A New Distributed and Probabilistic Approach for Traffic Control in LPWANs

Kawtar Lasri, Yann Ben Maissa, Loubna Echabbi, Oana Iova, Fabrice Valois

HAL Id: hal-03178390

https://hal.archives-ouvertes.fr/hal-03178390

Submitted on 23 Mar 2021

A New Distributed and Probabilistic Approach for Traffic Control in LPWANs

Kawtar Lasri[1], Yann Ben Maissa[2], Loubna Echabbi[3], Oana Iova[4], and Fabrice Valois[5]

[1,4,5]Univ Lyon, INSA Lyon, Inria, CITI, EA3720, 69621 Villeurbanne, France
[1,2,3]INPT Rabat, Morocco

Abstract

Low-Power Wide Area Networks (LPWANs) are wireless networks with very low power consumption and wide area coverage. They are capable of supporting the traffic of nearly a thousand nodes with a duty cycle of less than 1%. However, the gradual densification of nodes increases the number of collisions and makes it more difficult to manage the upstream traffic. To mitigate this problem, we propose a new distributed and probabilistic traffic control algorithm, DiPTC, which allows nodes to adapt their traffic according to the needs of the application (e.g., receiving K measurements over a time period) while being agnostic to the number of nodes and to the network topology. A control message is broadcast by the gateway to all nodes each period when the objective is not reached, so that nodes can re-adapt their traffic. We evaluate the proposed solution in simulation and we compare it with the LoRaWAN protocol. The results show that our algorithm is able to reach the objective while keeping a low number of collisions, with a longer network lifetime. Compared to LoRaWAN, our solution shows a three times increase in the success rate and a decrease by a factor of 10 in the collision rate. [1]

1 Introduction

Data collection is one of the main applications used to deploy sensors in smart buildings and smart cities. To rapidly provide a data collection infrastructure at reasonable costs, while covering several kilometers, Low Power Wide Area Networks (LPWANs) are a very good solution [7].

Despite their attractiveness, these networks suffer from limitations due to the used frequencies (free ISM band, *e.g.* 868MHz in Europe) and medium access protocol (pure Aloha). In order to achieve long running networks and to support scalable deployments, traffic control is mandatory, especially in two different scenarios. First, in applications where the objective is to get a sampling of a given situation in an area (e.g. environmental monitoring), only a fixed number of measurements per time unit is necessary. On one hand, to receive more information does not bring any added-value and it only leads to network overload, increased collisions and energy consumption. On the other hand, to receive less information leads to a non efficient application. The second scenario is given when an LPWAN is operated by a given telecommunication operator. The operator provides coverage and connectivity for its clients. A service-level agreement (SLA) is defined to specify the capacity provided by the network provider, and the traffic model of the client. Unfortunately, there is no mechanism in LPWAN to manage the traffic sent by a user and the number of nodes connected to the network. Thus, a client can send more data and can connect more end devices, leading to network congestion. We claim that these scenarios supported in LPWANs will be more and more common. Note that mechanisms such as the adaptive data rate (ADR) implemented in LoRaWAN are not able to cope with this challenge. The objective of ADR is to minimize the energy consumption and to adapt the data rate according to the radio link budget and environmental conditions, but it is not the answer to congested networks. ADR is a local mechanism for each end device, without any capabilities to adapt the traffic of a group of nodes.

In this paper, we propose DiPTC, a new distributed, probabilistic and network topology agnostic algorithm for traffic control in LPWANs that allows the network manager and the applications to better steer the data collection. In a nutshell, DiPTC works as follows: (i) A central server (i.e., the network server in LoRaWAN, or the core network server in Sigfox) acts as traffic policy enforcer by running a control loop. If the amount of received data does not correspond to the expected value the server schedules a control message to be sent by a gateway to the end devices. (ii) A local node mechanism adapts the traffic intensity taking into account the information received in the control message. It is important to notice that the end devices do not have any neighborhood knowledge and do not use any inter-node communication.

The rest of the paper is organized as follows: after discussing the state of the art for traffic control in wireless sensor networks and LPWANs in Section 2, we introduce and explain our proposed solution in Section 3. Section 4 describes the simulation setup and the scenarios that are used to validate, discuss and compare our results to the baseline LoRaWAN solution in Section 5. We conclude and present our future work in Section 6.

2 Related work

In wireless sensor networks, spatial and temporal data aggregation are presented by many researchers as a solution for traffic reduction, and thus collision reduction. Spatial data aggregation is based on the organisation of nodes in the network and is done by choosing a messenger or a group of messenger device(s) that are responsible to send the gathered data to the gateway [4, 6]. To gather the data, a communication between the messenger(s) and the nodes is essential. A temporal data aggregation based on data prediction is described in [5, 12], where the authors focus on the temporal aggregation functions using ARIMA and LMS-PCA prediction models. Given an error of exceeded threshold, the nodes adapt their prediction model and send their new model coefficients to the gateway. Those coefficients are used by the gateway to predict data. In order to use data aggregation methods to solve the stated problem, nodes must have a large memory capacity, a neighborhood knowledge and the possibility to communicate between them. Unfortunately these constraints cannot be satisfied in LPWANs, therefore it is not possible to apply temporal aggregation techniques to our problem.

In LoRaWAN, researchers try to reduce the number of collisions and improve scalability by optimizing the allocation of resources, such as spreading factor and transmission power [1], [11], [13]. Ta *et al.* present LoRa-MAB, a flexible decentralized learning resource allocation algorithm based on the reinforcement learning problem Multi-Armed-Bandit [11]. Their approach far exceeds the results of the classic Adaptive Data Rate (ADR) algorithm implemented in LoRaWAN networks. Besides looking at the spreading factor, Luo *et al.* take into account the periodicity characteristics of traffic usually present in LPWAN applications, to propose a transmission scheduler (S-MAC) [13]. However these approaches only adapt the physical layer parameters of the existing traffic to the environmental conditions. They do not control the amount of traffic sent by the nodes, and they do not ensure that the traffic respects the constraints of the application. To the best of our knowledge no such traffic control mechanism exists today for LPWANs.

3 Towards a distributed and probabilistic approach

We propose to control the uplink traffic through a distributed and probabilistic selection of a set of nodes, denoted as active nodes, which will be responsible to send the required data to the gateway, instead of all the nodes in the network.

3.1 Assumptions

Without loss of generality, we consider the case of a single gateway covering a given area. We also consider that the number of nodes in the network, denoted as N, is unknown. This number can evolve over time depending on successive deployments, hardware failures, or depleted batteries. This approach may target monitoring applications of a physical quantity varying over time, such as pollution [3], where the application needs K measurements over a given area periodically, every ΔT, whatever the number of nodes in the network.

In LPWANs, the downlink is mainly used to acknowledge correctly received packets at the gateway [9]. In this study, we propose to use a broadcast downlink as a source of feedback information, which can indicate, for example, the number of packets actually received during the previous period ΔT as well as the number of

the K packets needed by the application. It is possible to consider any other type of message in order to notify if the objective has been reached. Thus, depending on the received information and the local regulations, each node can adapt its traffic to meet the needs of the application. The implementation of the broadcast downlink in an LPWAN is out of the scope of this paper.

3.2   The DiPTC algorithm

We propose DiTC, an algorithm for controlling the traffic in the network based on the additive increase multiplicative decrease mechanism. When too much data is sent by the nodes, we apply a decreasing policy using the decrease factor x_D. Contrary, when not enough data is sent by the nodes, we apply an increasing policy using the increase factor x_I. This mechanism provides a rapid response by adjusting the traffic intensity with respect to the information transmitted by the gateway at each time period ΔT. To reduce the payload of the downlink message, we consider the feedback information sent by the gateway as a binary value. In other words, the gateway sends 0 when the number of received packets exceeds the needed ones, and 1 when the nodes do not send enough packets. When the objective is reached, no downlink is sent by the gateway, and the nodes continue their current traffic rate. As long as the gateway does not receive the K packets needed by the application per time period ΔT, each node increases its traffic intensity linearly (coefficient x_I). Otherwise, when the gateway receives more than K packets, exceeding the threshold, the nodes reduce exponentially their traffic intensity (coefficient x_D).
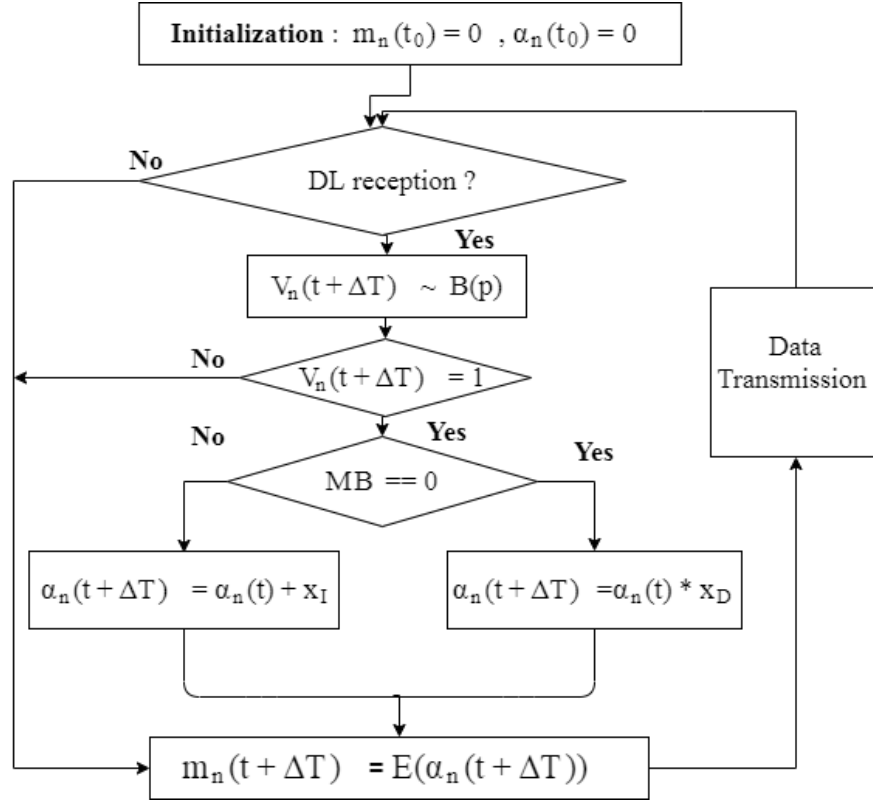

Figure 1: Finite state machine of the DiPTC algorithm.

The finite state machine in Fig. 1 describes the DiPTC algorithm adopted by each node in the network. At first, each node awaits the reception of the broadcast message (MB) to decide on the number of messages, $m_n(t + \Delta T)$, to transmit in the next period ΔT. The broadcast message MB is sent by the gateway only if the number of received messages is different from the needed one, therefore reducing the downlink traffic. Then, each node decides to increase or to decrease its previous traffic intensity $\alpha_n(t)$ according to the binary value of the broadcast message received from the gateway. In other words, if the value of MB is equal to 1, the

3

number of messages that the node will transmit is increased by x_I, $x_I \in]0,1]$. Otherwise, the node decreases its traffic intensity exponentially using the multiplication factor x_D, $x_D \in]0,1]$. The additive factor x_I and the multiplicative factor x_D are constant over time and equal for all nodes. They represent respectively, the proportion of traffic to increase or to decrease from the previous period. The larger x_I is, the faster is the increase, and the larger x_D is, the slower is the decrease. Next, the node transmits its $m_n(t + \Delta T)$ messages, which are the integer portion of $\alpha_n(t + \Delta T)$. We consider a node active when $m_n(t + \Delta T) > 0$. Finally, in order to avoid the synchronization of traffic intensities for all the nodes we introduced a local variable that works as an adaptation probability. More specifically, each node uses a random variable $V_n(t + \Delta T)$ that follows a Bernoulli distribution of parameter p, to decide if it should take into account ($p = 1$) or not ($p = 0$) the broadcast message.

4 Simulation Model

While DiPTC can be applied to any LPWAN technology, we chose LoRaWAN as our testing protocol, which we also use as baseline for comparison. We implemented and evaluated our solution in LoRaSim [2], a well know discrete event simulator based on SimPy that allows simulating collisions, the capture effect and interference in LoRaWAN. We enhanced this simulator with a downlink model and a battery depletion model, which we present next.

4.1 Wireless environment

LoRaSim does not implement downlink communication in LoRaWAN, so we enhanced the simulator by:

1. Adding a downlink model. For the LoRaWAN protocol used as baseline, the downlink is implemented in a unicast mode, to acknowledge the received packets. For DiPTC, the downlink is implemented in a broadcast mode, from the gateway to all the nodes.

2. Modeling the reliability of the downlink through a random variable that follows a Bernoulli distribution of parameter p_{DL} for both solutions.

To model the collisions and the interference on the uplink, we kept the ones used by Roedig et al. [2] for both approaches:

$$P_{rx} = P_{tx} + GL - L_{pl}(d) \tag{1}$$

$$L_{pl}(d) = \bar{L}_{pl}(d_0) + 10\gamma\log(d/d_0) + X_\sigma \tag{2}$$

where P_{rx} is the power of the received signal, P_{tx} the transmission power, GL the accumulated general gains losses along the communication path, $L_{pl}(d)$ the log-distance path loss model determined by the nature of the communication environment, $L_{pl}(d)$ the path loss in dB at the communication distance d, $\bar{L}_{pl}(d_0)$ the mean path loss at the reference distance d_0, γ the path loss exponent, and $X_\sigma \sim N(0,\sigma^2)$ the normal distribution with zero mean, and σ^2 the variance to account for shadowing.

4.2 Energy consumption

LoRaSim does not implement any energy consumption model for the nodes. We extended the simulator by adding the energy consumption model described in [8]. The energy consumed E_c after transmitting m messages for each node n, depends on the time on air TOA, the power consumed in the receiver mode $P_{w_{Rx}}$ and the power consumed in the transmitter mode $P_{w_{Tx}}$.

$$E_c = TOA * (P_{w_{Tx}} * m + P_{w_{Rx}}) \text{ for DiPTC} \tag{3}$$

$$E_c = TOA * m * (P_{w_{Tx}} + P_{w_{Rx}}) \text{ for LoRaWAN} \tag{4}$$

The values used for the parameters in these models are presented in Table 3.

5 Performance evaluation

This section presents the evaluation of our proposed algorithm DiPTC. We start by first presenting the setup used in our simulations then we comment on the results.

5.1 Scenarios and network parameters

We consider the case of one centered gateway surrounded by N nodes distributed randomly with LoRa parameters chosen randomly. We consider the two simulation scenarios described in Table 2 in order to evaluate the performance of our proposed solution DiPTC, and to compare it against the baseline LoRaWAN. For each scenario, we specify the number of nodes N in the network, the number of measurements K needed by the application, the time period ΔT, the increasing and decreasing parameters (x_I, x_D), the adaptation probability and the simulation time *Simtime*. The adaptation parameters x_I, x_D and p depend on the number of nodes in the network and the traffic intensity, and were chosen accordingly. Given the limited number of pages we will not detail here how they were chosen, their optimization in function of different network conditions is the subject of future work.

In the first scenario, denoted as BASIC, we aim to evaluate DiPTC in the case of a normal traffic, while in the second scenario, denoted as INTENSIVE, we consider a significantly higher traffic load ($\times 10$) to test the scalability of our proposed solution. While in DiPTC the traffic intensity of each node evolves dynamically according to the feedback information received from the gateway, in baseline LoRaWAN, the traffic intensity is constant. To propose a fair comparison and to respect the application constraints, i.e., receiving K measurements every time period ΔT, in baseline LoRaWAN nodes send their packet following a Poisson distribution with the rate: $AVG = \frac{\Delta T}{K} \times N$.

Table 1: Simulation scenarios.

Scenario	BASIC	INTENSIVE
DiPTC		
N	150	150
K	1	10
ΔT	10 min	1 min
x_I	0.5	0.2
x_D	0.5	0.5
p	0.5	0.06
Simtime	1 year	1 year
LoRaWAN		
AVG	1500 min	15 min

Table 2: Simulation scenarios.

Table 3: LoRa and network parameters.

Parameters	Values
Payload (PL)	20 bytes
Header (H)	0
Preamble symbols	8
Downlink reception probability p_{DL}	0.99
Transmission power P_{Tx}	14dBm
Gain and Loss GL	0
Path Loss exponent γ	2.08
Reference distance d_0	40m
Max. distance to the gateway	300m
Path Loss at the reference distance $L_{pl}(d_0)$	-127.41dB
Variance X_σ	N(0,3.57)
Current drawn during the receive mode I_{Rx}	11.2mA
Current drawn during the Transmission mode I_{Tx}	90mA
Current drawn during the sleep mode I_{Sleep}	1μA
Supply voltage	3V
Battery capacity	30J

In baseline LoRaWAN we configured a lost packet to be re-transmitted a maximum of 8 times before being dropped, a value commonly used in different implementations, considering that the standard specifies a maximum number of 15 re-transmissions. In DiPTC there are no re-transmissions, as the nodes do not receive unicast acknowledgments from the gateway.

The values of the propagation model are determined empirically in [2], and those of the energy consumption model can be found in Semtech SX1272 LoRa transceiver Datasheet [10]. Table 3, summarizes these values.

5.2 Performance evaluation in the BASIC scenario

We first evaluate the performance of DiPTC in the BASIC scenario by looking at the evolution of the number of packets sent and received, the number of collisions, the number of dead nodes, and the downlink and uplink losses throughout the whole lifetime of the network (Fig. 2). This type of detailed analysis allows us to dissect the behavior of DiPTC and better showcase its performance.

The first thing that we can notice in Fig. 2a is an alternation of transient and stationary states. Transient states of very short duration (due to the death of a node or a packet loss) are followed by long stationary states.
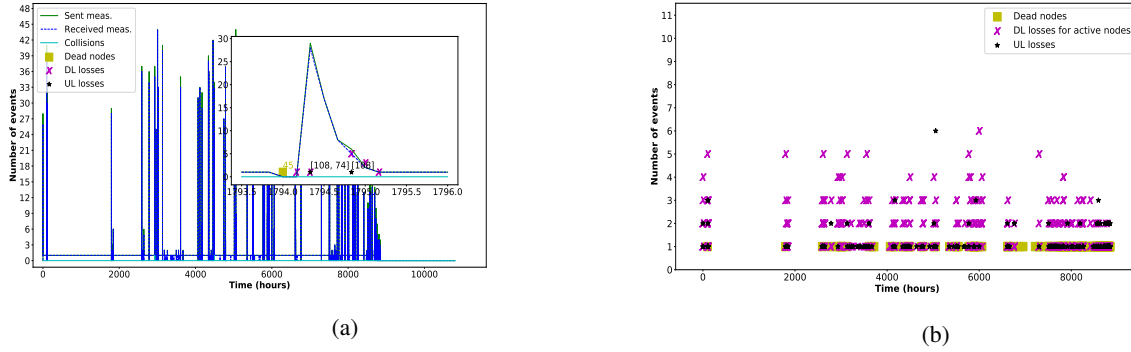
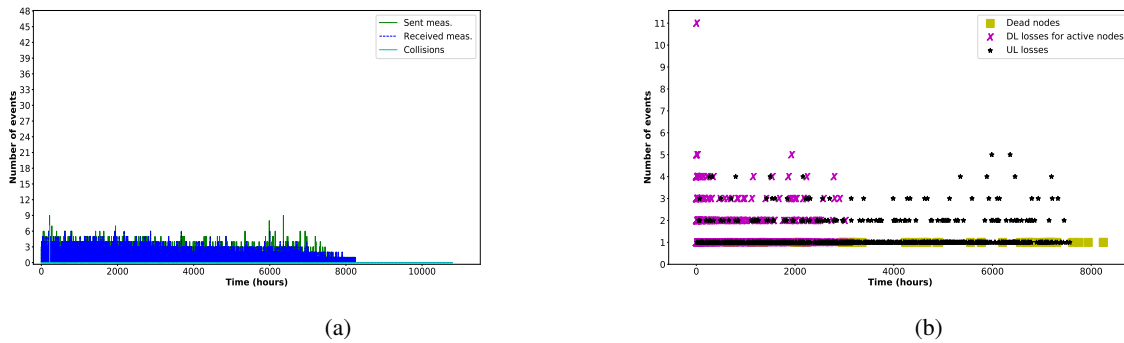Figure 2: Traffic evolution in the BASIC scenario for DiPTC.

Figure 3: Traffic evolution in the BASIC scenario for LoRaWAN.

This shows that our DiPTC algorithm is able to converge quickly to the desired number of measurements *K*, despite environmental challenges, and to remain stable for a significant time. Fig. 2b shows that the transient states are either caused by the death of a node or an uplink loss or both events. The loss of a downlink does not impact the convergence of our algorithm, as the node continues with the same traffic intensity until it receives a feedback from the gateway telling it otherwise. The frequency of the long duration stationary states is induced by the stability property of our mechanism and by the lack of collisions. The low collision rate is due to a low traffic intensity, consequence of a low number of expected measurements in this scenario (*K*=1). This low traffic also has a positive effect on the lifetime of a node as it reduces the battery depletion process.

In fact, as shown in the zoom part in Fig. 2a, the death of node 45 (time: 1794.0) generates a new transient state that lasts for $1h : 10min$ before a new stationary state takes place with a new active node with ID 108. Note that the downlink loss in 1795h delays the convergence because the feedback information is missing. Thus, we lost $\Delta T = 10min$ but the convergence is not affected. However the uplink losses in 1794.8h and 1794.3h extend the transient regime.

If we compare the performance of DiPTC with that of baseline LoRaWAN from Fig. 3a, the first thing that we notice is that the traffic in LoRaWAN appears to be lower than in DiPTC. Indeed, the maximum number of received messages is 9 messages per period (ΔT=10 minutes), while in DiPTC is 43. However, the oscillations in baseline LoRaWAN are more frequent than in DiPTC. This behaviour is due to the fact that the packet generation in baseline LoRaWAN follows a Poisson distribution of rate 0.0006, meaning that, in average, one packet is sent every 1500 minutes, leading the nodes to traffic less but more often, more *regularly*. In DiPTC, the traffic intensity is controlled by the additive increase multiplicative decrease mechanism that adapts the node activity when the goal of having 1 measure per 10 minutes is not met, and stabilizes it otherwise.

When the goal of the application is to receive a specific amount of messages per period of time, an important metric that allows us to evaluate the performance of different solutions is the absolute error. Con-
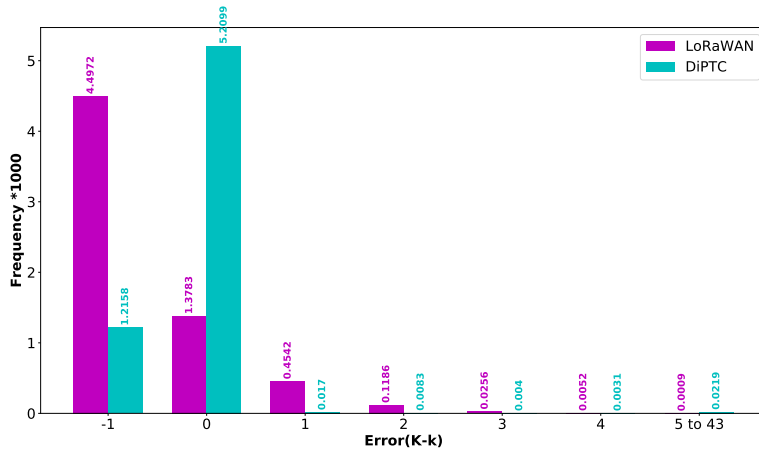
6

Figure 4: Error frequency in the BASIC scenario for DiPTC and Baseline LoRaWAN.
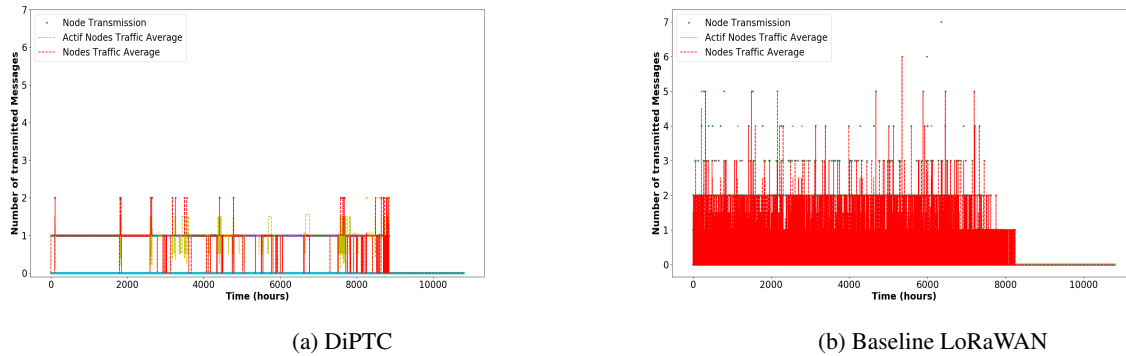
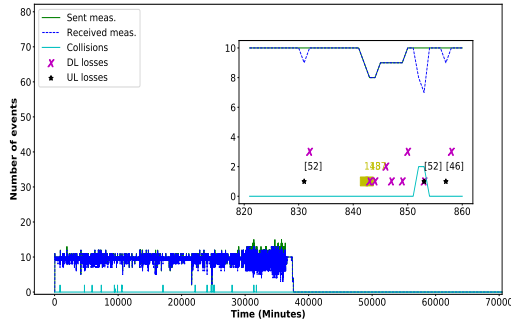

(a) DiPTC

(b) Baseline LoRaWAN

Figure 5: Nodes traffic evolution in the BASIC scenario.

sequently, we plotted in Fig. 4 the application error frequency for both baseline LoRaWAN and DiPTC. We can see that for DiPTC the application error is 0 most of the time, with some cases when it is slightly under. Contrary, LoRaWAN under-performs most of the time.

Finally, Fig. 5 shows the node traffic evolution in the BASIC scenario for baseline LoRaWAN and DiPTC. Note that the active node traffic average corresponds to the traffic of the nodes when $\alpha_n(t) > 0$, and the nodes traffic average correspond to those when $m_n(t) > 0$. These figures illustrate the need to have a data flow control in LoRaWAN. Note that the maximum number of packets that a node can send in DiPTC is 3 times smaller than the baseline LoRaWAN and the network traffic average is smoother than baseline LoRaWAN.

5.3 Impact of the traffic intensity

Fig. 6 illustrates the evolution of the number of collisions, the number of sent and received packets, the number of dead nodes, and the number of downlink and uplink losses, in the case of the INTENSIVE scenario for DiPTC. Unlike the BASIC scenario, the INTENSIVE presents several alternations between long duration transient states and short duration stationary states. These alternations are the result of destabilization produced by several factors: frequent and successive uplink and downlink losses, active node deaths and a high collision probability. The collision probability is higher in this scenario, since we increase the number of needed measurements by the gateway (K=10), while reducing the time period ($\Delta T = 1$ min). We observe in Fig. (6a) that even with death nodes and downlink losses, the received measurements are close to the objective of 10 measurements per time period. The convergence of our proposal depends mainly on the uplink
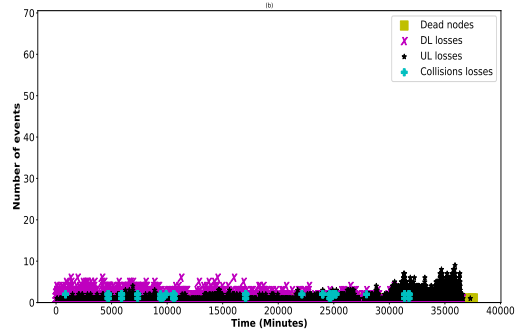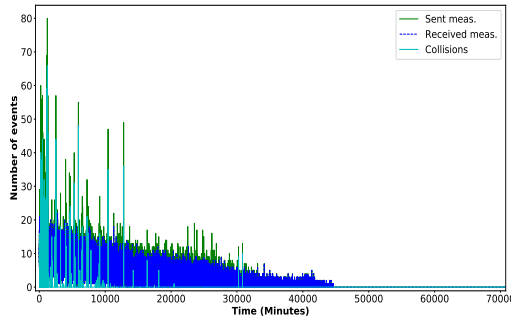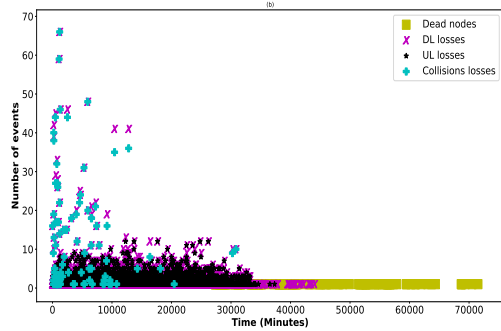
Figure 6: Traffic evolution in the INTENSIVE scenario for DiPTC.

Figure 7: Traffic evolution in the INTENSIVE scenario for Baseline LoRaWAN

losses. Moreover, the maximum absolute error in this case is 3, which means that the worst case is to receive 13 messages at the gateway. Considering the network density, this error is acceptable.

5.4   Overall performance evaluation

We finally make and overall performance evaluation of the baseline LoRaWAN and DiPTC using the following metrics:

- **Success rate** μ_c: measures the number of times that the base station receives the K needed measurements per period ΔT.

- **Collisions rate** τ_c: measures the number of packet collisions in the network.

- **Network lifetime** t_l: measures the duration the network is able to support the application requirements, i.e., the nodes are able to send the K needed measurements per period ΔT before the exhaustion of their battery.

We consider the same simulation scenarios and network parameters as above with longer simulation time, *Simtime* = 15 months. Table 4 presents the simulation results of both scenarios. For each metric, we calculated the mean and the standard deviation over 10 simulations.

We note that in the BASIC scenario DiPTC has a success rate of almost 100%, which is 3 times greater than LoRaWAN. This has an obvious positive effect on the network lifetime, which is longer for DiPTC.

In the INTENSIVE scenario DiPTC has a success rate of 29.64%, which is 20 times greater than LoRaWAN. This shows that DiPTC can meet the needs of the application more often than baseline LoRaWAN,

8

even in the case of intensive traffic. Note that the collision rate in DiPTC is 8 time less than in LoRaWAN. Moreover, the network lifetime in DiPTC is considerably longer.

Table 4: Baseline and DiPTC performances.

Sc.	Measures	baseline LoRaWAN	DiPTC
Basic	$(\overline{\mu_c},\sqrt{\sigma_{\mu_c}})$	(32.45%,0.12)	(99.18%,0.24)
	$(\overline{\tau_c},\sqrt{\sigma_{\tau_c}})$	(0%,0)	(0.0%,0.0)
	$(\overline{t_l},\sqrt{\sigma_{t_l}})$	(51823.5h,14.84)	(53024h,54.29)
	$(\overline{E_c},\sqrt{\sigma_{E_c}})$	(1044.97J,2.82)	(3469.76J,140.05)
Intensive	$(\overline{\mu_c},\sqrt{\sigma_{\mu_c}})$	(1.47%,1.23)	(29.64%,0.513)
	$(\overline{\tau_c},\sqrt{\sigma_{\tau_c}})$	(0.96%,0.20)	(0.12%,0.02)
	$(\overline{t_l},\sqrt{\sigma_{t_l}})$	(22563.0min,1082.64)	(34102.0min,50.00)
	$(\overline{E_c},\sqrt{\sigma_{E_c}})$	(7081.54J,80.9)	(10696.34J,24.99)

5.5 Insights

The DiPTC algorithm that we propose shows very good results regarding the convergence time, not only for basic networks, but also for those with an intensive traffic. Thanks to the stability property of the adapting mechanism, our approach manages to converge towards the needed measurements K with a reasonable absolute error. Indeed, DiPTC is able to control the traffic in LPWANs, especially in LoRaWAN, with a low collision probability and only a small overhead. After considering different traffic intensities in our evaluation, we can sum-up the following findings:

1. The reliability of the downlink does not affect the convergence of DiPTC even for low downlink reception probabilities.

2. The death of an active node due to its battery depletion generates a new transient mode of a short duration.

3. The reliability of the uplink has a negative impact on the convergence of DiPTC as it introduces long-term transient states and short-term stationary states.

4. In the BASIC scenario, DiPTC converges exactly towards K measurements per time period. We observe quite long duration for the stationary states compared to the INTENSIVE scenario.

5. In the INTENSIVE scenario, DiPTC converges towards the needed measurements (K) with a reasonable absolute error, but with more frequent and longer transient states.

6. The loss of a transmission during a collision has the same effect as its loss due to the propagation model.

7. Compared to baseline LoRaWAN, DiPTC is able to reach the objective within reasonable deadlines, while keeping a low number of collisions and a low energy cost with a longer network lifetime.

6 Conclusion

In this work, we propose a new distributed and probabilistic algorithm for traffic control in LPWANs. This simple idea, yet powerful, of controlling the traffic using an additive increase multiplicative decrease mechanism and a binary broadcast message sent by the gateway as a feedback information shows very good results. Indeed, DiPTC reduces collision probability and insures convergence to the needs of the application (K measurements per period), within reasonable delays. We also found that compared to the baseline LoRaWAN, DiPTC shows a three times increase in the success rate.

In our future work, we plan to investigate and analyze closely the impact of the traffic adaptation factors x_I and x_D on the equity, the absolute error and the responsiveness of the network, for different traffic and topology scenarios. In fact these parameters affect the duration of the transient states and their oscillation domains, which impact the energy network consumption. Therefore, the values of x_I and x_D introduce a trade-off between the network energy consumption and its responsiveness.

References

[1] A.Azari and C.Cavdar. Self-organized low-power iot networks:a distributed learning approach. In *IEEE GLOBECOM*, Abu Dhabi, UAE, December 2018.

[2] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso. Do LoRa low-power wide-area networks scale? In *IEEE MSWiM*, Malta, November 2016.

[3] A. Boubrima, W. Bechkit, and H. Rivano. On the deployment of wireless sensor networks for air quality mapping: Optimization models and algorithms. *IEEE/ACM Transactions on Networking*, 27(4):1629–1642, 2019.

[4] H. Ennajari, Y. B. Maissa, and S. Mouline. Energy efficient in-network aggregation algorithms in wireless sensor networks: a survey. In *UNet*, Casablanca, Morocco, June 2016.

[5] C. Liu, K. Wu, and M. Tsao. Energy efficient information collection with the ARIMA model in wireless sensor networks. In *IEEE GLOBECOM*, St. Louis, MO, USA, November 2005.

[6] Y. Ma, Y. Guo, X. Tian, and M. Ghanem. Distributed clustering-based aggregation algorithm for spatial correlated sensor networks. *IEEE Sensors Journal*, 11(3):641–648, 2010.

[7] K. Mekki, F. Bajic, E.and Chaxel, and F. Meyer. A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT express*, 5(1):1–7, 2019.

[8] M. Slabicki, G. Premsankar, and M. Di Francesco. Adaptive configuration of LoRa networks for dense IoT deployments. In *IEEE/IFIP NOMS*, Taipei, Taiwan, April 2018.

[9] N. Sornin, Eirich L. M., T. Kramp, and O. Hersent. LoRaWAN specification. *LoRa Alliance*, 2015.

[10] SX1272/73. Semtech datasheet - 860 MHz to 1020 MHz Low Power Long Range Transceiver, rev. 4. January 2019.

[11] Khawam Ta, D. T. and al. Lora-mab: A flexible simulator for decentralized learning resource allocation in iot networks. In *IEEE Wireless and Mobile Networking Conference*, Paris, France, September 2019.

[12] L. Tan and M. Wu. Data reduction in wireless sensor networks: A hierarchical LMS prediction approach. *IEEE Sensors Journal*, 16(6):1708–1715, 2015.

[13] J.Luo Z.Xu and al. S-mac: Achieving high scalability via adaptive scheduling in lpwan. In *IEEE INFOCOM*, Virtual Conference, July 2020.