



HAL
open science

Optimality and Approximation Ratios of Demodulator Allocation Strategies in LoRa Multi-Gateway Networks

Alexandre Guitton, Megumi Kaneko, Nancy El Rachkidy

► **To cite this version:**

Alexandre Guitton, Megumi Kaneko, Nancy El Rachkidy. Optimality and Approximation Ratios of Demodulator Allocation Strategies in LoRa Multi-Gateway Networks. 2023. hal-04162716

HAL Id: hal-04162716

<https://hal.science/hal-04162716>

Preprint submitted on 16 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Optimality and Approximation Ratios of Demodulator Allocation Strategies in LoRa Multi-Gateway Networks

Alexandre Guitton^{*†}, Megumi Kaneko[‡], Nancy El Rachkidy^{*}

^{*}Université Clermont-Auvergne, CNRS, Mines de Saint-Etienne, Clermont-Auvergne-INP, LIMOS

[†]Univ Lyon, INSA Lyon, Inria, CITI

[‡]Information Systems Architecture Science Research Division, National Institute of Informatics

Abstract—Long Range (LoRa) is a wireless modulation used to provide long-range connectivity to low-power devices of the Internet of Things (IoT). LoRa gateways use hardware chips called demodulators in order to demodulate concurrent frames on different channels or spreading factors. In this paper, we establish several theoretical results on the performance of fundamental demodulator allocating strategies. In the single gateway scenario, we first prove the optimality of the basic preemptive allocation strategy. Next, in the multi-gateway scenario, the greedy non-collaborative strategy, which is currently implemented in LoRa gateways, is shown to achieve a very large approximation ratio. As an alternative, we demonstrate that a preemptive non-collaborative strategy has an approximation ratio of 2, and furthermore, that a preemptive smart-collaborative strategy reaches an approximation ratio of 3/2, for two gateways. These seminal results imply that the current performance of multi-gateway LoRa deployments can be largely improved by the proposed collaborative demodulator allocation strategies.

I. INTRODUCTION

Low-Power Wide Area Networks (LPWANs) are networks where low-power end-devices can communicate wirelessly over long distances. LPWANs are largely used in the Internet of Things (IoT), thanks to their long-range capabilities in environmental monitoring applications, or to their low-cost and scalability in smart cities. Long Range (LoRa) is one of the main technologies for LPWANs: it is a very robust wireless modulation that can achieve a communication range of tens of kilometers outdoor. LoRa is often used in combination with Long Range Wide Area Network (LoRaWAN), which defines a simple medium access control protocol and network topology on top of LoRa.

In LoRaWAN, the end-devices communicate to a network server through gateways, as shown in Figure 1. Each gateway in range of an end-device might intercept its frames, decode them, and forward them to the network server. The network server is in charge of removing frame duplicates. The gateways have the ability to decode several incoming transmissions at the same time, provided that they are sent on different channels or that they use different spreading factors (SFs). To do so, each gateway listens for preambles on all channels and SFs at all time, and allocates a demodulator to each incoming frame. The total number of demodulators per gateway is limited, typically to eight [1].

Several researchers have shown that the limited number of demodulators reduces the throughput of LoRa [2], [3], [4]. Thus, there have been a few attempts to allocate the demodulators efficiently [5], [6], [7]. This is a challenging issue, as the demodulator allocation is an online scheduling problem with specific constraints.

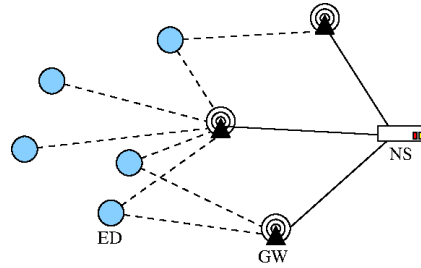


Figure 1. End-devices (EDs) communicate to all gateways (GWs) in range through LoRa links (represented with dashed lines). The gateways are all connected to the network server (NS) through an IP backhaul (represented with solid lines).

In this paper, we investigate the theoretical performance of demodulator allocating strategies, in terms of the number of demodulated frames. We demonstrate the following results, which are also summarized in Table I:

- In the single gateway scenario, with a single demodulator, the approximation ratio of the greedy algorithm is equal to $f_{\max} + 1 = 72$, where f_{\max} is the maximum number of short frames (i.e., with SF7) that can fit during a single long frame (i.e., with SF12). Meanwhile, a simple preemptive algorithm is optimal.
- In the single gateway scenario with $D = 2$ demodulators, the simple preemptive algorithm is still optimal.
- In the multi-gateway scenario with $M \geq 2$ gateways, the approximation ratio of the non-collaborative greedy algorithm is at least Mf_{\max} , and at least f_{\max} with simple collaboration. Meanwhile, the preemptive algorithm achieves an approximation ratio of 2, even without collaboration, for $M = 2$ gateways. By using a smart collaboration, the preemptive algorithm reaches an approximation ratio of 1.5 for $M = 2$ gateways.

Finally, these theoretical findings have been validated and illustrated through extensive computer simulation evaluations.

Table I
SUMMARY OF OUR THEORETICAL RESULTS.

Number of gateways	Number of demodulators	Greedy		Preemptive		
		non-collaborative	simple collaboration	non-collaborative	simple collaboration	smart collaboration
1	1	$= f_{\max} + 1$ (Theorem 1)		$= 1$ (Theorem 2)		
1	$D \geq 2$	$\geq f_{\max} + 1$ (corollary of Theorem 1)		$= 1$ for $D = 2$ (Theorem 3)		
$M \geq 2$	1	$\geq M \times (f_{\max} + 1)$ (corollary of Theorem 1)		$= 2$ for $M = 2$ (Theorem 4)	$= 2$ for $M = 2$ (Theorem 5)	≤ 1.5 for $M = 2$ (Theorem 6)

The remainder of this paper is organized as follows. Section II presents the existing works on demodulator allocation strategies, as well as similar scheduling problems. Section III presents our system model. Section IV presents the four algorithms we consider, as well as an optimal solution using a mixed integer linear program (MILP). Section V focuses on the single gateway scenario, with either one or two demodulators for the gateway. Section VI focuses on the multi-gateway scenario, with a single demodulator per gateway. Section VII-B describes our simulation results. Finally, Section VIII concludes our work.

II. STATE-OF-THE-ART

Several researchers have shown that the limited number of demodulators in LoRa gateways reduces the overall network performance. In [2], the authors included the number of demodulators as a parameter of their mathematical model of LoRaWAN, and showed that this parameter has a significant impact on the throughput. In [3], the authors also considered the limited number of demodulators as a practical hardware limitation. In [4], the authors studied the impact of several parameters, and notably the number of parallel reception paths which is equal to the number of demodulators, on the performance of LoRaWAN.

A. Related work on demodulator allocation strategies

Few research works attempted to improve the usage of demodulators, probably due to the fact that the strategy implemented in the firmware of existing LoRaWAN gateways is not documented.

In [5], the authors described the conventional first-in-first-out (FIFO) greedy strategy, as well as two improved strategies called FIFO-RR1 and FIFO-RR2, for a single gateway scenario. FIFO-RR1 and FIFO-RR2 make use of the unused time between the detection of the preamble and the beginning of the payload in order to demodulate short frames. The FIFO-RR2 strategy can also plan in advance the demodulation of a long frame before the end of the demodulation of another frame.

In [6], the authors proposed two strategies for a multi-gateway scenario: an explicit collaboration strategy, and a random strategy. The explicit collaboration strategy enables a gateway to ignore the demodulation of frames that are planned to be demodulated by another gateway. This is achieved through control messages, but requires long SFs. The random strategy attempts to make gateways randomly choose different frames to demodulate, and is intended for small SFs.

In [7], an auction-based optimization approach was designed in order to pre-assign each IoT device to a unique gateway, so as to maximize the total amount of demodulated transmissions without redundancy at the network server, given the constraint of the limited number of demodulators per gateway. This optimization method was shown to significantly outperform benchmark algorithms, and to be suitable for mobile IoT applications.

B. Similar scheduling problems

The demodulator allocation can be seen as a special form of online scheduling problem with strict deadlines, where each task is an incoming frame, and each gateway can process up to eight tasks at the same time, given that existing LoRaWAN gateways are equipped with only eight demodulators [1]. Tasks have varying durations (depending on the SF and on the payload length), and are released at the time when the preamble is detected by the gateway.

Compared to usual scheduling problems, the main specificity of the demodulator allocation problem is that the processing of our tasks cannot be postponed. Indeed, our tasks have to be processed exactly from the preamble start to the preamble end. To the best of our knowledge, this specificity has not been studied deeply by the research community on scheduling problems.

However, some results from the literature are still related to ours. For instance, in [8], the authors prove that the problem is NP-hard on a single processor (that is, with a single gateway). However, they show that it becomes polynomial if the task have to be scheduled upon arrival and if there is no preemption, which is similar to a case we consider. In [9], the author proves the NP-hardness of the problem when tasks have deadlines (in addition to due dates). Finally, our problem is also related to soft real-time scheduling [10], where the real-time comes from the fact that tasks cannot be postponed, and the softness comes from the fact that some tasks are allowed to be dropped. Unfortunately, we did not find algorithms that can solve our problem, nor fundamental results on the performance of such algorithms.

III. SYSTEM MODEL AND NOTATIONS

According to the specifications of the hardware component of LoRaWAN gateways (that is, either the SX1301 or the SX1302 chip) [1], we consider that each gateway listens for preambles on all channels and spreading factors, at all times. Each time a new frame is detected (which occurs during the preamble of the frame), the gateway decides

whether it allocates a demodulator for this frame or not. To be successfully demodulated, at least one gateway has to demodulate the whole payload of a frame, without interruption nor demodulator change.

In order to simplify the proofs, we take a number of assumptions as follows:

- different frames cannot start or finish at exactly the same time, which corresponds to the reality.
- in the algorithms, the preamble detection time is equal to the payload starting time (although in practice, the preamble is detected at about half of the preamble duration [11], [12]).
- there is no loss of payload for a frame whose preamble was detected, hence a frame is successfully decoded as long as at least one gateway has allocated a demodulator during the whole frame payload.
- for collaborative strategies, we consider that two gateways can determine whether two preambles received by two gateways correspond to the same frame or not¹.

Notations: Let \mathcal{F} be a set of frames. The payload of each frame $f \in \mathcal{F}$ starts at $t_f^{(s)}$ and ends at $t_f^{(e)}$. Given an algorithm A , let us denote by $\Gamma_t(A)$ the total number of frames that have been demodulated by A until time t (included). Note that if t is omitted, it is assumed to be equal to $\max_{f \in \mathcal{F}} t_f^{(e)}$. Finally, let $\Gamma_{t',t}(A)$ be the number of frames successfully demodulated by an algorithm A and whose ending times are within $]t'; t]$.

In the following, we will show that some algorithms are optimal in some conditions.

Definition: An algorithm OPT is said to be optimal if, for all sets of frames \mathcal{F} and for all algorithms A , we have

$$\Gamma(OPT(\mathcal{F})) \geq \Gamma(A(\mathcal{F})). \quad (1)$$

We will also compute the approximation ratio of some algorithms in some conditions.

Definition: An algorithm A is said to have an approximation ratio $r \geq 1$ for a maximization problem if for all sets of frames \mathcal{F} , we have

$$\frac{\Gamma(OPT(\mathcal{F}))}{\Gamma(A(\mathcal{F}))} \leq r. \quad (2)$$

This means that the solution obtained by A for any set of frames \mathcal{F} cannot be r times lower than the optimal solution. This notion is very important as it gives a worst bound on the quality of the solution computed by A , compared to the optimal solution.

IV. ALGORITHMS

In the following, we will consider the following algorithms:

- G (see Algorithm 1) is a greedy algorithm: demodulators are allocated using a FIFO strategy. Algorithm G models the method implemented in current LoRaWAN gateways.

¹We believe this is possible by identifying the channel, spreading factor, preamble detection time, and Carrier Frequency Offset (CFO). Notably, each gateway can estimate the CFO of each node through a training period, and use this CFO later to help determine whether two preambles received by two gateways correspond to the same frame. A more detailed discussion on frame identification is in [7].

- P (see Algorithm 2) is a greedy algorithm with preemption: a demodulator can drop its current frame c in order to demodulate f , if f ends earlier than c .
- P^C (see Algorithm 3) is a greedy algorithm with preemption, which takes into account a simple form of collaboration among gateways (through the network sever): each frame is demodulated by a single gateway (chosen arbitrarily).
- P^S (see Algorithm 4) is a greedy algorithm with preemption, which takes into account a more complex form of collaboration among gateways: when a frame is demodulated by several gateways, these gateways (except one) can later drop the common frame to accept a new frame.
- OPT denotes the optimal algorithm, namely the one with maximal gain Γ .

Algorithm 1 Algorithm G

Upon detecting the preamble of a new frame f :
if a demodulator is idle **then**
 demodulate f
end if

Algorithm 2 Algorithm P

Upon detecting the preamble of a new frame f :
if a demodulator is idle **then**
 demodulate f
else
 if there exists a demodulator with a frame c such that $t_f^{(e)} < t_c^{(e)}$ **then**
 $\delta \leftarrow$ the demodulator demodulating the frame c with the largest $t_c^{(e)}$
 make demodulator δ drop its frame
 demodulate f (with δ)
 end if
end if

Algorithm 3 Algorithm P^C

Upon detecting the preamble of a new frame f :
 apply algorithm P
 if several gateways demodulate f **then**
 drop f for all but one gateway (chosen arbitrarily)
 end if

The optimal algorithm is defined as a mixed-integer linear programming (MILP) problem, as follows. The inputs are the number of frames F , the number of demodulators D , the number of gateways M , an array Col such that $Col[f_1, f_2]$ is true if and only if frames f_1 and f_2 overlap in time (i.e., there is a collision between f_1 and f_2), and an array Rx such that $Rx[f, g]$ is true if and only if frame f is received by gateway g . Note that \mathcal{F} is modeled indirectly through the array Col .

The MILP uses three arrays of binary variables: $Dem[g, d, f]$ is true if and only if the d -th demodulator of gateway g demodulates frame f , $Dem_{1d}[g, f]$ is true if and only if at least one demodulator of gateway g demodulates

Algorithm 4 Algorithm P^S

Upon detecting the preamble of a new frame f :
if a demodulator is idle **then**
 demodulate f
else
 if a demodulator is demodulating a frame c which is also demodulated by at least another gateway **then**
 drop c
 demodulate f
 else
 if there exists a demodulator with a frame c such that $t_f^{(e)} < t_c^{(e)}$ **then**
 $\delta \leftarrow$ the demodulator demodulating the frame c with the largest $t_c^{(e)}$
 make demodulator δ drop its frame
 demodulate f (with δ)
 end if
 end if
end if

frame f , and $Dem_{1g}[f]$ is true if and only if at least one gateway demodulates frame f .

The MILP is defined by the following equations. Equation (3) is the objective function: the goal of OPT is to maximize the number of frames demodulated by any gateway. Equation (4) states the constraint that a demodulator d of a gateway g cannot demodulate a frame f if f is not received by gateway g . Equation (5) constrains that it is not possible for two frames f_1 and f_2 to be demodulated by the same demodulator d if $Col[f_1, f_2]$ is true, that is if f_1 and f_2 overlap in time. Indeed, if f_1 is not chosen by d , then the right part of the equation is equal to F , which does not impose any constraint on the left part of the equation. However, if f_1 is chosen by d (and thus, $Dem[g, d, f_1] = 1$), all other frames f_2 chosen by d have to be such that $Col[f_1, f_2] = 0$. Equations (6) and (7) ensure that $Dem_{1d}[g, f]$ is true when at least one demodulator of g demodulates f , based on the value of Dem . Equations (8) and (9) ensure that $Dem_{1g}[f]$ is true when at least one gateway demodulates f , depending on the value of Dem_{1d} .

$$\begin{aligned} & \max \sum_{f \in [1; F]} Dem_{1g}[f] \\ \text{s.t. } & Dem[g, d, f] \leq Rx[f, g], \quad \forall g \in [1; M], d \in [1; D], f \in [1; F] \quad (4) \\ & \sum_{f_2 \neq f_1} Dem[g, d, f_2] Col[f_1, f_2] \leq (1 - Dem[g, d, f_1])F, \\ & \quad \forall g \in [1; M], d \in [1; D], f_1 \in [1; F] \quad (5) \\ & Dem_{1d}[g, f] \geq Dem[g, d, f], \quad \forall g \in [1; M], d \in [1; D], f \in [1; F] \quad (6) \\ & Dem_{1d}[g, f] \leq \sum_{d \in [1; D]} Dem[g, d, f], \quad \forall g \in [1; M], f \in [1; F] \quad (7) \\ & Dem_{1g}[f] \geq Dem_{1d}[g, f], \quad \forall g \in [1; M], f \in [1; F] \quad (8) \\ & Dem_{1g}[f] \leq \sum_{g \in [1; M]} Dem_{1d}[g, f], \quad \forall f \in [1; F]. \quad (9) \end{aligned}$$

V. SINGLE GATEWAY SCENARIO

In this section, we consider a single gateway. We will show three results: algorithm G has a large approximation ratio for $D = 1$ demodulator (see Theorem 1), P is optimal for $D = 1$

demodulator (see Theorem 2), and P is also optimal for $D = 2$ demodulators (see Theorem 3).

Theorem 1 (Approximation ratio of Algorithm G for $D = 1$).
For all \mathcal{F} , $\frac{\Gamma(OPT(\mathcal{F}))}{\Gamma(G(\mathcal{F}))} \leq f_{\max} + 1$ for $D = 1$, with f_{\max} the maximum number of short frames that can fit in a single long frame. This bound is tight.

Proof. Let \mathcal{F} be any set of frames, and let us compare the solution found by algorithm G with the optimal solution found by OPT . Let us denote by n_1 the number of frames of OPT that start while G is not demodulating, and by n_2 the number of frames of OPT that start while G is demodulating. Let us further refine n_2 by having n_3 denote the number of frames of OPT that start and finish while G is demodulating the same frame, and n_4 denote the other frames of n_2 . If we denote by f_{\max} the maximum number of short frames that can fit during a single long frame (with the assumption that two frames cannot start exactly at the same time), then for each frame of G , there can be at most f_{\max} frames in OPT . In other words, $n_3 \leq f_{\max} \Gamma(G(\mathcal{F}))$. Moreover, $n_4 \leq \Gamma(G(\mathcal{F}))$ since each frame counted in n_4 corresponds to at most one starting frame in G . Finally, $n_1 = 0$, as it is not possible for OPT to choose a frame while G is not demodulating, since G is a greedy algorithm. Thus, we have $\Gamma(OPT(\mathcal{F})) = n_1 + n_2 = n_1 + n_3 + n_4 \leq (f_{\max} + 1) \Gamma(G(\mathcal{F}))$, and consequently $\frac{\Gamma(OPT(\mathcal{F}))}{\Gamma(G(\mathcal{F}))} \leq f_{\max} + 1$.

This bound is tight, and can be obtained with the following set of frames \mathcal{F} : an SF12-frame starts first, and then $f_{\max} + 1$ SF7-frames follow shortly after. Note that the $(f_{\max} + 1)$ -th SF7-frame has to start before the single SF12-frame finishes. Then, G selects the single SF12-frame (resulting into $\Gamma(G(\mathcal{F})) = 1$), and OPT selects the $(f_{\max} + 1)$ SF7-frames (resulting into $\Gamma(OPT(\mathcal{F})) = f_{\max} + 1$). \square

Note that $f_{\max} = 71$ with the European LoRaWAN regional settings. Indeed, the payload duration of a short frame (payload of 10 bytes with SF7) is 28.68 ms, and the payload duration of a long frame (payload of 51 bytes with SF12) is 2064.38 ms, as shown on Table II. In both cases, we considered a coding rate of 4/5 and a bandwidth of 125 kHz.

Table II
PAYLOAD DURATION OF LORA FRAMES AS A FUNCTION OF THE PAYLOAD SIZE.

SF	Largest payload (bytes)	Payload duration	
		(10 bytes)	(largest payload)
7(5)	242	28.68 ms	366.60 ms
8(6)	242	47.10 ms	641.02 ms
9(6)	115	94.20 ms	565.24 ms
10(7)	51	188.42 ms	516.10 ms
11	51	376.84 ms	1114.12 ms
12(8)	51	589.82 ms	2064.38 ms

(9)

Theorem 2. Algorithm P is optimal for $D = 1$.

Lemma 1. Let \mathcal{F} be a set of frames and A be an algorithm. Let t be a time such that $\Gamma_t(OPT(\mathcal{F})) > \Gamma_t(A(\mathcal{F}))$, and let $t' < t$. If $\Gamma_{t',t}(OPT(\mathcal{F})) \leq \Gamma_{t',t}(A(\mathcal{F}))$, then t' is a time where $\Gamma_{t'}(OPT(\mathcal{F})) > \Gamma_{t'}(A(\mathcal{F}))$.

Proof. By assumption, we have $\Gamma_t(OPT(\mathcal{F})) \geq \Gamma_t(A(\mathcal{F})) + 1$. We also have $\Gamma_t(OPT(\mathcal{F})) = \Gamma_{t'}(OPT(\mathcal{F})) + \Gamma_{t',t}(OPT(\mathcal{F}))$ by definition of $\Gamma_{t',t}$. Then, we have $\Gamma_{t'}(OPT(\mathcal{F})) = \Gamma_t(OPT(\mathcal{F})) - \Gamma_{t',t}(OPT(\mathcal{F})) \geq \Gamma_t(OPT(\mathcal{F})) - \Gamma_{t',t}(A(\mathcal{F})) \geq \Gamma_t(A(\mathcal{F})) + 1 - \Gamma_{t',t}(A(\mathcal{F})) = \Gamma_{t'}(A(\mathcal{F})) + 1$, with $t' < t$. Thus, $\Gamma_{t'}(OPT(\mathcal{F})) > \Gamma_{t'}(A(\mathcal{F}))$. \square

Proof of Theorem 2. Let us reason by contradiction, and assume that for a given \mathcal{F} , there is a first time t such that $\Gamma_t(OPT(\mathcal{F})) > \Gamma_t(P(\mathcal{F}))$. This means that at t , OPT finished demodulating a frame x that was not chosen by P . Either P was busy demodulating another frame y when x started, or P also chose x but dropped it later by preemption for another frame y . Note that since P is a greedy algorithm, it is not possible for P to have an available demodulator and not choose x . In both cases, y is a frame ending before $t_x^{(e)}$. Let $t' = t_x^{(s)} < t$. We have $\Gamma_{t',t}(OPT(\mathcal{F})) = 1$ since $D = 1$ and OPT demodulates only x during $]t'; t]$. We also have $\Gamma_{t',t}(A(\mathcal{F})) \geq 1$ since P demodulates at least y during $]t'; t]$. Thus, Lemma 1 holds, which brings a contradiction since t is the first time such that $\Gamma_t(OPT(\mathcal{F})) > \Gamma_t(P(\mathcal{F}))$. Consequently, there is no time t where $\Gamma_t(OPT(\mathcal{F})) > \Gamma_t(P(\mathcal{F}))$, which means that P is optimal. \square

Theorem 3. *Algorithm P is optimal for $D = 2$.*

Proof. Let t be the first time where $\Gamma_t(OPT(\mathcal{F})) > \Gamma_t(P(\mathcal{F}))$. Let x be the last demodulated frame of OPT , and x' be the one before².

If x was chosen by P , then let $t' = t_x^{(e)}$. In this case, $\Gamma_{t',t}(OPT(\mathcal{F})) = 1$ (corresponding to frame x , since x' is the last demodulated frame before x) and $\Gamma_{t',t}(P(\mathcal{F})) \geq 1$ (corresponding to x and possibly others). Thus, Lemma 1 holds and we have a contradiction with the definition of t . Note that this requires P to apply the best preemption, that is to preempt with the demodulator having the frame finishing the latest. Indeed, a badly-designed preemptive algorithm could select a very long frame y on one demodulator (with $t_y^{(e)} > t$), choose x on the second demodulator, then drop x (instead of y) to preempt frame x' , as shown on Figure 2. This badly-designed preemptive algorithm would not have finished demodulating y at t , and thus would be such that $\Gamma_{t',t}(P(\mathcal{F})) = 0$.

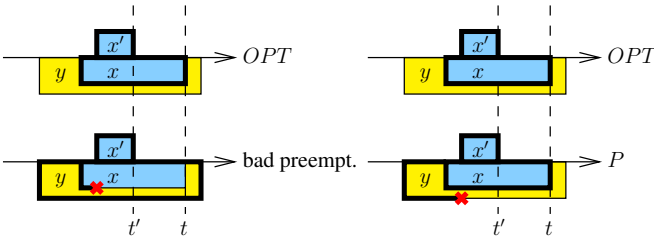


Figure 2. Comparison of a badly-designed preemptive algorithm (left) and P (right) in the proof of Theorem 3. Frames are depicted with colored rectangles. Demodulated frames are highlighted with thick edges.

²Note that if $\Gamma_t(OPT(\mathcal{F})) > \Gamma_t(P(\mathcal{F}))$, then there are at least two frames demodulated by OPT . If there was only one frame demodulated by OPT , P would have chosen this short frame (possibly by preemption).

If x was not chosen by P , let us consider the following cases.

- Case 1: x' was chosen by P and $t_x^{(s)} < t_x^{(s)}$, as shown on the left of Figure 3. Let $t' = t_x^{(s)}$. $\Gamma_{t',t}(OPT(\mathcal{F})) \leq 2$ (one frame is x , and the other is x' if and only if $t_x^{(s)} < t_{x'}^{(e)}$). At t' , both demodulators of P were busy with frames finishing before $t_x^{(e)}$ (one of them possibly being x') since x was not chosen, even by preemption. Thus, $\Gamma_{t',t}(P(\mathcal{F})) \geq 2$ (corresponding to these two frames). Thus, Lemma 1 holds.

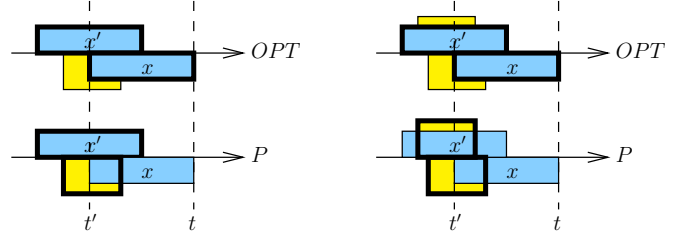


Figure 3. Illustration of Case 1 (left) and Case 3 (right) of the proof of Theorem 3.

- Case 2: x' was chosen by P and $t_x^{(s)} < t_{x'}^{(s)}$, as shown on Figure 4. Let $t' = t_x^{(s)}$. Since x was not chosen by P at t' , the two demodulators of P were busy demodulating frames whose ending time are before $t_x^{(e)}$: let us denote them y_1 and y_2 . Let us denote by n the number of frames demodulated by OPT between $t_x^{(s)}$ and $t_{x'}^{(s)}$, and let us call them x_1, \dots, x_n . For each frame x_i , either x_i is also demodulated by P , or P is busy demodulating a frame starting before $t_{x_i}^{(s)}$ (due to the greediness of P) and finishing before $t_{x_i}^{(e)}$ (due to the preemptiveness of P)³. Thus, during $]t'; t]$, P demodulates y_1 and y_2 , as well as x' and at least $n - 1$ other frames corresponding to each x_i except x_1 . Overall, $\Gamma_{t',t}(OPT(\mathcal{F})) = n + 2$ (including all the x_i frames, as well as x and x'), and $\Gamma_{t',t}(P(\mathcal{F})) = (n - 1) + 3 = n + 2$ (including x' , y_1 and y_2). Thus, Lemma 1 holds.

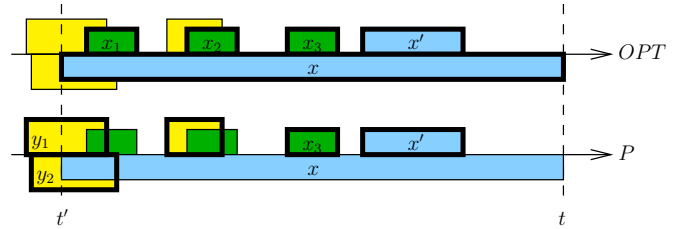


Figure 4. Illustration of Case 2 of the proof of Theorem 3.

- Case 3: x' was not chosen by P , as shown on the right side of Figure 3. Let $t' = \max(t_x^{(s)}, t_{x'}^{(s)})$, $\Gamma_{t',t}(OPT(\mathcal{F})) \leq 2$. Since neither x nor x' were chosen by P , it means that when x and x' started, the two demodulators of P were used. Thus, at t' , the two demodulators are busy with frames finishing before the

³For x_1 , this frame could be y_1 or y_2 .

end of the frame starting at t' . Since $\max(t_x^{(e)}, t_{x'}^{(e)}) \leq t$, we have $\Gamma_{t',t}(P(\mathcal{F})) \geq 2$, and Lemma 1 holds.

In all cases, we have a contradiction with the definition of t as the earliest time where $\Gamma_t(OPT(\mathcal{F})) > \Gamma_t(P(\mathcal{F}))$. In other words, there is no such time t , and thus P is optimal. \square

VI. MULTI-GATEWAY SCENARIO

In this section, we consider $M \geq 2$ gateways. We will show three results: the non-collaborative algorithm P has an approximation ratio of 2 for $D = 1$ demodulator (see Theorem 4), the collaborative algorithm P^C also has an approximation ratio of 2 for $D = 1$ demodulator (see Theorem 5), and the collaborative algorithm P^S has an approximation ratio of 1.5 for $D = 1$ demodulator (see Theorem 6).

Theorem 4 (Approximation ratio of algorithm P for $M = 2$ and $D = 1$). *For all \mathcal{F} , $\frac{\Gamma(OPT(\mathcal{F}))}{\Gamma(P(\mathcal{F}))} \leq 2$ for $M = 2$ and $D = 1$. This bound is tight.*

Proof. Let \mathcal{F} be an arbitrary set of frames. From $P(\mathcal{F})$, we will build the solution $OPT(\mathcal{F})$ with the largest gain, and show that $\frac{\Gamma(OPT(\mathcal{F}))}{\Gamma(P(\mathcal{F}))} \leq 2$. Let us divide the time into intervals depending on $P(\mathcal{F})$: a new interval δ_i starts each time P finishes demodulating a new frame on any gateway, for $i \in [1; F]$. Let $t_{\delta_i}^{(s)}$ be the starting time of δ_i , and $t_{\delta_i}^{(e)}$ the ending time of δ_i . Let us denote by $\Gamma_{\delta_i}(A(\mathcal{F}))$ the number of frames that finish their demodulation within $]t_{\delta_i}^{(s)}; t_{\delta_i}^{(e)}]$ for algorithm A . By definition of δ_i , we have $\Gamma_{\delta_i}(P(\mathcal{F})) = 1$ (for all gateways), for every δ_i . This equality to one is obtained thanks to the assumption that two different frames cannot start or finish at exactly the same time, as stated in Section III.

Let us introduce the following three notations concerning $OPT(\mathcal{F})$:

- 1) $n_{\delta_i}^{(s_{bef})}$ is the number of frames that started strictly before $t_{\delta_i}^{(s)}$ and are demodulated within $]t_{\delta_i}^{(s)}; t_{\delta_i}^{(e)}]$,
- 2) $n_{\delta_i}^{(e_{dur})}$ is the number of frames that started and ended within $]t_{\delta_i}^{(s)}; t_{\delta_i}^{(e)}]$,
- 3) $n_{\delta_i}^{(e_{aft})}$ is the number of frames that started before or at $t_{\delta_i}^{(e)}$ and that will end strictly after $t_{\delta_i}^{(e)}$.

Let us consider any interval δ_i , for $i \in [1; F]$. For each of the following cases, let us show that $n_{\delta_i}^{(e_{dur})} + n_{\delta_i}^{(e_{aft})} \leq 2\Gamma_{\delta_i}(P(\mathcal{F})) = 2$.

- Case 1: both gateways demodulate the same frame f . This means that all other frames f' starting during δ_i are such that $t_{f'}^{(e)} > t_f^{(e)}$. If f is chosen by OPT , we have $n_{\delta_i}^{(e_{dur})} = 1$ and $n_{\delta_i}^{(e_{aft})} \leq 1$. Otherwise, we have $n_{\delta_i}^{(e_{dur})} = 0$ and $n_{\delta_i}^{(e_{aft})} \leq 2$.
- Case 2: the frame f finishing in P is demodulated by GW_1 , but not by GW_2 . If OPT chooses f on GW_1 , then $n_{\delta_i}^{(e_{dur})} = 1$ and $n_{\delta_i}^{(e_{aft})} \leq 1$ for GW_2 . If OPT does not choose f on GW_1 , then there can be another frame f' starting on GW_1 . For this frame, $t_{f'}^{(e)} > t_f^{(e)}$ otherwise P would have chosen f' by preemption. There might also be another frame f'' starting on GW_2 , but then $t_{f''}^{(e)} > t_f^{(e)}$, otherwise either P would have chosen f'' , or P would

have dropped its potential frame on GW_2 for this one. Thus, $n_{\delta_i}^{(e_{dur})} = 0$ and $n_{\delta_i}^{(e_{aft})} \leq 2$.

- Case 3: the frame f finishing in P is demodulated by GW_2 , but not by GW_1 . This is similar to Case 2.
- Case 4: no more frames are demodulated during δ_i , which occurs only for $i = n$. This means that no frame can start for OPT from $t_{\delta_n}^{(s)}$. Thus, $n_{\delta_n}^{(e_{dur})} = 0$ and $n_{\delta_n}^{(e_{aft})} = 0$.

In all cases, we have $n_{\delta_i}^{(e_{dur})} + n_{\delta_i}^{(e_{aft})} \leq 2 = 2\Gamma_{\delta_i}(P(\mathcal{F}))$.

Finally, we have:

$$\begin{aligned} \Gamma(OPT(\mathcal{F})) &= \sum_{\delta_i, i \in [1; F]} \Gamma_{\delta_i}(OPT(\mathcal{F})) \\ &= \sum_{\delta_i, i \in [1; F]} \left(n_{\delta_i}^{(s_{bef})} + n_{\delta_i}^{(e_{dur})} \right) \\ &= \sum_{\delta_i, i \in [2; F]} n_{\delta_i}^{(s_{bef})} + \sum_{\delta_i, i \in [1; F]} n_{\delta_i}^{(e_{dur})} \\ &\leq \sum_{\delta_i, i \in [1; F-1]} n_{\delta_i}^{(e_{aft})} + \sum_{\delta_i, i \in [1; F]} n_{\delta_i}^{(e_{dur})} \\ &\leq \sum_{\delta_i, i \in [1; F]} \left(n_{\delta_i}^{(e_{dur})} + n_{\delta_i}^{(e_{aft})} \right) \\ &\leq \sum_{\delta_i, i \in [1; F]} 2\Gamma_{\delta_i}(P(\mathcal{F})) = 2\Gamma(P(\mathcal{F})), \end{aligned}$$

since $n_{\delta_1}^{(s_{bef})} = 0$, $n_{\delta_{i+1}}^{(s_{bef})} \leq n_{\delta_i}^{(e_{aft})}$ and $n_{\delta_F}^{(e_{aft})} = 0$. This completes the proof.

This bound is tight, and can be obtained with the following set \mathcal{F} . A frame x is received by both gateways, and another frame y such that $t_x^{(s)} < t_y^{(s)} < t_x^{(e)} < t_y^{(e)}$ is received by GW_2 only. P chooses x on both gateways, and does not choose y (as y does not preempt x). In the meanwhile, OPT chooses both x on GW_1 and y on GW_2 . \square

Theorem 5 (Approximation ratio of algorithm P^C for $M = 2$ and $D = 1$). *For all \mathcal{F} , $\frac{\Gamma(OPT(\mathcal{F}))}{\Gamma(P^C(\mathcal{F}))} \leq 2$ for $M = 2$ and $D = 1$. This bound is tight.*

Proof. The proof of the approximation ratio is the same as the proof of Theorem 4, without the possibility of Case 1.

This bound is tight, and can be obtained with the following set \mathcal{F} . A frame x is received by both gateways, and another frame y such that $t_x^{(s)} < t_y^{(s)} < t_x^{(e)} < t_y^{(e)}$ is received by GW_2 only. P^C sees x on both gateways, and decides to demodulate x on GW_2 . Then, P^C can not choose y , as y is not received by GW_1 and y does not preempt x on GW_2 . In the meanwhile, OPT chooses both x and y , on GW_1 and GW_2 respectively. \square

Theorem 6 (Approximation ratio of algorithm P^S for $M = 2$ and $D = 1$). *For all \mathcal{F} , $\frac{\Gamma(OPT(\mathcal{F}))}{\Gamma(P^S(\mathcal{F}))} \leq 1.5$ for $M = 2$ and $D = 1$.*

Lemma 2. *Let x_1, x_2, \dots, x_n and y_1, y_2, \dots, y_n be two non-empty sequences of n strictly positive integers. Let $X_1^{(n)} = \sum_{i \in [1; n]} (x_i + 1)$, $X_2^{(n)} = \sum_{i \in [1; n]} (x_i + 2)$ and $Y^{(n)} = \sum_{i \in [1; n]} y_i$.*

$$\text{Then, } \frac{X_2^{(n)} + Y^{(n)}}{X_1^{(n)} + Y^{(n)}} \leq \frac{X_2^{(n)}}{X_1^{(n)}}.$$

Proof.

$$\begin{aligned} \frac{X_2^{(n)} + Y^{(n)}}{X_1^{(n)} + Y^{(n)}} - \frac{X_2^{(n)}}{X_1^{(n)}} &= \frac{(X_2^{(n)} + Y^{(n)})X_1^{(n)}}{(X_1^{(n)} + Y^{(n)})X_1^{(n)}} - \frac{X_2^{(n)}(X_1^{(n)} + Y^{(n)})}{X_1^{(n)}(X_1^{(n)} + Y^{(n)})} \\ &= \frac{Y^{(n)}(X_1^{(n)} - X_2^{(n)})}{X_1^{(n)}(X_1^{(n)} + Y^{(n)})} \\ &\leq 0. \end{aligned}$$

Indeed, $X_1^{(n)}(X_1^{(n)} + Y^{(n)}) > 0$, $Y^{(n)} \geq 0$ and $X_1^{(n)} \leq X_2^{(n)}$. \square

Lemma 3. Let x_1, x_2, \dots, x_n be a non-empty sequence of n strictly positive integers. Let $X_1^{(n)} = \sum_{i \in [1; n]} (x_i + 1)$ and $X_2^{(n)} = \sum_{i \in [1; n]} (x_i + 2)$. Then, $\frac{X_2^{(n)}}{X_1^{(n)}} \leq \frac{3}{2}$.

Proof. Let us prove this lemma by recursion on n . When $n = 1$, we have:

$$\frac{X_2^{(1)}}{X_1^{(1)}} - \frac{3}{2} = \frac{x_1 + 2}{x_1 + 1} - \frac{3}{2} = \frac{1 - x_1}{2(x_1 + 1)} \leq 0.$$

Indeed, $x_1 \geq 1$ since each integer of the sequence is strictly positive.

Let us now assume that the property is true for $n - 1 \geq 1$, that is $\frac{X_2^{(n-1)}}{X_1^{(n-1)}} \leq \frac{3}{2}$. Let us show that the property is true for n too. We have:

$$\begin{aligned} \frac{X_2^{(n)}}{X_1^{(n)}} - \frac{3}{2} &= \frac{2X_2^{(n)} - 3X_1^{(n)}}{2X_1^{(n)}} \\ &= \frac{2(X_2^{(n-1)} + x_n + 2) - 3(X_1^{(n-1)} + x_n + 1)}{2X_1^{(n)}} \\ &= \frac{1 - x_n + 2X_2^{(n-1)} - 3X_1^{(n-1)}}{2X_1^{(n)}} \leq \frac{1 - x_n}{2X_1^{(n)}} \leq 0. \end{aligned}$$

This comes from the fact that $2X_2^{(n-1)} - 3X_1^{(n-1)} \leq 0$, according to the recursion hypothesis, and that $x_n \geq 1$. \square

Proof of Theorem 6. Let \mathcal{F} be an arbitrary set of frames. From $P^S(\mathcal{F})$, we will build the solution $OPT(\mathcal{F})$ with the largest gain, and show that $\frac{\Gamma(OPT(\mathcal{F}))}{\Gamma(P^S(\mathcal{F}))} \leq 3/2$. Let us divide the time into new intervals depending on $P^S(\mathcal{F})$: a new interval starts as soon as both gateways GW_1 and GW_2 demodulated at least a frame each. Then, for any time interval δ_i , we use the same notations as in Theorem 4 for $t_{\delta_i}^{(s)}$, $t_{\delta_i}^{(e)}$, $\Gamma_{\delta_i}(A(\mathcal{F}))$, $n_{\delta_i}^{(s\text{bef})}$, $n_{\delta_i}^{(e\text{dur})}$ and $n_{\delta_i}^{(e\text{aft})}$. Note that, unlike for intervals defined for P , it is here possible to have $\Gamma_{\delta_i}(P^S(\mathcal{F})) > 1$ since both gateways should decode at least one frame each during δ_i .

Let us consider any interval δ_i , for $i \in [1; n]$. For each of the following cases, we will show that $n_{\delta_i}^{(e\text{dur})} + n_{\delta_i}^{(e\text{aft})} \leq \Gamma_{\delta_i}(P^S(\mathcal{F}))$ in some cases, that $n_{\delta_i}^{(e\text{dur})} + n_{\delta_i}^{(e\text{aft})} \leq \Gamma_{\delta_i}(P^S(\mathcal{F})) + 1$ in the other cases, and that we always have $\Gamma_{\delta_i}(P^S(\mathcal{F})) \geq 1$.

- Case 1: both gateways demodulate the same frame f . This means that no other frame started during δ_i , otherwise one of the gateway would have dropped f with P^S . Thus, $n_{\delta_i}^{(e\text{dur})} \leq 1$, $n_{\delta_i}^{(e\text{aft})} = 0$ and $\Gamma_{\delta_i}(P^S(\mathcal{F})) = 1$.

- Case 2: the last demodulated frame f finishing in P^S is demodulated by GW_2 , but not by GW_1 . According to the construction of δ_i , this means that GW_1 demodulated ≥ 1 frames with P^S , denoted a_1, a_2, \dots, a_x , while GW_2 demodulated only one, denoted f .

- On GW_1 , there can be no frame f' in OPT such that $t_{f'}^{(s)} < t_{a_j}^{(e)}$ and $t_{f'}^{(e)} < t_{a_{j+1}}^{(s)}$, nor such that $t_{a_j}^{(s)} < t_{f'}^{(s)}$ and $t_{f'}^{(e)} < t_{a_j}^{(e)}$, for all $j \in [1; x]$. However, for each $j \in [1; x]$, OPT can choose either a_j or a frame starting after $t_{a_j}^{(s)}$ and ending after $t_{a_j}^{(e)}$. Thus, $n_{\delta_i, GW_1}^{(e\text{dur})} + n_{\delta_i, GW_1}^{(e\text{aft})} \leq x$.
- On GW_2 , there can be no frame f' such that $t_f^{(s)} < t_{f'}^{(s)}$ and $t_{f'}^{(e)} < t_f^{(e)}$, otherwise P^S would have dropped f for this f' . OPT can choose either to demodulate f on GW_2 , or a frame starting after $t_f^{(s)}$ and finishing after $t_f^{(e)}$. OPT can also demodulate a frame a_j of P^S , if this frame was received by both GW_1 and GW_2 , provided that this frame was dropped by P^S on GW_2 in order to start frame f . Thus, $n_{\delta_i, GW_2}^{(e\text{dur})} + n_{\delta_i, GW_2}^{(e\text{aft})} \leq 2$.
- Overall, when considering both GW_1 and GW_2 , we have: $n_{\delta_i}^{(e\text{dur})} + n_{\delta_i}^{(e\text{aft})} \leq x + 2$, with $\Gamma_{\delta_i}(P^S(\mathcal{F})) = x + 1$ and $x \geq 1$.

- Case 3: the last demodulated frame f finishing in P^S is demodulated by GW_1 , but not by GW_2 . This is similar to Case 2.
- Case 4: no more frame is demodulated on GW_1 during δ_i , which occurs only when $i = n$. This means that no frame can start for OPT from $t_{\delta_n}^{(s)}$ on GW_1 . In this case, P^S works on the single gateway GW_2 , and is thus optimal according to Theorem 2. Thus, $\Gamma_{\delta_i}(P^S(\mathcal{F})) = \Gamma_{\delta_i}(OPT(\mathcal{F}))$.
- Case 5: no more frame is demodulated on GW_2 during δ_i , which occurs only when $i = n$. This is similar to Case 4.

Let $\mathcal{C}_{1,4,5}$ be the indexes such that δ_i is in Case 1, Case 4 or Case 5, and let $\mathcal{C}_{2,3}$ be the indexes such that δ_i is in Case 2 or Case 3. We have

$$\begin{aligned} \Gamma(P^S(\mathcal{F})) &= \sum_{i \in \mathcal{C}_{1,4,5}} \Gamma_{\delta_i}(P^S(\mathcal{F})) + \sum_{i \in \mathcal{C}_{2,3}} \Gamma_{\delta_i}(P^S(\mathcal{F})) \\ &= Y^{(n)} + \sum_{i \in \mathcal{C}_{2,3}} (x_i + 1), \end{aligned}$$

where we define $Y^{(n)} = \sum_{i \in \mathcal{C}_{1,4,5}} \Gamma_{\delta_i}(P^S(\mathcal{F}))$, and it was shown that $x_i = \Gamma_{\delta_i}(P^S(\mathcal{F})) - 1$. With the same argument as in Theorem 4, we also have

$$\Gamma(OPT(\mathcal{F})) = \sum_{i \in [1; n]} \left(n_{\delta_i}^{(e\text{dur})} + n_{\delta_i}^{(e\text{aft})} \right).$$

From the fact that $n_{\delta_i}^{(e\text{dur})} + n_{\delta_i}^{(e\text{aft})} \leq \Gamma_{\delta_i}(P^S(\mathcal{F}))$ for Cases 1, 4 and 5, and that $n_{\delta_i}^{(e\text{dur})} + n_{\delta_i}^{(e\text{aft})} \leq \Gamma_{\delta_i}(P^S(\mathcal{F})) + 1$ for

Cases 2 and 3, we have:

$$\begin{aligned} \Gamma(OPT(\mathcal{F})) &= \sum_{i \in \mathcal{C}_{1,4,5}} \Gamma_{\delta_i}(OPT(\mathcal{F})) + \sum_{i \in \mathcal{C}_{2,3}} \Gamma_{\delta_i}(OPT(\mathcal{F})) \\ &\leq Y^{(n)} + \sum_{i \in \mathcal{C}_{2,3}} (x_i + 2). \end{aligned}$$

Then, setting $X_1^{(n)} = \sum_{i \in \mathcal{C}_{2,3}} (x_i + 1)$ and $X_2^{(n)} = \sum_{i \in \mathcal{C}_{2,3}} (x_i + 2)$, we get

$$\frac{\Gamma(OPT(\mathcal{F}))}{\Gamma(P^S(\mathcal{F}))} \leq \frac{Y^{(n)} + X_2^{(n)}}{Y^{(n)} + X_1^{(n)}} \leq \frac{X_2^{(n)}}{X_1^{(n)}}$$

Finally, we obtain $\frac{\Gamma(OPT(\mathcal{F}))}{\Gamma(P^S(\mathcal{F}))} \leq \frac{3}{2}$, according to Lemma 3. \square

VII. NUMERICAL EVALUATIONS

This section describes our numerical evaluations.

A. Simulation settings

We consider the following six configurations corresponding to various pairs (M, D) :

- one gateway and one demodulator (condition of application of Theorem 1 and Theorem 2),
- one gateway and two demodulators per gateway (for Theorem 3),
- one gateway and three demodulators per gateway,
- two gateways and one demodulator per gateway (for Theorem 4, Theorem 5 and Theorem 6),
- two gateways and three demodulators per gateway,
- three gateways and three demodulators per gateway.

For each configuration, we generated a given number of frames, either equal to $100 \times M \times D$ (in order to have a charge proportional to the overall demodulation capacity) or to 100 (in order to evaluate the impact of adding more gateways, or of more demodulators per gateway). Each frame has a random payload size varying from 10 to 51 bytes (where 51 bytes is the maximum payload size in LoRaWAN) and corresponds to a random SF from 7 to 12, as shown in Table II [13], [14]. We considered a bandwidth of 125 kHz and a coding rate of $4/5$. The starting time of each frame is chosen randomly. When $M \geq 2$, the frame is received by at least one gateway chosen at random, and by every other gateway with a probability of 30%. Each simulation lasts for 100 seconds. For each repetition, we use the MILP solver GLPK [15] to solve Problem (3)-(9), in order to obtain the optimal solution (or an interval in which the optimal solution lies). GLPK is used with a timeout of 60 s, on a computer with an Intel(R) Core(TM) i7-4790 CPU at 3.60GHz. When the solver does not reach an optimal solution, it still outputs an interval $[LB; UB]$ (hereafter referred to as the optimality interval), where LB is a lower bound, namely the gain of a good solution (possibly sub-optimal), and UB is an upper bound of the optimal solution (possibly unreachable). We generated 100 repetitions in order to obtain average values.

B. Simulation results

Figure 5 shows the percentage of demodulated frames with a traffic load proportional to the demodulation capacity (and computed as $100 \times M \times D$ frames). Overall, the algorithms can always be ranked by performance as follows : $G < P < P^C < P^S$.

For $M = 1$, P , P^C and P^S yield the same results, as expected. Moreover, it can be seen that they achieve an excellent percentage of demodulated frames. They appear to be optimal for $D = 1$ (which is expected from Theorem 2), and are within the optimality interval for $D = 2$ (which is expected from Theorem 3), and $D = 3$. Please note that more simulations results are provided for $M = 1$ and $D = 2$ in the sequel.

For $M = 2$ and $D = 1$, the algorithms are clearly sub-optimal since they are below the LB level, and decode from about 70% to about 75% of the frames, while LB is slightly above 80%. This shows that while the approximation ratio of these algorithms is between 1.5 and 2, they still perform well. For the two configurations with multiple gateways and $D = 3$, the solver was not able to obtain a tight interval, and was even rarely able to find a solution before the timeout of 60 s expired. However, all algorithms are able to demodulate a very large number of frames, namely around 85% or more.

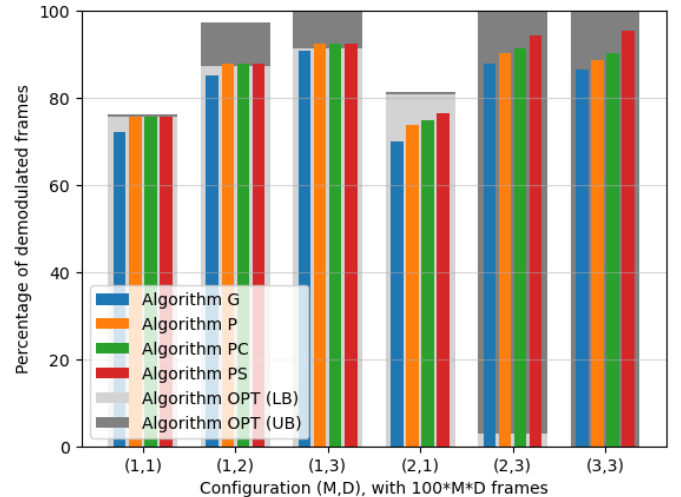


Figure 5. Performance of all algorithms with a traffic load proportional to the demodulation capacity.

Figure 6 shows the performance of all algorithms, with a constant traffic load, in two cases:

- with $M = 1$ gateway and a varying number of demodulators (on the left),
- with a varying number of gateways and $D = 3$ demodulators (on the right).

Obviously, as the demodulation capacity increases (either when M or D increases), the performance of all algorithms increases. On the left side of Fig. 5 (that is, when $M = 1$), P , P^C and P^S have the same performance, and are all within the optimality interval. On the right side of Fig. 5 (that is, when $D = 3$), the solver was not able to obtain a tight interval.

Moreover, it can be seen that algorithm P^S ensures the best performance among all algorithms, in all cases.

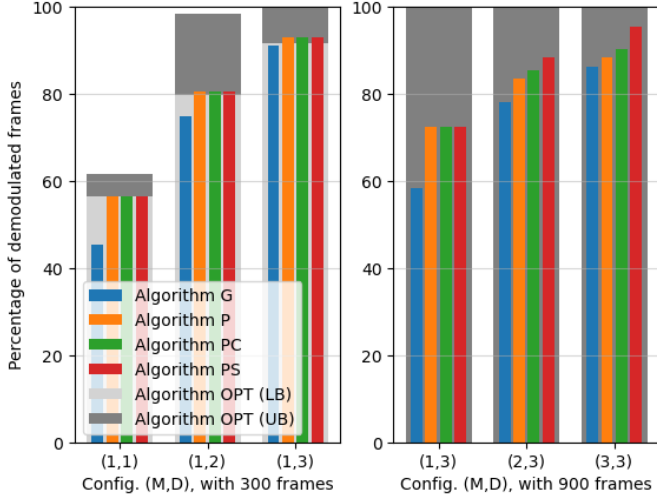


Figure 6. Performance of all algorithms, with a varying number of demodulators (left) or with a varying number of gateways (right), with a constant traffic load.

Note that we also explicitly tested Theorem 3 by running many repetitions comparing OPT and P with $M = 1$ and $D = 2$. We used relatively small instances of 20 frames and a simulation time of 20 seconds in order to obtain optimal solutions faster, and we set a longer timeout of 10 minutes for the solver. Repetitions for which the optimal solution was not found before the timeout were discarded, until we obtained 1000 repetitions with the optimal solution. For each repetition, we verified that the optimal gain was equal to the gain of P . The average gain was 80.08% for both algorithms.

VIII. CONCLUSION

Given their limited number of demodulators, when LoRaWAN gateways detect more than eight incoming frames, they have to decide which ones to demodulate and which ones to discard. We showed that the default behavior of LoRaWAN gateways, which is to demodulate the earliest frames in a FIFO manner, is far from optimal as it is possible to demodulate up to 72 times more frames (in the worst case). In the single gateway case, the simple preemptive algorithm P is optimal for one and two demodulators, and reaches good performance with a larger number of demodulators. In the multi-gateway case, both the non-collaborative algorithm P and the simple-collaborative algorithm P^C yield an approximation ratio of 2 (for two gateways and one demodulator per gateway), while the smart-collaborative algorithm P^S has a lower approximation ratio of 1.5 (under the same conditions). We believe that designing simple collaborative strategies for demodulator allocation can help to enhance LoRaWAN throughput, with minimal additional overhead in order to implement this collaboration framework. The optimality or approximation ratios of the proposed collaborative algorithms have not only been analyzed theoretically, but also validated through extensive simulations. The proposed approach thus appears as an effective solution towards the future challenges of massive IoT

connectivity, by smartly leveraging the increasing density of LoRaWAN gateways.

REFERENCES

- [1] Semtech, “SX1301 - digital baseband chip LoRaWAN macro gateways,” Wireless & Sensing Products, datasheet version 2.4, 06 2017.
- [2] R. B. Sorensen, N. Razmi, J. J. Nielsen, and P. Popovski, “Analysis of LoRaWAN uplink with multiple demodulating paths and capture effect,” in *ICC (IEEE International Conference on Communications)*. Shanghai, China: IEEE, 2019, p. 6.
- [3] P. K. Dalela, S. Sachdev, and V. Tyagi, “LoRaWAN network capacity for practical network planning in India,” in *AP-RASC (URSI Asia-Pacific Radio Science Conference)*. New Delhi, India: URSI, 2019, p. 4.
- [4] D. Magrin, M. Capuzzo, and A. Zanella, “A thorough study of LoRaWAN performance under different parameter settings,” *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 116–127, 2020.
- [5] A. Guitton and M. Kaneko, “Improving LoRa scalability by a recursive reuse of demodulators,” in *Globecom (IEEE Global Communications Conference)*. Taipei, Taiwan: IEEE, 2020, p. 6.
- [6] —, “Multi-gateway demodulation in LoRa,” in *IEEE Globecom (Global Communications Conference)*. Rio de Janeiro, Brasil: IEEE, 2022, pp. 2008–2013.
- [7] J.-T. Chen, M. Kaneko, and A. Guitton, “An auction-based assignment method for LoRa multi-gateway networks,” in *ICC (IEEE International Conference on Communications)*. Rome, Italy: IEEE, 2023, p. 6.
- [8] X. Miao, P. B. Luh, and D. L. Kleinman, “Dynamic job scheduling with strict deadline,” in *IEEE Conference on Decision and Control*. Honolulu, United States of America: IEEE, 1990, pp. 116–121.
- [9] E. L. Lawler, “A dynamic programming algorithm for preemptive scheduling of a single machine to minimize the number of late jobs,” *Ann. Oper. Res.*, vol. 26, pp. 125–133, 1990.
- [10] J. P. Erickson and J. H. Anderson, *Soft Real-Time Scheduling*. Singapore: Springer, 2022, pp. 233–267.
- [11] A. Rahmadhani and F. Kuipers, “When LoRaWAN frames collide,” in *WiNTECH (International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization)*. New Delhi, India: ACM, 2018, pp. 89–97.
- [12] J. Haxhibeqiri, F. Van den Abeele, I. Moerman, and J. Hoebeke, “LoRa scalability: a simulation model based on interference measurements,” *Sensors*, vol. 17, no. 6, p. 1193, 2017.
- [13] Semtech Corporation, “AN1200.22 LoRa Modulation Basics,” Semtech, Application note Revision 2, 2015. [Online]. Available: <http://www.semtech.com/uploads/documents/an1200.22.pdf>
- [14] —, “LoRaWAN Specification v1.1,” Semtech, Tech. Rep. Revision B, 2017. [Online]. Available: https://loralliance.org/sites/default/files/2018-04/lorawantm_specification_v1.1.pdf
- [15] A. Makhorin, “GLPK (GNU linear programming kit),” Free Software Foundation, 2012. [Online]. Available: <https://www.gnu.org/software/glpk/>