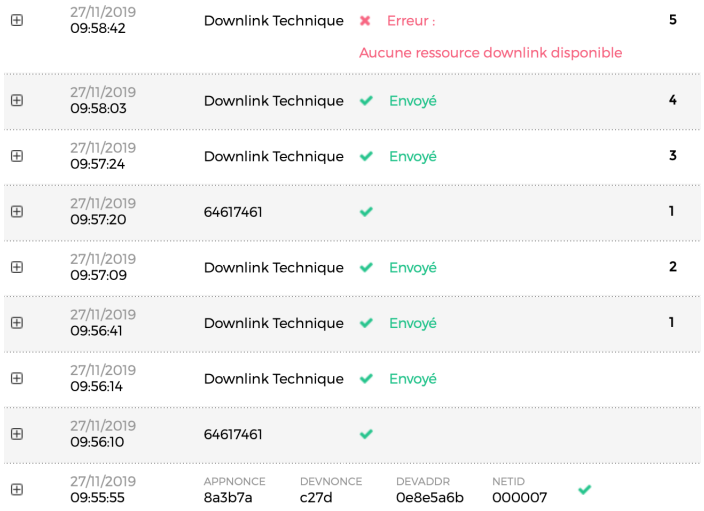

Figure 5. Uplink after restoring session element without uplinkCpt up to date 1/2

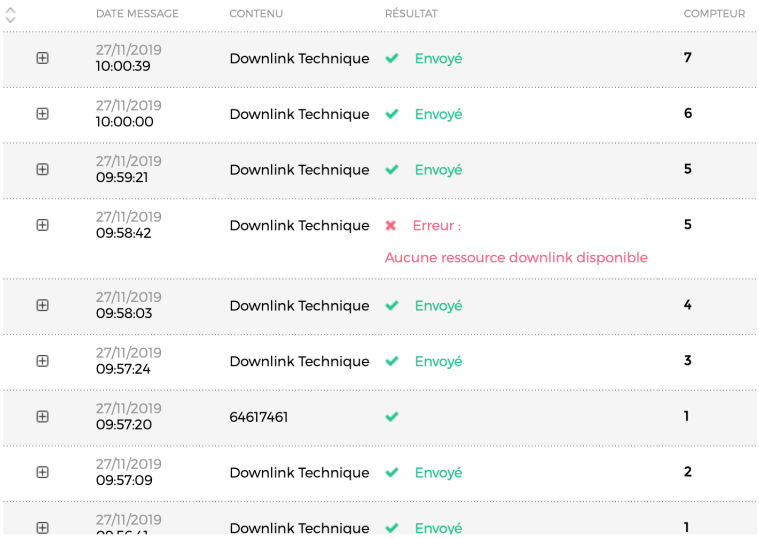

Figure 6. Uplink after restoring session element without uplinkCpt up to date 2/2

To find the source of this problem, we have read the frames sent in LoRaWAN by the relay node. After collected these packets and separated each field, thanks to LoRaWAN packet decoder [3], we obtain the LoRaWAN packets of Table 1. The only difference between the 2 packets, it's the FCnt value. This erroneous FCnt value is the reason for the problem that appeared in Figures 5 and 6. When restoring session elements, it is also necessary to remember to save the current state of the uplink counter. In the program, this variable is named uplinkCounter and corresponds to the packet in the FCnt field. To solve this problem, each time the radio protocol is changed to LoRaWAN, we use the Activation By Personalization (ABP) functions.

Fields	Packet 1	Packet 2 and more
Message Type	Data	Data
PHYPayload	4032180F0E8000000231D17 93997B7AA376FE3632A4B0D	4032180F0E80010002F4DA24 0493C487880F706FC5CAB3
MHDR	40	40
MACPayload	32180F0E8000000231 D1793997B7AA376FE3	32180F0E80010002F4 DA240493C487880F70
MIC from packet	632A4B0D	6FC5CAB3
MIC expected	632A4B0D	6FC5CAB3
FHDR	32180F0E800000	32180F0E800100
FPort	02	02
FRMPayload encrypted	31D1793997B7AA376FE3	F4DA240493C487880F70
FRMPayload decrypted	70B3D54999AEEF65FF15	70B3D54999AEEF65FF15
DevAddr	0E0F1832	0E0F1832
FCtrl	80	80
FCnt	**0000 (Big Endian)**	**0001 (Big Endian)**
FOpts	null	null
Message Type	Unconfirmed Data Up	Unconfirmed Data Up
Direction	up	up
FCnt	**0**	**1**
FCtrl.ACK	false	false
FCtrl.ADR	true	true

Table 1. Different LoRaWAN packet send by relay node

The management of messages received in LoRa is dynamically managed in memory. It is customary to free up allocated memory that has become useless over time. However, it appears that despite the call for this liberation function, it is not being carried out. To make sure that this problem comes from memory allocations, the code has been modified by setting limits on the memory allocation. It turns out that the problem still exists. The second remark concerns the way in which the memory has been managed in the program provided by the STM32 library. There are memory allocations executed and not released inside the library. Because of the change in radio context from LoRa RAW to LoRaWAN. This action is not provided for in the LoRaWAN implementation or specification [4].

During our tests, we also tried to see if the behavior obtained on the Spot platform would be different on that of The Things Network. Which was the case, the Downlink, and Uplink appear? This with the same code, the only difference is the one of the AppEUI, AppKey necessary to change the destination platform. It is necessary at this stage to see why the behavior is different between these two platforms. If we refer to the documentation it is explained that at reception

[2]https://spot.objenious.com/

[3]https://lorawan-packet-decoder-0ta6puiniaut.runkit.sh

[4]Specification LoRAWAN : https://lora-alliance.org/resource-hub/lorawanr-specification-v103

the counter is supposed to be kept in synchronization with the one previously transmitted. The value received has incremented compared to the current counter value and is less than the value specified by MAX_FCNT_GAP1 after considering counter rollovers. If this difference is greater than the value of MAX_FCNT_GAP then too many data frames have been lost then subsequent will be discarded. The FCnt is not incremented in case of multiple transmissions of an unconfirmed frame, or in the case of a confirmed frame that is not acknowledged.

This would mean that The Things Network[5] does not use the same values of the different MAX_FCNT_GAP. Or that a choice has been made to not take into account this gap. It remains to talk about a last aspect at the development level is the transmission of data from the isolated node and the Relay node using the LoRaWAN.

Data rate	Configuration SF and Bandwidth	Max Payload (M) in bytes	Max application payload (N) in bytes
0	LoRa: SF12 / 125 kHz	59	51
1	LoRa: SF11 / 125 kHz	59	51
2	LoRa: SF10 / 125 kHz	59	51
3	LoRa: SF9 / 125 kHz	123	115
4	LoRa: SF8 / 125 kHz	230	222
5	LoRa: SF7 / 125 kHz	230	222
6	LoRa: SF7 / 250 kHz	230	222
7	FSK : 50 kbps	230	222
8...15	RFU	not defined	not defined

Table 2. EU863-870 maximum payload size based on Regional Parameters

Knowing that the maximum size of the payload depends on several factors as shown in Table 2, for a given spread factor, if the data to send to the LoRaWAN gateway is greater than the maximum payload value, then the two messages cannot be aggregated and we will derogate from the duty cycle rule, defined in the ETSI EN300.220. In this case, we can use the DutyCycleReq command. This command is used by the network coordinator to limit the maximum aggregated transmit duty cycle of an end-device. The aggregated transmit duty cycle corresponds to the transmit duty cycle overall sub-bands. It exists a parameter called MaxDutyCycle which is defined between values 0 and 15. A value of 0 corresponds to "no duty cycle" limitation except the one set by the regional regulation. It is, therefore, necessary to modify either the duty cycle on each sensor or to give an adaptation via the LoRaWAN protocol.

5 Power consumption and security

To save the power of each node and to limit the number of emission from the relay node, one solution is to aggregate small messages send by the end device. To achieve this goal, the end device can be configured to send only small messages. Concerning the safety aspect, which is closely linked to energy consumption by calculation principle, if we want to encrypt the exchanges made between, we can imagine something based on a Diffie Hellmann exchange [5]. However, the search for a sufficiently large prime number will be reduced by the sensor architecture, which does not work on a 64-bit processor but 32-bit. It would then be necessary to implement or use an existing library for big integers. All this to generate a common secret that will be defined as a basis for the AES-128 encryption key, which is an existing module on the nodes. This encryption mode already exists due to the LoRaWAN standard which requires it. There is another possibility to encrypt between the relay node and the isolated node. In the future LoRaWAN standard, a secure element is described that could play this role [6].

6 System Integrity

6.1 Fault on system nodes

In this section, we will discuss the problems of data redundancy and therefore data loss. During the synchronization mechanism that takes place between the relay nodes and the isolated node, it will be necessary to set up a certain number of timers. By basing each exchange on Timeouts, for example when retrieving the data that in our example in Figure 4. Alternatively, some tests could be used to retrieve information from the isolated node to save energy. This could, therefore, allow the detection of a failure at the isolated nodes. We can also consider at the level of possible errors the disappearance of a relay node. We can set up this same kind of timer mechanism to avoid never reporting our data.

6.2 Additional nodes

What happens when there are several relay nodes in the system within the range of a single isolated node? A limit can be set on the number of nodes that a relay node could manage concerning the duty cycle. As for the isolated node during the synchronization process, it will have to answer the choice to be made as to the relay it will have to choose like in Figure 2. The main problem that would emerge from several isolated nodes to relay would be the local synchronization on sliding RXs for each isolated node. If the synchronization mechanism is not implemented, this could have an impact on the number of colliding messages. This would result in a higher number of messages being sent, which would impact the battery. Several parameters would come into play, the number of isolated nodes attached to a relay node, the number of messages to retrieve per cycle (i.e Duty Cycle).

So here is a graph showing the difference between aggregation and non-aggregation mode at the message level. We can notice that in the following figures that a linear aspect stands out clearly on the variation of the number of messages on the part of a relay node whatever the mode (with aggregation or without aggregation). What we can also notice is the number that as expected is bigger and grows much faster without the aggregation mode, Fig 7. What is described as IN on the curves are Isolated Nodes, A means aggregation and NA without aggregation? What can, therefore, be considered significant enough. Indeed, by adding relay node data to those of isolated nodes. The number of messages in cluster mode is reduced to the same number as if the isolated node were relays nodes and therefore nodes connected as LoRaWAN Gateway.

The objective was to check if the intuitive idea is right: the architecture with intermediary relay node and aggrega-

[5]The Things Network https://www.thethingsnetwork.org

[6]https://lora-alliance.org/resource-hub/lorawanr-specification-v11

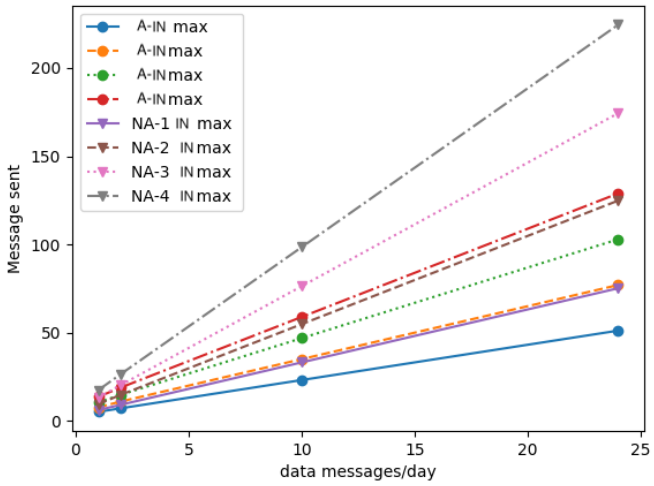

Figure 7. With aggregation vs without aggregation

tion returns, in the number of messages and thus the lifespan of the network, to the model without relay node, the model or each client speaks live with antennas always up.

7 Conclusion

In this paper, we presented a solution to gather data of an isolated node from a LoRaWAN network. To do that, we implemented a new LoRa to LoRaWAN relay protocol in a Single Input Single Output device. We also proposed a synchronization solution to save energy of each device and to avoid collisions. Our relay node is fully LoRaWAN compatible, even if it switches often in LoRa. The isolated node has only to the ability to receive and transmit in LoRa, without necessarily respecting LoRaWAN standard and ETSI EN300.220. However, there are still several points to study like fault tolerance. It would be interesting to analyze with experimentation on differents microcontroller and LoRa chipset if our solution is always efficient. It would also be helpful to reduce the power consumption of the relay nodes. An industrial partnership is in progress to implement the first version of our solution.

8 Acknowledgement

This work is funded by Objenious and Bouygues Telecom.

9 References

[1] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne. Understanding the limits of lorawan. *IEEE Communications Magazine*, 55(9):34–40, Sep. 2017.

[2] F. Alassery, W. K. M. Ahmed, and V. Lawrence. Mdsa: Multi-dimensional slotted aloha mac protocol for low-collision high-throughput wireless communication systems. In *2015 36th IEEE Sarnoff Symposium*, pages 179–184, Sep. 2015.

[3] M. A. Alvarez and U. Spagnolini. Collision vs non-collision distributed time synchronization for dense iot deployments. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.

[4] M. A. Alvarez and U. Spagnolini. Distributed synchronization for massive iot deployments. In *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 1–6, March 2017.

[5] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.

[6] A. Ebner, H. Rohling, M. Lott, and W. Halfmann. Decentralized slot synchronization in highly dynamic ad hoc networks. In *The 5th International Symposium on Wireless Personal Multimedia Communications*, volume 2, pages 494–498 vol.2, Oct 2002.

[7] E. Felemban. Advanced border intrusion detection and surveillance using wireless sensor network technology. *International Journal of Communications Network and System Sciences*, 6(5):pp. 251–259, 2013.

[8] I. Gitman. On the capacity of slotted aloha networks and some design problems. *IEEE Transactions on Communications*, 23(3):305–317, March 1975.

[9] O. Gonzalez and R. Kohno. Performance of a spread slotted cdma/aloha with hybrid arq system with variations on the fec code capabilities. In *VTC2000-Spring. 2000 IEEE 51st Vehicular Technology Conference Proceedings (Cat. No.00CH37026)*, volume 3, pages 2262–2266 vol.3, May 2000.

[10] L. Leonardi, G. Patti, and L. Lo Bello. Multi-hop real-time communications over bluetooth low energy industrial wireless mesh networks. *IEEE Access*, 6:26505–26519, 2018.

[11] G. Patti, L. Leonardi, and L. Lo Bello. A bluetooth low energy real-time protocol for industrial wireless mesh networks. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 4627–4632, Oct 2016.

[12] Qi Yang and Xanghong Shi. An interference elimination method for decentralized slot synchronization in tdma-based wireless ad hoc network. In *2007 International Symposium on Intelligent Signal Processing and Communication Systems*, pages 236–239, Nov 2007.

[13] A. M. Sarafi, G. I. Tsiropoulos, and P. G. Cottis. Hybrid wireless-broadband over power lines: A promising broadband solution in rural areas. *IEEE Communications Magazine*, 47(11):140–147, November 2009.

[14] Shulong Wang, Yibin Hou, Fang Gao, and Songsong Ma. A novel clock synchronization architecture for iot access system. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pages 1456–1459, Oct 2016.

[15] Q. Yang and J. Shi. A slot synchronous method and performance analysis of tdma ad hoc network. In *2007 International Workshop on Anti-Counterfeiting, Security and Identification (ASID)*, pages 340–343, April 2007.