Enhanced Dynamic Duty Cycle in LoRaWAN Network

Norhane Benkahla, Hajer Tounsi, Ye-Qiong Song, Mounir Frikha

Enhanced dynamic duty cycle in LoRaWAN network

Norhane Benkahla[1], Hajer Tounsi[1], Ye-Qiong Song[2], and Mounir Frikha[1]

[1]Higher school of communication Tunis, Tunisia
{*norhane.benkahla, hajer.tounsi, mounir.frikha*}*@supcom.tn*
[2]LORIA University of Lorraine Nancy, France
ye-qiong.song@loria.fr

Abstract. LoRa's long-range and low-power features have made it an attractive candidate for IoT devices in various fields. In this work, we present an enhanced LoRaWAN protocol. LoRaWAN MAC protocol is characterized by the restrictive use of the channel, limited by the regulatory authorities to a 1% duty cycle per cycle (i.e., 36 seconds per hour) per node. This regulation penalizes the nodes which require a channel access time greater than the limited duty cycle to occasionally transmit a large amount of data such as video surveillance or access control information in applications like smart school surveillance. However, some other nodes like environment sensors sharing a same LoRaWAN server may send very small amounts of information (e.g. temperature, humidity, ...) and under-use the authorized activity time of 1% duty cycle. Hence the idea of implementing an activity time sharing mechanism among nodes that allows devices to borrow additional activity time from a device or set of devices that have completed the transmission of their packets and do not need the remaining time of the corresponding duty cycle. Our work extends and improves the activity time sharing mechanism initially proposed in [1]. Instead of FIFO sharing-time allocation based on a global activity time, which may lead to the starvation of the nodes that are others than that in the head of FIFO line, we propose a new time allocation algorithm based on the classification of the different requests according to their needs in terms of their QoS requirements. It allows to satisfy a larger number of nodes requiring extra time, with less control overheads while ensuring fairness. Our time-sharing algorithm has been implemented and tested on the wasp-mote chip of libelium, showing the performance improvement and its practical usability.

Keywords: IoT, LPWAN, LoRa, Duty cycle, QoS

1 Introduction

IoT devices are used in a wide variety of intelligent systems: smart cities, smart transport, smart industry applications, security and emergencies, e-Health applications, etc. [2]. The deployment of these IoT devices is limited by challenges such as range, cost, autonomy, etc. The use of traditional long-range and high-speed (3G, 4G) technologies for IoT is constrained by the relatively high cost and high energy consumption. Low power wireless technologies such as low energy Bluetooth or ZigBee are limited in their short range and are not suitable for applications requiring deployment in rural or isolated environments. In order to reduce the complexity and obtain longer range at low cost, the concept of extended networks low-power, low-bandwidth and long range (LPWAN) [3, 4] is proposed. This type of networks is suitable for equipment that requires several years of autonomy and does not require high throughput. It is therefore promising for the Internet of Things. Among several LPWAN technologies, Semtech's LoRa is an interesting one, which can reach a range of 5 km in the urban area and 15 km in the rural area exploiting license-free sub-GHz frequency bands [5]. The flexibility of LoRa's long-range transmission is in expense of a low throughput and a limited channel activity time. In fact, the ETSI regulations impose to each node a maximum use of duty cycle of 1% per cycle, that means 36 sec per hour. If this duty cycle limitation does not bother traditional low data rate IoT applications, it may exclude important applications where some sensors occasionally need more than 36 seconds per one hour cycle to transmit their urgent data following burst events detection, exceeding the per cycle authorized channel activity time. To introduce more flexibility while still keeping the total duty cycle limitation of an application composed of n sensor nodes, in [1], Pham has first introduced the idea to allow some throughput demanding nodes to occasionally exceed their 1% limitation (a kind of ”last chance” solution) while still keeping the global duty cycle of the application under $n \times 1\%$. The basic mechanism, called activity time sharing, consists in broadcasting a ”global activity time” G_{AT} informing each node the total remaining available time of the current cycle. A node needing more time will be allowed to use additional time until this G_{AT}. This approach provides indeed more flexibility to better manage the QoS issues, as demonstrated in [1]. It arises however some additional issues. One is the risk to starve the flowing nodes, especially if we assume that devices share 100% of their local activity time or the LoRa gateway allows the use of the maximum G_{AT} (i.e., $\alpha = 100\%$ where α is the G_{AT} ratio allowed for usage). Another issue is related to the way to broadcast G_{AT}, which introduces overheads and needs a synchronized radio wake-up mechanism.

In this paper, we leverage those limitations. We extend and improve the activity time sharing mechanism proposed in [1]. Instead of FIFO sharing-time allocation which may lead to the starvation of the nodes other than that in the head of FIFO line, we propose to define three classes of nodes (classical, donor and requester) and a new time allocation algorithm based on the classification of the different requests according to their needs in terms of QoS. It allows to satisfy a larger number of nodes requiring extra time, with less control overheads while ensuring fairness and without impacting other nodes' due time.

The rest of the paper is organized as follows. In Section 2, we present the LoRa/ LoRaWAN technology. In Section 3, we review and discuss the principle of the time activity sharing mechanism presented in [1]. In Section 4, we present our solution and proposed improvements to solve the above-mentioned issues by describing the new activity time sharing mechanism and the time allocation algorithm according to a classification strategy that depends on a QoS criterion. Our implementation and our experimentation scenario will be discussed in Section 5 by showing how the mechanism is adapted to the devices and the server. We conclude in Section 6.

2 LoRa Network overview

In this section, we present the LoRa/ LoRaWAN technology. Even if the terms LoRa and LoRaWAN are used interchangeably but they refer to two different concepts in the network. In fact LoRa corresponds to the PHYSICAL layer and precisely to the modulation technique used and LoRaWAN defines the LoRa MAC layer.

2.1 LoRa Modulation: Physical layer

LoRa technology is a proprietary physical modulation designed and patented by Semtech Corporation. It is based on Chirp-Spread Spectrum (CSS) modulation [8] with Integrated Forward Error Correction. LoRa operates in the lower ISM bands (EU: 868 MHz and 433 MHz, USA: 915 MHz and 433 MHz). It offers different configurations (data rates, transmission range, energy consumption and resilience to noise) according to the selection of four parameters which are Carrier Frequency (CF), Bandwidth (BW), Coding Rate (CR) and Spreading Factor (SF). Each LoRa symbol is composed of 2^{SF} chirps [7], where SF represents the corresponding spreading factor in the range of 6 to 12. SF6 means a shortest range, SF12 will be the longest. Each step up in spreading factor doubles the time on air to transmit the same amount of data. The use of a larger SF decreases the bit rate and increases the time on Air (ToA) which induces greater power consumption.

In fact, in the case of a 125khz bandwidth and a coding rate 4/5, the bit rate is equal to 250 bps for SF12 and it is equal to 5470 bps for SF7 [9]. With LoRa, transmissions on the same carrier frequency but with different spreading factors are orthogonal, so there is no interference.

2.2 LoRaWan: LoRa Mac Layer

Unlike the proprietary LoRa protocol, LoRaWAN is an open protocol defined by LoRa Alliance. A LoRaWAN network is based on star-of-stars topology and is composed of three elements.

- End devices: nodes that send uplink (UL) traffic and receives Downlink (DL) traffic through LoRa gateways. The communication between end-devices and gateways is based on LoRa modulation.
- LoRa gateways dispatch the LoRaWAN frames received from end devices via IP connections (using Ethernet, 3G, 4G or Wi-Fi etc.) to a network server.
- A network server decodes the packets, analyzes information mined by end devices and generate the packets that should be sent to end devices.

LoRaWAN end devices implement three classes: a basic LoRaWAN named Class A and optional features (class B, class C) [13].

LoRaWAN operates in ISM bands (863 - 870 MHz band in Europe) which are subject to regulations on radio emissions, thus radios are required to adopt either a Listen-Before-Talk (LBT) policy or a duty cycled transmission to limit the rate at which the end devices can actually generate messages. The current LoRaWAN specification exclusively uses duty-cycled limited transmissions to comply with the ETSI regulations [9]. In fact, each device is limited to an aggregated transmit duty cycle of 1% that means 36 seconds per hour.

LoRaWAN defines three MAC message types in [13] which are: the join message for connecting a device with a network server, the confirmed message which have to receive an ACK from a network server, and the unconfirmed message without ACK. A MAC payload length varies between 59 and 250 Bytes depending on the modulation rate [9].

3 Related work on LoRa performance enhancement

In order to optimize the performance of a LoRa network and the quality of service, we identified three complementary approaches: 1) parameter selection, 2) data compression, 3) activity time sharing.

3.1 LoRa parameter selection

As explained in previous section, for satisfying a desired performance level, one can choose his configuration by combining the various parameters CR (4/5, 4/6, 4/7 and 4/8), BW (125 kHz, 250 kHz and 500 kHz), SF (from 7 to 12) and TP (2 dBm to 17 dBm), resulting in total 1152 combinations. In [10] the authors studied the impact of LoRa parameter settings (bandwith, coding rate, spreading factor, transmission power, etc.) on energy consumption and communication reliability. They proposed a mechanism to automatically select LoRa tranmission parameters that satisfy the performance requirements. This solution is optimal for a given application scenario, but it is not convenient when traffic dynamically changes.

3.2 Data compression

The authors in [11] were interested in data compression in order to reduce the size of the data sent and thus minimize the transmission time and optimize the energy consumption. A swapped huffman tree coding has been applied to transmit the necessary data with a compression ratio of 52.3%. Data compression has been used in various LoRa sensors in the industry [12] in order to reduce energy consumption and thus reduce the data transmission time that will provide better optimization of the LoRa network. The two studies mentioned above were interested in optimizing energy consumption without worrying about the regulatory constraints relating to the channel occupancy time.

3.3 Activity time sharing mechanism

[1], proposes a mechanism for sharing the channel occupancy time in order to improve the overall performance of the network. We give more details on this mechanism, to which we are interested in our work. [1] proposed an activity time sharing mechanism in a long-range unlicensed LoRa network to face the problem of activity time limitation in the case of video surveillance applications. The proposed mechanism supposes that all devices that will participate in the sharing mechanism register with the LoRa gateway and announce their local remaining activity time (initially can be the total authorized activity time or just a fraction). Thus, the gateway computes the global activity time allowed for usage which can be an addition of the allowed time of each device "*Global Time*" (1) or just a fraction of it. After it informs it to all devices which will share it. This step is performed each cycle (every hour).

As long as this global activity time allows, a node Di that exhausts its duty cycle (allocated activity time) and needs additional time to send its data borrows the remaining time from the global time. A global view of the total remaining activity time is maintained by the LoRa Gateway (LR-BS) on reception of packets and sent back to devices at the appropriate moments.

$$GlobalTime = n \times 36sec \tag{1}$$

In [1], the author did not evaluate nor propose a mechanism for selecting devices that will benefit the shared extra time. Indeed, he limited himself to serving the first applicant. Moreover, [1] assumes that all the nodes participating in the sharing mechanism must be on standby to be able to receive from the gateway the updated information of the global activity time and the list of nodes involved in the loan. Otherwise they must wake-up periodically to receive this update. This would not correspond to the behavior of class A nodes but rather to class B nodes. We believe that the activity time sharing mechanism proposed in [1] improves the quality of service but lacks an additional time allocation mechanism by a priority classification or a strategy that satisfies a larger number of requesting devices taking into account the range of a device and its battery level in the management of the allocation of additional time. In the next section, we will describe our solution to those above-mentioned issues.

4 Synopsis and detailed description of the proposed algorithm

The main idea of our mechanism is to provide devices, that do not use all their maximal allowed activity time, with the ability to share their remaining transmission time to devices that need to exceed the 1% duty cycle restriction in order to provide a better global quality of service. Unlike [1], we will consider two modes of LoRa "Sleep mode" and "Standby mode".

We suppose that the number of packets that each device has to send is known at the beginning of the transmission, so, each device can calculate the needed time on air per cycle instead of using Last packet flag as in [1]. The use of this flag in [1] is to indicate that the device will finish its transmission during the cycle. The management of this remaining time will be in the server which doesn't need to broadcast the information of the remaining global activity time as in [1] since if a node decides to share its remaining activity time then it will no longer need to use it. We also allow all the devices to benefit from the dynamicity of duty cycle when the remaining time is sufficient. We propose that each node informs the server of its role in the sharing mechanism during the OTAA (Over The Air Activation) registration process [13]. We define 3 node's roles:

- ”Classical” devices: those who will neither give nor receive any additional activity time. So, they don’t participate in the sharing mechanism.
- ”Donor” devices: those that do not consume all their activity time during 1 hour cycle, so they will give their remaining time to the devices that will need it.
- ”Requester” devices: they are the devices whose local activity time is not sufficient to transmit their data in a cycle of 1h. They try to benefit from the offer of the potential donors.

4.1 Registration phase

At each cycle, during the registration phase, each node specifies in the ”join request” message its role according to the size of the data to be sent during this cycle and therefore the transmission time that it needs. The estimation of the time on air is calculated according to the Semtech formula (6) [14], where T_{sym} is the time taken to send 2^{SF} chips at the chip rate and is calculated using (2). (3) defines a preamble duration where $n_{preamble}$ is the number of programmed preamble symbols. The number of symbols that make up the packet payload and header is given by (4) where PL is the payload size in bytes, CR is 16 if the CR is enabled and zero otherwise, H is 20 when the header is enabled and zero otherwise and DE is two when the low data rate optimization is enabled and zero otherwise. The payload duration is then the symbol period multiplied by the number of payload symbols which is defined in (5). Finally, the time on air is simply the sum of the preamble and payload duration (6).

Fig.1 illustrates the treatment done by a device to register with the server and be part of the sharing mechanism. In case of the device is a ”Donor” (estimated transmission time<duty cycle), it will also indicate the remaining time it is willing to lend (7). Otherwise, if it is a ”Requester” (estimated transmission time>duty cycle), it specifies the needed time to borrow before starting transmission (7). We propose to encode the information of the role of the equipment and the calculated time (for the loan or the donation) respectively in the fields RFU and FOpts of the MHDR field [13].

$$T_{sym} = \frac{2^{SF}}{BW} \tag{2}$$

$$T_{preamble} = (n_{preamble} + 4,25) \times T_{sym} \tag{3}$$

$$PayloadSymbNb = 8 + max\left(ceil\left(\frac{8PL - 4SF + 28 + 16 - 20H}{4(SF - 2DE)}\right)(CR + 4), 0\right) \tag{4}$$

$$T_{payload} = payloadSymbNb \times T_{sym} \tag{5}$$

$$ToA_E = Tpreamble + Tpayload \tag{6}$$

$$(Requested/Given)time = LimitToA - ToA_E \tag{7}$$

When the server receives the registration message, it saves the received information according to the type of each device in the corresponding table and assigns a priority to the node according to some criteria (battery level, range, etc).

4.2 Description of the activity sharing mechanism

After the registration phase, each node starts to send its data according to the LoRaWAN protocol. The node continues to send information about its participation in the activity sharing mechanism. Therefore, it can possibly change its offer or request dynamically. Also this allows a node that did not participate in the sharing mechanism during the join phase to do so during the transmission of its data. Upon receipt of the registration phase or a data frame from a given node, the server verifies the received information and takes the necessary actions depending on the role of the equipment. So if it is a donor node then the server just checks that the information is stored in the appropriate table otherwise it adds it. If it is a requesting node, then the server checks according to the priority given to it, if it could benefit from additional time then the server grants the requested time,

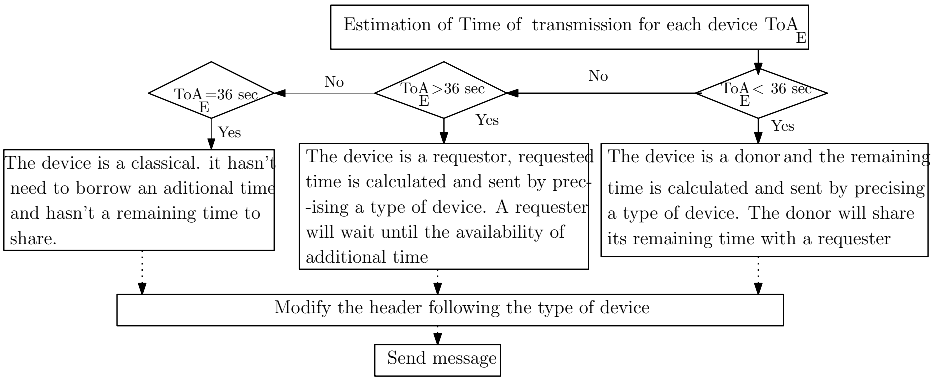

Estimation of Time of transmission for each device ToA E
ToA < 36 sec E
No
ToA >36 sec E
No
ToA =36 sec E
Yes
Yes
Yes
The device is a classical. it hasn't need to borrow an aditional time and hasn't a remaining time to share.
The device is a requestor, requested time is calculated and sent by prec--ising a type of device. A requester will wait until the availability of additional time
The device is a donor and the remaining time is calculated and sent by precising a type of device. The donor will share its remaining time with a requester
Modify the header following the type of device
Send message

Fig. 1. device's state registration

otherwise the server ignores its request. Different criteria can be considered to choose the highest priority requesting node to be satisfied. In our work, we are interested in two approaches.The first one consists of serving first the lowest demand in order to satisfy the maximum number of demands. The second approach is to first satisfy the nodes with the lowest battery to avoid that battery runs out before the end of transmission of their messages in case of equality of the battery level, then we choose the farthest node whose transmission is the most unfavorable. The distance of each device is calculated according to (8).

$$distance(m) = 10^{(RSSIm-ReceivedRSSI)/20} \tag{8}$$

Fig.2 shows the treatment done by the server upon the reception of a frame. Once the request received from a given node can be satisfied (the time offered by the donors is sufficient) then the server updates the remaining activity time of the donors, grants the borrowed time to the requester by sending to it an update message confirming the allocation of the additional activity time. The update message will be sent only for responding to the demands of the requesters. The update message (ACK0 to say "unauthorized borrowing" and ACK1 to say "allocation of additional time is done") must be sent during the two reception windows (class A). After additional time's allocation, the server removes the requester from the appropriate table. In case of both windows are not open, the server will wait for the next opening of both windows reception.

5 Implementation and discussion of scenarios

To evaluate our proposed mechanism, we have implemented it on waspmote SX1272 devices (wasp-mote SX1272 LoRa clients cards and a wasp-mote SX1272 LoRa gateway) [15, 16]. We have integrated a flag in the ACK frame so that the requesters know if their requests have been accepted or not yet. We have tested several scenarios that will be presented and analyzed in the next subsections.

We consider an environment of smart school and smart bus scenario, where children take school buses to get school and to get home from school. The parents need to ensure the security of their children by checking the time of arrival at school or at home. A monitoring system allows a mobile application to notify parents during the get in and get out of their children from the bus. Children are identified by checking their cards and their information are sent to a server through a LoRa network.

5.1 Test Scenario without activity time sharing

We consider a first simple scenario with two devices to verify our implementation. The first (D_1) is dedicated for capturing image. An image of 2000 bytes should be sent in 8

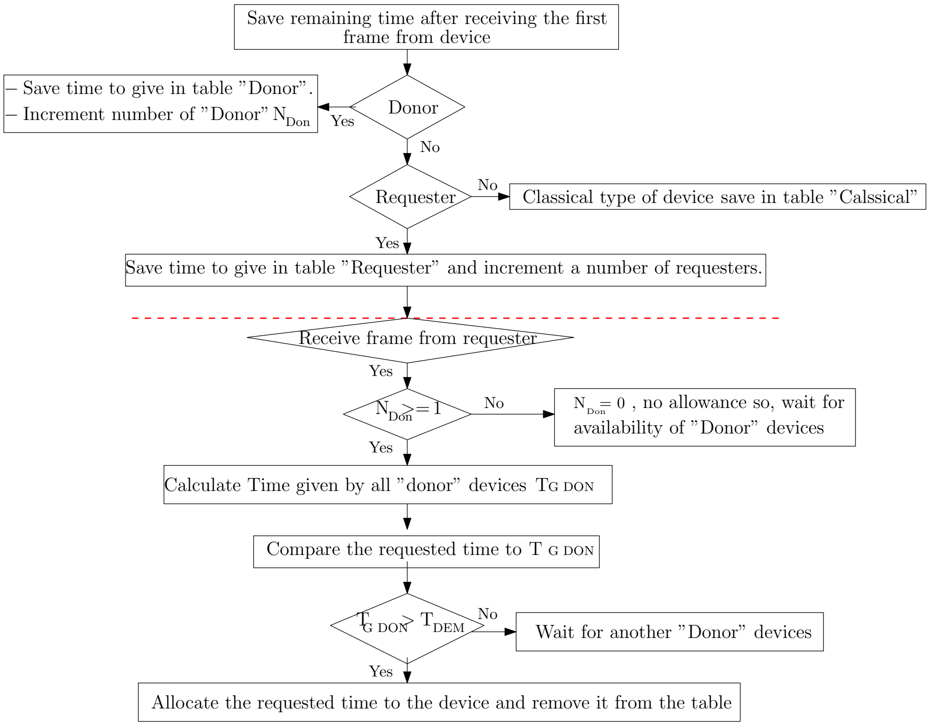

Save remaining time after receiving the first frame from device
Donor
– Save time to give in table "Donor".
– Increment number of "Donor" N_Don
Yes
No
Requester
No
Classical type of device save in table "Calssical"
Yes
Save time to give in table "Requester" and increment a number of requesters.
Receive frame from requester
Yes
N_Don >= 1
No
N_Don = 0 , no allowance so, wait for availability of "Donor" devices
Yes
Calculate Time given by all "donor" devices T_G DON
Compare the requested time to T_G DON
T_G DON > T_DEM
No
Wait for another "Donor" devices
Yes
Allocate the requested time to the device and remove it from the table

Fig. 2. server processing

Table 1. LoRa modes for 250 bytes

LoRa mode	BW(kHz)	CR	SF	250 bytes
1	125	4/5	12	8295.41 ms
2	250	4/5	12	4509.69 ms
3	125	4/5	10	2312.50 ms
4	500	4/5	12	2254.84 ms
5	250	4/5	10	1147.90 ms
6	500	4/5	11	1050.30 ms
7	250	4/5	9	625.15 ms
8	500	4/5	9	312.57 ms
9	500	4/5	8	176.76 ms
10	500	4/5	7	99.90 ms

Table 2. LoRa modes for 100 bytes

LoRa mode	BW(kHz)	CR	SF	100 bytes
1	125	4/5	12	4104.18 ms
2	250	4/5	12	2052.09 ms
3	125	4/5	10	1067.00 ms
4	500	4/5	12	1026.04 ms
5	250	4/5	10	533.50 ms
6	500	4/5	11	504.46 ms
7	250	4/5	9	287.23 ms
8	500	4/5	9	143.61 ms
9	500	4/5	8	79.48 ms
10	500	4/5	7	44.86 ms

packets because of the constraint of the maximum payload length of LoRa packet which is 250 Bytes. The second device (D_2) is a simple temperature sensor which sends 100 bytes' message containing Temperature, date and time. Each device is limited by a maximum transmission time "Limit-ToA = 36 sec/cycle".The devices use LoRa mode 1, that means they use BW= 125kHz and SF= 12. This choice is made because SF12 is the default one in some devices such as the waspmote ones and also because it is the most suitable for long range and in presence of obstacles which is the case of urban areas[17]. Indeed, our tests done in our school where the device was on the 1st floor and the gateway on the 2nd floor, at a distance of about 43 meter using a transmission power of 14dB have confirmed that SF7 and SF8 are not adequate for great distances inside a school. The same test was done using the SF12 under the same conditions, the transmission of all packets was successful.

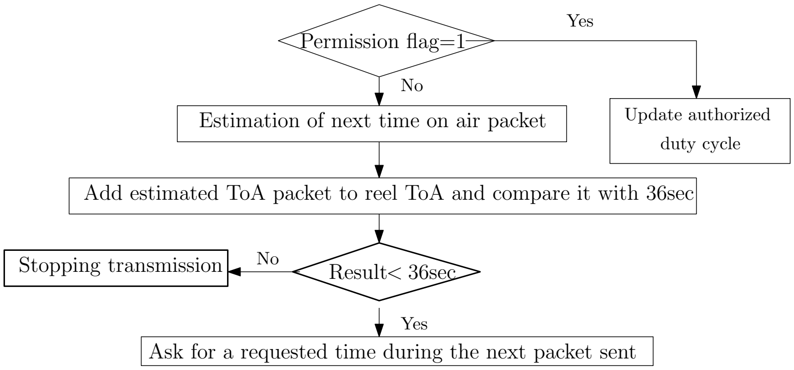

Permission flag=1
Yes
No
Update authorized duty cycle
Estimation of next time on air packet
Add estimated ToA packet to reel ToA and compare it with 36sec
Stopping transmission
No
Result< 36sec
Yes
Ask for a requested time during the next packet sent

Fig. 3. The ACK response in the node

Tables 1 and 2 present the time on air respectively of a 250 Bytes and 100 bytes according to several parameters. The scenario is shown in Fig.4. Table 3 shows the time of transmission consumed after the transmission of each packet for D_1. The time on air of the packet sent by D_2 is $\delta_1 = 2052.09$ ms. These time on air values are the same in all tests. After receiving the join accept, the cycle begins for all the clients in the network.

Table 3. ToA for each packet sent by D_1

ζ_1 (ms)	ζ_2 (ms)	ζ_3 (ms)	ζ_4 (ms)	ζ_5 (ms)	ζ_6 (ms)	ζ_7 (ms)	ζ_8 (ms)
8295.41	16590.82	24886.23	33181.64	41477.05	49772.46	58067.87	66363.28

Sending the fifth packet, the device D_1 realizes that it has consumed all its allowed activity time (36 seconds) so it reached the limit-ToA, the transmission of its following packets is stopped and it enters in a sleep mode until the next cycle when it will have 36 seconds again to finish its transmission. The same thing will happen for the data that should be sent during the second cycle. The image can only be decoded if all its packets are totally received which could be very late. Unlike the first device, the second client D_2 sends its packet and waits for an ACK, after that it enters in a sleep mode because it has finished its transmission during the first cycle.

5.2 Test Scenarios with dynamic activity time sharing

In the two following scenarios (1 and 2) we suppose that there is no registration phase in the join procedure. we treat two cases: first when the donor is available before any request and second when the requester's demands are sent to the server but there are not yet available donors.

Scenario 1: "Donor available before the transmission time limit of the requester" We will test the same previous scenario using activity time sharing without a registration phase. D_1 begins the transmission of its image at t_0 and D_2 begins transmitting its 100 bytes at t_1 . The scenario is shown in Fig.5. The client D_1 informs the server that it is requesting a time of 30.37 seconds in its first frame and D_2 informs the server that is a donor of 33.94 seconds.

As soon as the first requester frame is received, the server starts updating its donor table to satisfy the requester. D_1 receives a favorable response to its request after sending its 4^{th} packet at ζ_2 because at this time D_2 is registered in the donor's table. As a result, D_1 has an additional time to complete its transmission through the activity sharing time mechanism.

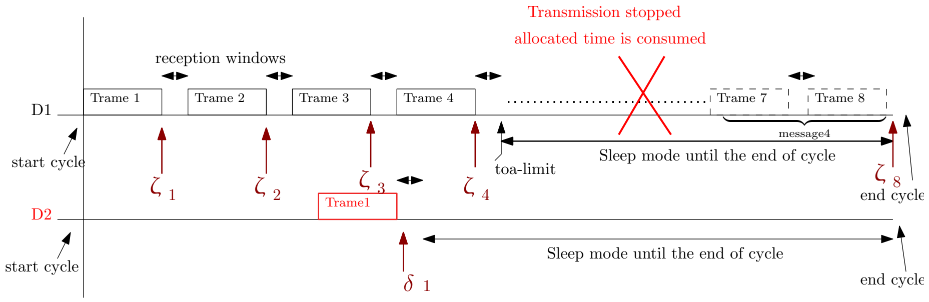


Fig. 4. LoRaWAN behavioral scenario

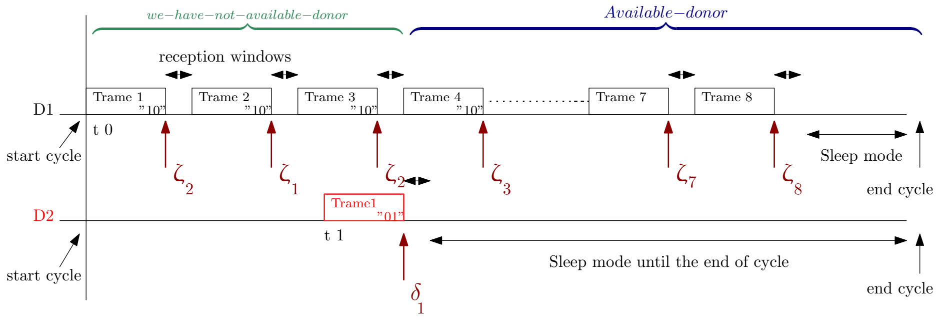


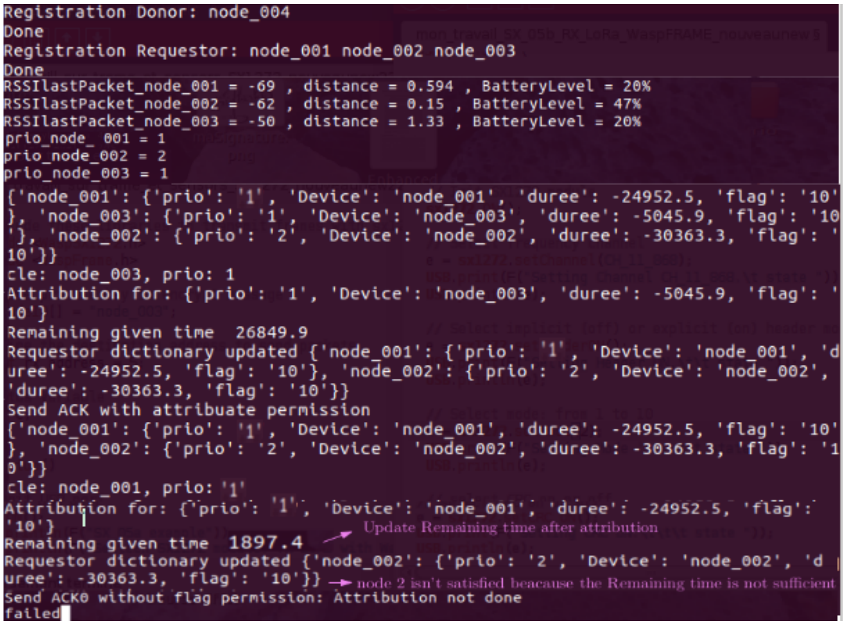
Fig. 5. Scenario with dynamic duty cycle

Scenario 2: “Donor not available before the transmission time limit of the requester” We take the same scenario in Fig.5, but a donor client will not be available before that requester client reaches a limit-ToA, D$_2$ will begin its transmission after that D$_1$ consumes its authorized duty cycle. As a result, the transmission of D$_1$ will be interrupted because the server could not answer the request of D$_1$ (there is no available donors at this time). D$_1$ can not benefit from the dynamic duty cycle for a non-availability of donors before achieving its limit-ToA.
For the following scenarios, we consider a registration phase during the join procedure.

Scenario 3: “Adding of a registration phase: first request before” The registration phase helps to satisfy the requester independently of the beginning of transmission of the donor, contrary to Scenario 2 where we haven’t integrate this registration phase. The requesters will benefit from the shared duty cycle algorithm as long as donors are available in the registration phase.The requester is satisfied by the allocation of additional time.

Scenario 4: “The weakest demand before” In this scenario, we consider a registration phase which each device is registered in the appropriate table. We have 4 devices (3 requesters and 1 donor), the first is dedicated for identification of children at the school, the second one is dedicated for capturing images, the third is dedicated to a stuff’s identification and the last one is dedicated for measurement of temperature (Date, time and temperature). The requesters will transmit at the same time using: Mode 1 for D_2, mode 2 for D_1 and mode 3 for D_3. The three requesters need respectively: 24.95 sec, 30.36 sec and 5 sec. The donor client gives 31.89 sec. We have add a strategy of additional time’s allocation which is ” The weakest demand is satisfied firstly ”. The server will begin by satisfying D$_3$ then we update the given time : 26.84 sec. Then we respond to D$_1$ and we update the given time : 1.89 sec. Arriving to D_2, the server realizes that the given time can not satisfy a D_2's demand. The goal of this strategy is to satisfy a maximum number of requesters, for example, if we have begin with the bigger demand, we will satisfy only D_2. The number of the satisfied requesters will decrease.

Scenario 5: "The lowest battery level before" We resume the same Scenario 4 but with a priority's strategy. D_1, D_2 and D_3 send their registration message as requesters. By receiving the requests for the equipment D_1, D_2 and D_3 during the registration phase, the server sets a priority for each of them according to the level of their batteries and their distance. So, the highest priority is given to the device with the lowest battery level and in case of equality the farthest one (the longest distance). The server will save the information (type of device, requested/given time, distance and the battery level) of each device. If there is a device with the same battery level and distance, we will firstly satisfy the smallest request in order to maximize the number of satisfied clients. The results of the attribution of additional time for Scenario 5 for each device are shown in Fig.7.



```
Registration Donor: node_004
Done
Registration Requestor: node_001 node_002 node_003
Done
RSSIlastPacket_node_001 = -69 , distance = 0.594 , BatteryLevel = 20%
RSSIlastPacket_node_002 = -62 , distance = 0.15 , BatteryLevel = 47%
RSSIlastPacket_node_003 = -50 , distance = 1.33 , BatteryLevel = 20%
prio_node_ 001 = 1
prio_node_002 = 2
prio_node_003 = 1
{'node_001': {'prio': '1', 'Device': 'node_001', 'duree': -24952.5, 'flag': '10'
}, 'node_003': {'prio': '1', 'Device': 'node_003', 'duree': -5045.9, 'flag': '10
'}, 'node_002': {'prio': '2', 'Device': 'node_002', 'duree': -30363.3, 'flag': '
10'}}
cle: node_003, prio: 1
Attribution for: {'prio': '1', 'Device': 'node_003', 'duree': -5045.9, 'flag': '
10'}
Remaining given time  26849.9
Requestor dictionary updated {'node_001': {'prio': '1', 'Device': 'node_001', 'd
uree': -24952.5, 'flag': '10'}, 'node_002': {'prio': '2', 'Device': 'node_002',
'duree': -30363.3, 'flag': '10'}}
Send ACK with attribuate permission
{'node_001': {'prio': '1', 'Device': 'node_001', 'duree': -24952.5, 'flag': '10'
}, 'node_002': {'prio': '2', 'Device': 'node_002', 'duree': -30363.3, 'flag': '1
0'}}
cle: node_001, prio: '1'
Attribution for: {'prio': '1', 'Device': 'node_001', 'duree': -24952.5, 'flag':
'10'}
Remaining given time  1897.4
Requestor dictionary updated {'node_002': {'prio': '2', 'Device': 'node_002', 'd
uree': -30363.3, 'flag': '10'}}
Send ACK0 without flag permission: Attribution not done
failed
```

Update Remaining time after attribution

node 2 isn't satisfied beacause the Remaining time is not sufficient

Fig. 6. Implementation result of Scenario 5 in the server

The devices begin the transmission of their first frames, the server will satisfy the requester having a highest priority then will pass to the requester whose priority is less. In our experimentation, D_1 and D_3 have the same battery level and the same distance, so the server will attribute the highest priority to D_3 because it has the lower request. After serving D_3, the server satisfies D_1. When the remaining given time is updated, the server verifies it and realizes that it's not sufficient compared to the client's demand. D_2 sends its frames but its transmission will be stopped when the limit-ToA is achieved, it will not benefit from the sharing duty cycle.

5.3 Synthesis

In Fig.7 we present the number of devices that have finished their transmission in each scenario for both methods (with and without time activity sharing). We can see that our approach maximizes the number of satisfied clients. For Scenario 1 and 2, we can benefit from the sharing mechanism if the available donors are present before a requester reaches its limited authorized time. In the opposite case we will have a basic behavior of the clients where a requester's transmission stops at 36 seconds. Regarding scenarios 3, 4 and 5, thanks to the configuration phase, we ensure the sharing of duty cycle even if a donor arrives after finishing the transmission of a requester. This means that the sharing do not depend anymore on the beginning of donor's transmission but on its availability in the network during the one-hour cycle (If a donor exists in a network regardless of its t_{start}, the requester can always benefit from the given time provided that it is sufficient to meet the demand of the requesting client). As can be seen, our approach satisfies a larger number of devices by using a additional time management strategy.

As conclusion, our algorithm, in addition to a dynamic duty cycle, allows the devices which have a low battery level (their batteries may be exhausted) and are distant from the LoRa Gateway to benefit from the duty cycle before the others devices. It means we start by the emergency cases (exhausted battery level). This proposition is a combination of the default algorithm, an enhanced duty cycle and a management mechanism of additional time in order to maximize the number of satisfied clients starting with the highest priority. The mechanism occurs normally in the case where the additional time is not enough to make a sharing mechanism, but when the conditions are good and the request can be satisfied the mechanism follows a sharing algorithm with selection of clients. In both cases each donor client register its remaining time in the server in order to allow the requester to benefit from an additional time when this remaining time is sufficient.

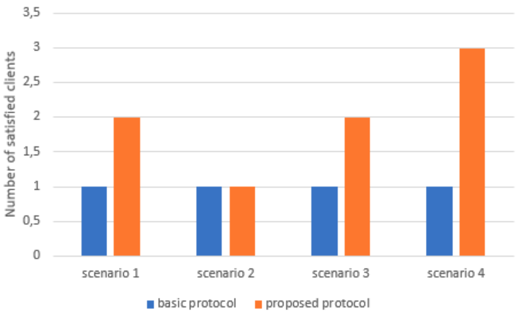


Fig. 7. Number of satisfied clients

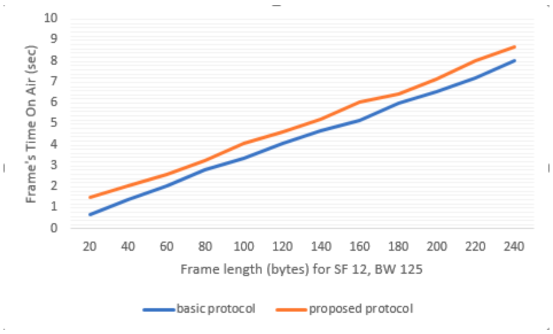


Fig. 8. Join request processing time

Table 4. Join request packet size and processing time

	Join Request Payload	Join Request processing Time
Basic LoRaWAN	18 bytes	1.48 sec
Proposal	24 bytes	1.646 sec

As can be seen in Table 4 and Fig.8, the payload length of the join request message is bounded by an increase of 6 bytes. The join request processing time (basic join request + registration phase) is increased by about 0.166 seconds on average in the proposed approach. This overhead is the same when coded in data frames and is kept scalable for practical use. Also our mechanism did not introduce new messages but was based on the LoRa messages (join, confirmed message).

6 Conclusions

In this paper, we proposed a new algorithm allowing dynamic time sharing among LoRaWAN nodes for allowing flexible node duty cycles while still respecting the total duty cycle limitation of 1% in average. We have improved the previous result in [1] based on the remaining activity time of each device, where the server will allow a client requiring additional time to finish its transmission by borrowing additional time from the unused time left by other clients, thus by adding an allocation management mechanism according on the QoS requirement of each client. This solves the problem of clients that want to exceed their limit-ToA for urgent applications for example. With this algorithm, we can maximize the number of satisfied requesters needing to occasionally exceed their duty cycle limitation. Our solution has been implemented and thoroughly tested. According to the experimental results, we have shown that the proposed sharing mechanism with an appropriate priority-based requester selection improves the overall quality of service. Also our mechanism includes both the sharing in a static mode (registration phase) and in a dynamic situation during the node transmission. Due to the limited available nodes in our lab, our study was limited to a few devices for validating the prove of our concept in real. In our future work, the scalability will be further conducted with a high number of devices. For further enhance the performance, different approaches must be combined. So we also plan to investigate the dynamic LoRa parameter selection and especially the adaptive data rate (ADR) and its combination with the activity time sharing approach.

References

[1] Pham, CongDuc.: QoS for long-range wirless sensors under duty-cycle regulation with shared activity time usage. TOSN 12 1-33 (Septembre 2016).

[2] Luigi, A., Antonio, I., Giacomo M.: The Internet of Things: A survey. Computer Networks 54, 27872805 (June 2010)

[3] L, Labs.: A Comprehensive Look At Low Power, Wide Area Networks For Internet of Things Engineers and Decision Makers, (2016).

[4] X. Xiong, K.Zheng, R. Xu, W. Xiang, and P. Chatzimisios.: Low Power Wide Area Machine-to-Machine Networks: Key Techniques and Prototype, IEEE Communications Magazine, vol. 53, no. 9, 64-71. (Sep. 2015).

[5] Rashmi Sharan Sinha, Yiqiao, W., Seung-Hoon, H.: A survey on LPWA technology: LoRa and NB-IoT. Volume 3. Pages 14-21. (March 2017).

[6] LoRa LoRaWAN Primer. Source : https://www.leverege.com/research-papers/lora-lorawan-primer.

[7] Sikken, B.: Project DecodingLoRa. 2016.

[8] Springer, A., Gugler, W., Huemer, M., Reind, L., Ruppel, C., Weigel, R.: Spread spectrum communications using chirp signals. Proceedings of the IEEE/AFCEA. Munich, Germany. 166-170 (19 May 2000).

[9] LoRa Alliance Technical committee LoRawan regional parameters, Revision 1.0. (July 2016).

[10] Martin, B., Utz, R.: LoRa Transmission parameter selection. 13th International Conference DCOSS, Ottawa, ON, Canada. (June 2017).

[11] Yun Seong J., Muhammad Rehan, U., Muhammad Arslan, U., and Soo Young, S.: Swapped Huffman Tree coding application for Low Power Wide Area Network (LPWAN). International Conference ICSGTEIS. (Octobre 2016).

[12] nkeWatteco,: LoRaWANTM PressO. V2.0 Updated 04/09/2016.

[13] Sornin, N (Semtech)., Luis, M (Semtech)., Eirich, T (IBM)., Kramp, T (IBM)., Hersent, O (Actility).: LoRa specification 1.0 LoRa alliance standard specification., (january 2015).

[14] Semtech corporation SX1272/3/6/7/8: LoRa modem an1200.13, Revision 1. (July 2017).

[15] Libelium comunicaciones Distribuidas S.L.: Waspmote LoRa 868 MHz 915 MHz SX1272 networking guide, Revision 4.2. (November 2015).

[16] Libelium Comunicaciones Distribuidas S.L.: Waspmote data frame programming guide. version 7.2. (October 2017).

[17] Norbert, B., Fernando, K.: LoRaWAN in the Wild: Measurements from The Things Network. ArXiv2017. (June 2017).