IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 July 2023, accepted 31 July 2023, date of publication 4 August 2023, date of current version 11 August 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3301991*

RESEARCH ARTICLE

# Adaptive Beacon Period Configurator for Scalable LoRaWAN Downlink Applications

**DAVID TODOLI-FERRANDIS[1], JAVIER SILVESTRE-BLANES[2], VÍCTOR SEMPERE-PAYÁ[3], (Associate Member, IEEE), AND SALVADOR SANTONJA-CLIMENT[1]**

[1]Instituto Tecnológico de Informática (ITI), Paterna, 46980 Valencia, Spain
[2]Departamento de Informática de Sistemas y Computadores (DISCA), ITI, Universitat Politècnica de València (UPV), 46022 Valencia, Spain
[3]Departamento de Comunicaciones (DCOM), ITI, Universitat Politècnica de València (UPV), 46022 Valencia, Spain

Corresponding author: Javier Silvestre-Blanes (jsilves@disca.upv.es)

This work was supported in part by the Horizon Europe Framework Program of the European Commission "AI Powered Human-Centred Robot Interactions for Smart Manufacturing (AI-PRISM)" under Grant 101058589 and in part by the Ministerio de Ciencia e Innovación [Ministry of Science and Innovation (MCIN)]/Agencia Estatal de Investigación [State Research Agency (AEI)]/10.13039/501100011033 and European Regional Development Fund (ERDF) "A way of making Europe" under Grant PID2021-123168NB-I00.

**ABSTRACT** Low-power wide-area networks (LPWAN) are commonly used because they meet the requirements of Internet-of-Things (IoT) networks with a large number of end devices, such as high network scalability, wide area coverage, low data rates, and delay tolerance while consuming very little energy. The LoRa wide-area network (LoRaWAN) is one of the most popular solutions, supporting three types of medium access control (MAC) options to handle distinct application demands. Class B shortens downlink frame transmission latency while maintaining low energy consumption in the end device. This article analyzes the operation of gateways with class B devices to determine the events that influence scalability and performance, presents an analytical model to describe these systems, and proposes an optimization mechanism called Adaptive Beacon Period Configurator (ABPC). ABPC changes the time-related parameters configuration to improve the usability of these networks in dynamic scenarios. The proposed solution is then simulated and tested against the analytical model. The tradeoff between the waiting time between messages, the probability of reception, and the energy consumption of an end device is shown in the results, describing how traffic density increases impacts in these Key Performance Indicators (KPI) and how to try to guarantee these requirements in a network deployment.

**INDEX TERMS** LoRaWAN, low power wide area network (LPWAN), scalability, network optimization.

## I. INTRODUCTION

Information and Communication Technologies (ICT), particularly IoT (Internet of things) related solutions, are playing a fundamental role to increase the efficiency, reduce the environmental impact and simplify and automatize tasks. IoT technologies can be used across all domains and use cases [1]. Innovation in the IoT domain is moving rapidly, so accommodating a fast-growing number of devices requiring wireless connection to a network (as in smart cities, smart metering or digitizing processes for industry 4.0) is a critical topic to address. Such network expansion should not come at the cost of QoS (quality of service) during operation.

The massive amount of network traffic derived from these IoT applications, with large scale deployments entails challenges, as devices struggle for a place in the radio spectrum. A solution oriented to interference immunity is crucial to overcoming IoT scalability issues for a reliable network operation in a saturated spectrum.

LPWANs (Low Power Wide Area Networks) are becoming increasingly important for IoT networks. This is because IoT applications are becoming more and more widespread, and there is a growing need for connectivity that can reach remote devices. LPWANs offer a number of advantages over other wireless technologies, such as cellular networks, for these

The associate editor coordinating the review of this manuscript and approving it for publication was Guangjie Han.

applications. With sub-GHz RF (radio frequency) technology, these networks can be segregated from other 2.4 GHz legacy systems (i.e., Wi-Fi, Bluetooth) to mitigate jitter and congestion issues, while benefiting from other LPWAN traits such as long range, low power consumption and increased noise tolerance [2]. As the IoT industry continues to grow, LPWANs are likely to play an important role. By addressing the challenges of scalability, performance, and energy consumption, LPWANs can become a reliable and cost-effective solution for a wide range of IoT applications.

Many wireless technologies promise to support thousands of devices per network, although frequently those numbers only take uplink traffic into consideration. Downlink traffic is also important, and many wireless technologies are not designed to handle large amounts of downlink traffic. Still, as soon as a growing number of endpoints needs to be integrated, network planning and configuration complexity can quickly rise to the point of being unmanageable. As the network size changes, a manual approach to deploying and operate devices is not usable anymore. For example, a manual approach is not scalable, and it is not possible to manually deploy and manage thousands of devices.

Plus, devices are often deployed at remote or unattended locations technicians rarely visit. For successful IoT scalability, network and device management must be planned from the start, and the network must be provided with tools and mechanisms to perform as required in a wide range of conditions. Features like making data available downwards (allowing devices to receive updates and instructions from the cloud) have an important impact into scalability.

First of all, there is not a one-size-fits-all solution, as the network changes, its configuration needs to change as well. The best configuration for a network will depend on the specific application and the environment in which it is deployed. Scalability supposes a challenge in every type of network, but gains even more relevance when resources are limited, and the number of devices can vary greatly from tens to thousands. Some devices need capacity for more messages than others and this can be hard to plan ahead. This leads to replanning and redeploying, which can be done at the cost of overspending time and resources. Planning ahead for the best-case scenario and setting achievable requirements help reduce the risk of not being able to meet the performance expectations of the deployed networks, and ensuring that the provided solution is flexible enough for any future needs. For example, adding new features to the IoT devices deployed may require exchanging more messages to work properly, so the network should be able to handle the increased traffic. Also, the number of connected devices changes up and down overtime, so a fixed planification is either inefficient or derives in a lack of connectivity, due to packet loss and congestion.

Among the LPWAN technologies, such as Nb-IoT or Sigfox, this article focuses on LoRaWAN which is very popular because of its good performance in range, power consumption, and robustness to noise [3]. LoRa, which is used for the physical layer (Layer 1) coding and modulation methods, operates within the Sub-GHz ISM (Industrial, Scientific and Medical applications) bands. Therefore, this technology is suitable for scenarios where communications in the 2,4 GHz band are banned or congested. LoRaWAN is used for the standardized channel access method and the corresponding system architecture at the link layer (Layer 2) over LoRa.

Although initially LoRaWAN was used primarily for telemetry and other uplink-centered applications, downlink-centered use cases and requirements are increasing, as this technology is becoming popular also for automatization or alternative user interfaces (such as wearables for instance) for those restricted scenarios, as in selected use cases in ZDMP (Zero Defect Manufacturing Platform) [4].

This means that instead of only focusing on end devices pushing data to the network through a gateway (GW), now end devices or nodes (ED in some figures) scattered in the floor-plant are the ones receiving orders, instructions and updates from monitoring platforms. Such a setting structure is very helpful for applications such as smart meters returning data information, and in general, for systems that inherit a polling type operation. For example, the use of class B in the field of telecontrol is the support from relevant parties such as in [5], where an important telecommunications provider enabled this technology for a variety of applications such as asset tracking, smart grid balancing control, air quality sensing, remote patient monitoring or public emergency systems. This presents a critical topic of research [6], as the number of gateways is lower than end devices, which means dividing airtime and resources among all end devices without forgetting to also spend time listening to whatever the end devices have to transmit. Thus, scalability becomes an issue suitable of optimization.

The solution proposed below in this article, composed by an analytical model and the algorithm to reconfigure the network synchronization and scheduling parameters, will allow the gateway to adapt to changes in either network size or application requirements, coming from increasing or decreasing the number of end devices and message rates. This approach enhances the usability of LoRaWAN for application where downlink traffic is important, and where the network suffers from poor performance due to traffic load, delay or high energy consumption that cannot be addressed with standard configurations. The testing results allow to extract conclusions necessary to feed any decision-making algorithms, so once the model is validated it can be used to improve and fine tune the response of the algorithm.

This article first presents related work covering this topic in section II. Then section III covers the LoRaWAN's channel access mode of operation, especially for downlink. This section also presents an analytical model for this downlink channel access and proposes an algorithm for changing and configuring temporal parameters involved in the downlink stage. Then, the proposal is tested via simulation and compared to the analytical model results in section IV.

Finally, section V shows the conclusions and future work inspired by these results.

## II. RELATED WORK

Several surveys and studies cover issues related to LoRaWAN network scalability and usability, such as [7] or [8], in a general approximation to this technology, providing useful information on trends, research challenges such as the scalability considerations introduced in section I, and simulation tools for research. Reference [9] presents a systematic review of literature and works focused on LoRaWAN downlink traffic in the period between 2016 and 2021, comparing various architecture models and their performance metrics, concluding that class B is the more efficient for downlink in an energy-latency trade-off, although scalability is not addressed as downlink messages are considered only as acknowledgements. In [10], researchers study the impact of bidirectional traffic in these types of networks, which as expected is affected by the traffic load and network size, exposing that scalability is a critical problem in LoRaWAN. The authors in [11] propose an automated mechanism for configuring end device radio parameters at run-time that supports multiple application requirements simultaneously and responds to changes in the network, but is intended for uplink applications and would not enhance downlink centric use cases. In [12], the author discusses the poor performance for download traffic and proposes different channel, band, and downlink window schemes for the European 868 MHz spectrum, although it focuses on downlink windows for receiving ACKs in class A devices, this is, downlinks are always initiated by uplinks. A similar study focusing on ACK traffic, surveying proposed methods is shown in [13]. This is another example showing that downlink traffic is an open research topic.

Regarding the application of class B in and downlink traffic in real use cases, authors in [14] and [15] have proposed the use of class B nodes in LoRaWAN to perform monitoring and control of Smart Grids operation. Instead, in [16] authors analyze how to use a LoRaWAN network to integrate with a Wi-SUN (Wireless Smart Ubiquitous Networks) Smart Grid network, which is a popular type of smart metering. One of the critical problems found is the downlink stage as the Wi-SUN network latency cannot manage acknowledged class A reception windows. Authors propose a buffered downlink approach that tries to set longer delays between uplinks and reception windows, whereas that problem could be directly solved by using class B nodes with a proper scheduling. Other interesting and innovative research is found in [17], which proposes using backscatter communication (battery-less data transmission where devices transmit data after harvesting energy from received signals from an RF-source gateway), based on LoRaWAN devices and network. Leaving beside the energy operation considerations, class B mode could enable this type of communication, by ensuring a periodical energy supply thanks to gateway downlink messages.

Focusing on the class B and other planification solutions, authors in [18] review how time-slotted solutions can address the aforementioned problems, with some considerations and open challenges. Scalability and collision free communications are some of the proposed areas requiring further research, and they are topics covered in the present article. Then in [19] and [20], LoRaWAN class B mode of operation, which is indeed a time-slotted solution, is simulated to evaluate the scalability of the network and gateways, finding that most packet loss comes from dropped packets due gateway's duty cycle limits. The article in [21] proposed a lightweight scheduling solution for LoRaWAN, but addresses overall traffic and requires the implementation of their own MAC (medium access control) layer called RS-LoRA, which would not comply with other devices running legacy standard. In [22] authors propose LoRaSync, which leverages class B beacons to provide synchronization, in order to setup a Slotted ALOHA access for uplink, instead of the legacy pure ALOHA scheme. Therefore, although using class B, the final aim of the article is optimizing uplink traffic.

In [23], authors propose a solution based on multiple gateways and optimal selection of which gateway transmits downlinks, in the fashion of load balancing. But it focusses mostly on acknowledgements for uplink or downlink messages, which is obviously an overload, and also cannot be applied to networks with a single gateway, which are not uncommon.

In [24], researchers propose an analytical model of class B mode and a trade-off function to try to optimize the slotted access. Some assumptions like the spreading factor (SF) used by nodes on downlink is not realistic, traffic load in simulations is also restricted and no duty cycle limitations are considered. Without introducing any changes to the standard, the results are limited and cannot address to a wider range of possible applications and scenarios. The authors in [25] provide a complete analysis of class B and study the impact of tuning some parameters. Nevertheless, the study does not consider individual slot length optimization. Otherwise, blank portions of the scheduled resources can be left unallocated which can impact greatly the scalability of the network.

The solution proposed in the following sections offers a model for the gateway buffer in class B operation fitting duty cycle regulations, and provides a mechanism that enables an adaptive slotted window configuration. This proposal includes the modification of the standard class B parameters not only regarding the slots enabled by each end device, but also the periodicity of the scheduled window, and optimization of slot duration for such non-standard configurations, which is a novel approach to class B. The proposal is based on the LoRaWAN MAC, which offers compatibility with other devices in the same network that may function with other LoRaWAN configurations. The study has been performed under realistic conditions that include duty cycle limitations for European regulations, but can be adapted to other regions. This is crucial as duty cycle limitation has been identified as the primary cause of losses. These limitations become more severe as the network grows, and the solution provided offers the possibility to adapt to new conditions by reaching a compromise between different QoS parameters. As the designed algorithm changes the periodicity of the scheduled window, other time related parameters need to be adapted. The following sections show a solution that considers the optimization of pingslot length parameter, which is a novel approach in LoRaWAN class B research, and how it used to optimize slot usage in variable beacon period sizes. In this way, the network performance can fulfill different requirements for different applications and network sizes deployed, in a changing environment such as industry.

## III. MATERIALS AND METHODS

### A. TECHNOLOGY OVERVIEW AND OPERATION IN LoRaWAN CLASS B MODE

At the physical layer, the radio transceivers of the end devices use Semtech's Long Range (LoRa) technology [26]. With this, wireless low power transmitters can forward small packets of data to a receiver, over long distances up to several kilometers depending on the environment, in a point-to-point link. LoRa modulation is based on Chirp Spread Spectrum (CSS), using linear frequency modulation chirp pulses with high bandwidth to encode information. These chirp pulses are sinusoidal signals shifting in frequency over time, which determines the symbols that represent the information. The number of values that can be encoded in each symbol is given by the spreading factor (SF). The range of SF values admitted is between 7 and 12.

The larger the SF used, the further the signal will be able to travel and still be received without errors by the receiver. The duration $T_s$(in seconds) of a symbol can be calculated as shown in (1), which also shows the data rate $R_b$. The bandwidth (BW) as set by the standard can be 125, 250 or even 500 kHz, and it is set to a fixed value for all network's lifetime. The code rate (CR), as shown in (1), is used for error correction in wireless communications by introducing redundant bits, with $1 < n < 4$, although in LoRaWAN standard the value set for CR is 4/5 (n=1) in order to reduce the overhead:

$$T_s = \frac{2^{SF}}{BW}, CR = \frac{4}{4+n}, R_b = SF \cdot \frac{CR}{T_s} \tag{1}$$

Derived from (1), increasing the SF value means lowering the bit rate and therefore, increasing the Time on Air (ToA) of a packet (as each symbol of the message has higher $T_s$). This also reflects in the power consumption of the device, as it needs to enable the radio interface for longer periods to send the data. On the other hand, the coverage range increases for higher SFs values as the energy of the total signal is now spread over a wider range of frequencies, allowing the receiver to decode a signal with a worse signal-to-noise ratio (SNR).

Working over LoRa technology, LoRaWAN is an open protocol defined by the LoRa Alliance, and is supported by a central Network server that orchestrates all the devices (end nodes and gateways, which form star topologies) of the network (for instance, selecting the best gateway for a node). On top of these layers, the LoRaWAN architecture relies on applications servers to relay the information to other systems and networks, supporting different application protocols, as shown in Fig. 1. The latest version of the protocol released by the LoRa Alliance is version 1.1 [27]. The gateways are connected to the network server through a conventional TCP/IP (Transmission Control Protocol/Internet Protocol) SSL (Secure Sockets Layer) network, while the end devices use LoRa to communicate with one or more gateways.

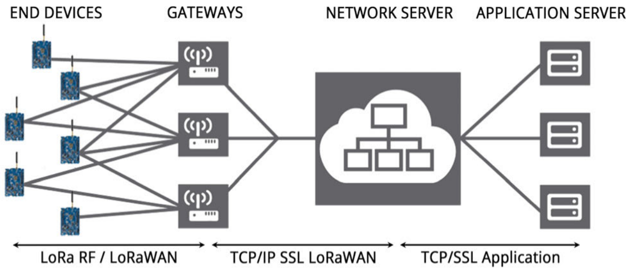


**FIGURE 1.** LoRaWAN network architecture.

**TABLE 1.** Duty cycle restrictions on EU bands.

| Name | Band (MHz) | Limitations |
|---|---|---|
| G | 863,0-868,0 MHz | EIRP < 25mW – duty cycle < 1% |
| G1 | 868,0-868,6 MHz | EIRP < 25mW – duty cycle < 1% |
| G2 | 868,7-869,2 MHz | EIRP < 25mW – duty cycle < 0.1% |
| G3 | 869,4-869,65 MHz | EIRP < 500mW – duty cycle < 10% |
| G4 | 869,7-870,0 MHz | EIRP < 25mW – duty cycle < 1% |

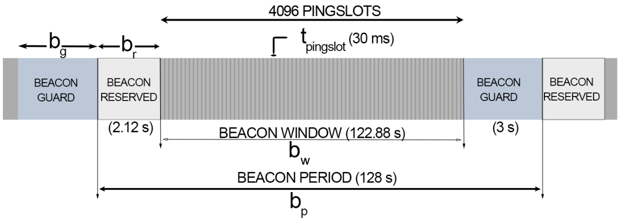


**FIGURE 2.** LoRaWAN Class B operation mode time parameters.

Another matter to keep in mind when sending and receiving messages in a LoRaWAN network is to comply with spectrum regulations. The Duty Cycle (***DC***), which is the percentage of time a device is using or occupying the channel, is regulated in Europe as seen in Table 1, or in detail in [28].

The default transmission configuration recommended for the downlink is 869.525MHz (G3 sub-band offering the highest ***DC***) with SF9 and 125kHz bandwidth [27]. Nevertheless, the default channel can be modified by gateways using MAC commands in accordance with the LoRaWAN Regional Parameters.

Data communication in Class B mode is based on the beacon period, which is set to 128 s. One beacon period, as illustrated in Fig. 2, consists of beacon reserved (2.12 s), beacon guard (3 s), and beacon window (122.88 s) [27]. When the beacon reserved interval expires, the beacon window interval begins, and each end device opens pingslots with a ping period inside this interval periodically to receive downlink data.

The beacon guard period is adjusted to 3 s, in order to avoid that the gateway is still sending downlink messages when it should send beacons. The beacon window is divided into 4096 pingslots with a duration of 30 ms. The number of pingslots (***pingNb***) available during a Beacon Window can be derived from the periodicity parameter, referred to as ***k*** for the rest of this article. It is fixed by the standard as in (2), therefore setting the number of pingslots usable by a node in each Beacon Period from 1 to 128 pingslots:

$$pingNb = 2^{7-k}\ where\ 0 \le k \le 7 \tag{2}$$

The ping period is the delay between two pingslots, and is constant, calculated as:

$$pingPeriod = \frac{4096}{2^k} slots \tag{3}$$

The offset between the beacon and the first pingslot is pseudo-randomly chosen after each beacon for each end device, based on the pingPeriod, device address and joining keys, and the last beacon time received:

$$Rand = aes128encrypt(Key, beaconTime|DevAddr |pad16) \tag{4}$$

$$\text{pingOffset} = (\text{Rand}[0]+\text{Rand}[1] \times 256)\text{mod}(\text{pingPeriod}) \tag{5}$$

Having a scheduled channel access implies that nodes have specific slots assigned, which eliminates the possibility of collisions with other nodes, but it also means that there are a limited number of available pingslots, which may be an issue as the network grows. Several events when a Gateway tries to transmit to class B nodes can alter the expected operation:

- The pingslot chosen has already been given to another node when the number of nodes exceeds the number of available pingslots, as shown in Fig. 3 in case A
- The pingslot assigned is already in use (due to the preceding downlink broadcast, or because there is an ongoing uplink message), as seen in Fig. 3 in case B.
- The allocated pingslot is too near to the preceding active one (even if it has completed) and cannot be utilized in order to comply with the duty cycle, as seen in Fig. 3 case C.

Please note that the time taken for the transmission of a message is dependent on its size, which translates in a given ***ToA*** that is always longer than the 30 ms of a pingslot. Having a pingslot assigned only marks the starting time of the transmission. This directly causes case B, and also impacts on the duty cycle reserved time in case C (the longer the message, the longer the waiting time).

**TABLE 2. Model related definitions.**

| Parameter | Description |
|---|---|
| $\boldsymbol{k}$ | Periodicity or pingslot parameter |
| $\boldsymbol{N}$ | Number of devices receiving messages in network |
| ***pingPeriod*** | Ping period for a given ***k*** |
| $\boldsymbol{b_w}$ | Beacon window size in ***s*** |
| $\boldsymbol{\lambda}$ | Interarrival rate of messages, following constant distribution of value ($\boldsymbol{b_w / N}$) |
| ***ToA*** | Time on air needed for reception of a frame, dependent on ***SF*** and packet size |
| $\boldsymbol{T_{DC}}$ | Duty cycle reserved time, value 9·***ToA*** (for 10% duty cycle) |
| $\boldsymbol{\mu}$ | Service rate, Uniform distribution variable in [***ToA, 10·ToA***] |
| $\boldsymbol{s}$ | Number of servers, in this case only 1 |
| $\boldsymbol{p_b}$ | Probability of busy channel |
| $\boldsymbol{\rho}$ | System utilization factor = l/sμ |
| $\boldsymbol{\gamma}$ | Traffic intensity = l/μ |
| $\boldsymbol{P_b}$ | Probability that the server is busy when a packet arrives |
| $\boldsymbol{P_0}$ | Probability that the server is idle or available |
| $\boldsymbol{L_q}$ | Number of packets in queue in G/G/s model |
| $\boldsymbol{L_{q\_MMS}}$ | Number of packets in queue in M/M/s model |

### B. SYSTEM MODEL

In the following analytical model proposed, and given the definitions shown in Table 2, assume that ***N*** uniformly distributed, Class B mode, end devices are connected to a gateway. Also, for the ongoing analysis, assume that the gateway receives data to distribute to end devices uniformly for each device (all devices configured with same ***pingNb*** and ***k***). This means the arrival rate to the system, λ, is deterministic and can be considered (***N***· ***pingPeriod***). Let's suppose that any packet that is not served during its beacon window is discarded, then the server (in this case the gateway) is represented by a system with finite capacity, in this case 1, as packets are discarded if medium is busy. The system model diagram is shown in Fig. 4.

It is also assumed that uplink data from devices, which could prevent the gateway from transmitting due to its radio being busy with reception, is negligible in this case as application is downlink centric only (sporadic uplinks in real world). It can be modelled as a D/G/1/2 system with a FIFO queue discipline.

End devices are configured for reception with the predefined parameters for downlink in class B, this is, SF9 and 125 kHz bandwidth, which implies a maximum packet size of 128 Bytes (including headers). This is relevant to calculate the ***ToA*** and therefore the duty cycle limitations, both variables affecting the pingslots that are truly available for downlink at a given moment. For the selected sub-band g3, duty cycle is set to 10%, which means the time required between the start of two consecutive frames to comply with regulations is 10·***ToA*** (as a 10% ***DC*** means leaving the channel free 90% of the time).

This means that regarding the service time $\mu$, it is important to note that the frame at the very front of the queue cannot be

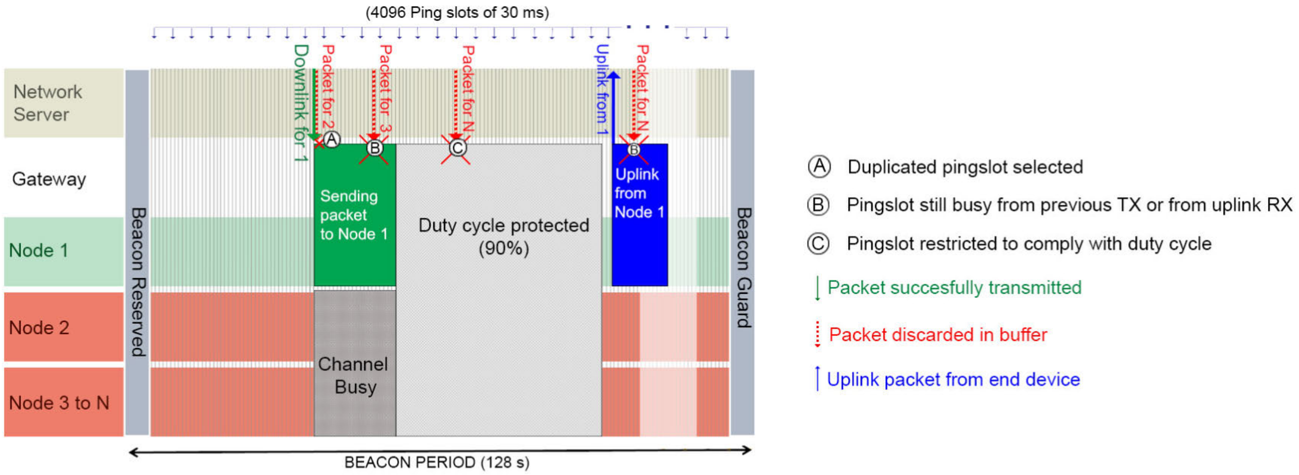

(4096 Ping slots of 30 ms)
Network Server
Gateway
Node 1
Node 2
Node 3 to N
Beacon Reserved
Downlink for 1
Packet for 2
Packet for 3
Packet for N
Uplink from 1
Packet for N
A
B
C
B
Sending packet to Node 1
Duty cycle protected (90%)
Channel Busy
Uplink from Node 1
Beacon Guard
BEACON PERIOD (128 s)
Ⓐ Duplicated pingslot selected
Ⓑ Pingslot still busy from previous TX or from uplink RX
Ⓒ Pingslot restricted to comply with duty cycle
Packet succesfully transmitted
Packet discarded in buffer
Uplink packet from end device

**FIGURE 3.** Possible events affecting downlink transmissions.

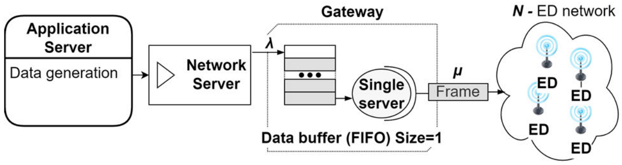


**FIGURE 4.** System model diagram.

served immediately after the previous frame transmission is completed due to the waiting time introduced by duty cycle. All packets arriving when server is busy are immediately discarded. The rest of the model descriptions and used parameters are shown in Table 2. This model is used to check results of the simulator in the results section.

However, this system classified as D/G/1/2 does not have a closed form formula to solve it. An approximation to a G/G/s model can be used, or it can be resolved by Monte Carlo discrete event solving methods [29], which is the solution used for the results section. For the approximation, the solving equations are as follows:

$$L_q = L_{q\_MMS} \cdot \frac{\mu^2 V(t) + V(t')\lambda^2}{2} \tag{6}$$

where $V(t)$ is the variance of service time and $V(t')$ is the variance of interarrival time.

The rest of measures are the same as in a M/M/s system, knowing that in this case $\boldsymbol{s}$=1, so $\gamma = \rho$. To complete (6) the equations of this system are shown in (7)-(9), where $\boldsymbol{n}$ is the packets in the system.

$$P_0 = \frac{1}{\sum_{n=0}^{s-1} \frac{\gamma^n}{n!} + \frac{\gamma^s}{s!(1-\rho)}} = \frac{1}{1+\gamma/(1-\gamma)} \tag{7}$$

$$P(n) = \begin{cases} \frac{P(n-1)\gamma}{n} n \leq 1 \\ P(n-1)\gamma n > 1 \end{cases} \tag{8}$$

$$L_{q\_MMS} = \frac{P_0 \gamma^s \rho}{s!(1-\rho)^2} = \frac{P_0 \gamma^2}{(1-\gamma)^2} \tag{9}$$

And therefore, the probability of finding the server busy, meaning the packet is dropped and lost, is:

$$P_b = \sum P(n)\, n \geq 1 \tag{10}$$

### C. ADAPTIVE BEACON PERIOD CONFIGURATOR PROPOSED ALGORITHM

The performance of LoRaWAN's Class B mode is mostly influenced by the pingslot configuration, which is critical in balancing the downlink packet transmission delay and energy consumption. With the Beacon Period and pingslot scheduling implementation defined by the standard, there is a limit on what trade-off configurations can be adopted to address different situations and requirements. For a rapidly increasing network with a high number of nodes that require downlink transmissions, the traffic intensity can rise enough to make delay to not meet acceptable values.

On the contrary, the network's devices number may decrease (or be low from the start). Based on given restrictions, it can be of interest to provide a mechanism that changes the Beacon Period time $\boldsymbol{b_p}$ (and therefore the beacon window $\boldsymbol{b_w}$ too) parameters according to the size of the network and the traffic load. To enable different Beacon Period configurations from the standard's proposal, the $\boldsymbol{k}$ parameter also needs to be set accordingly, and not only in $0 \leq k \leq 7$ interval.

Regarding the pingslot configuration, the adopted convention is a duration of 30 ms, which is calculated as the minimum time needed to detect the preamble of a transmitted frame, regardless of the ***SF*** (and therefore the ***ToA*** of the frame). So, trying to force a shorter duration for a pingslot is not possible, as receivers would fail to detect incoming messages. Also, the duration of the pingslots relate to another key parameter: the number of pingslots in a beacon window. As derived from Fig. 2 or Fig. 5, pingslots of 30 ms in 122.88 s

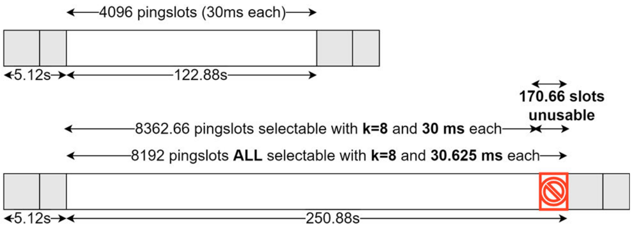

4096 pingslots (30ms each)
5.12s
122.88s
170.66 slots unusable
8362.66 pingslots selectable with **k=8** and **30 ms** each
8192 pingslots **ALL** selectable with **k=8** and **30.625 ms** each
5.12s
250.88s

**FIGURE 5.** **Pingslot duration selection based on periodicity *k*.**

**TABLE 3.** **Pingslot duration according to beacon period times.**

| Beacon Period | $t_{pingslot}$ | $n_{pingslot}$ | $k_{max}$ |
|---|---|---|---|
| 64 s | 57.5 ms (28,75 ms) | 1024 (1962,66) | 5 (6) |
| 128 s * | 30 ms | 4096 | 7 |
| 256 s | 30.625 ms | 8192 (8362.66) | 8 |
| 512 s | 31.094 ms | 16384 (16896) | 9 |
| 1024 s | 31.095 ms | 32768 (43133.33) | 10 |

*128 s corresponds to standard LoRaWAN class B parameters

results in the 4096 pingslots that can be, a priori, used. This value, along with the periodicity $\boldsymbol{k}$, is used to calculate the ***pingPeriod***, influencing, as seen in the model in (6)-(10), the loss/success probability of the system.

So, a first consideration when proposing changing the Beacon Period, is considering the effect of pingslot duration if the window duration is different.

Assuming the beacon guard and beacon reserved fields to remain fixed (3+2.12 s), as they are calculated to ensure the reception of broadcasted beacons in every end device, if the Beacon Period is doubled, to 256 s in total, that would mean the effective time $\boldsymbol{b}_w$ for pingslots is 250.88 s, which results in 8362.66 slots of 30 ms.

Furthermore, given that the addressable (as in which are really used) pingslots, coming from equations (2) and (3), are powers of two, so the closer configuration would be, with a $\boldsymbol{k}$=8, having 8192 pingslots, so 170.66 pingslots would be wasted, reducing the theorical capacity to send downlink frames. This is better seen in Fig. 5.

Based on this information, and to keep the optimal pingslot duration to maintain the capacity of the gateway, the first step to implement a variable Beacon Period is to provide a solution to select pingslot duration according to maximum periodicity $\boldsymbol{k}_{max}$ available and Beacon Period length:

$$t_{pingslot} = \frac{b_p - b_g - b_r}{2^{k_{max}+5}} \tag{11}$$

where $\boldsymbol{k}_{max}$ value is not fixed as 7 as proposed by the standard and described in (2) but increases or decreases by 1 unit if the Beacon Period is doubled or halved, respectively. The parameters $\boldsymbol{b}_p$, $\boldsymbol{b}_g$ and $\boldsymbol{b}_r$ can be seen in Fig. 2.

Table 3 shows values of $\boldsymbol{t}_{pingslot}$ for different Beacon Period sizes. Accordingly, pingPeriod now is not defined by the 4096 pingslot, but with the number of available pingslots ($\boldsymbol{n}_{pingslot}$) according to $\boldsymbol{k}_{max}$. Between parenthesis in the $\boldsymbol{n}_{pingslot}$ column it is shown the number of pingslots for a fixed pingslot length of 30 ms.

Note that if the Beacon Period falls below the 128 s indicated in the standard, $\boldsymbol{t}_{pingslot}$ would fall below the minimum of 30 ms required, and therefore $\boldsymbol{k}_{max}$ needs to be decreased further to comply. At that point, pingslot is almost doubled to 57.5 ms, which is in fact less efficient from the energy point of view. Nevertheless, using a configuration with such parameters can only be oriented to applications with a very low number of end devices which require shorter delays, so probably energy consumption has no weight in the decision. On the other hand, higher values of Beacon Period do not impact significantly in the duration of the pingslot but enable more pingslot resources for networks with increasing number of end devices, provided delay is not a concern within some limits. Therefore, these longer beacon periods are oriented for big networks without strict time requirements, but can benefit from energy savings, as they spend less time listening to the channel (the number of beacons is halved with each period doubling).

The proposed algorithm called **Adaptive Beacon Period Configurator** (ABPC), configures the Beacon Period size, the $\boldsymbol{k}_{max}$ periodicity and the pingslot period $\boldsymbol{t}_{pingslot}$ according to the number of devices (or traffic load). This solution aims to provide a better network response in situations where scalability, delay, or power consumption may pose a problem, considering that optimizing for specific requirements come at a cost or trade-off, as maximizing a given parameter directly affects the others.

Therefore, for this ABPC set up some parameters are fixed and predefined, in this case the size of the downlink messages payload (which could be also changed during runtime if needed), while the rest of parameters will change to adapt to the scenario and the running application, as intended by the mechanism. In order to tune the mechanism based on results, the size of the network to simulate is chosen upon the assumption that during a Beacon Period there is a theoretical maximum number of pingslots used to send packets successfully, $\boldsymbol{S}_{max}$ (maximum usable slots), taking only duty cycle into account, without considering the random slot selection at the start of each window, as seen in Table 4. Simulation is based on the libraries for ns3 of [30] with corresponding changes and modifications for the described ABPC.

$\boldsymbol{S}_{max}$ is used as input to select the number of nodes in the network during simulation, and as the triggering parameter for the ABPC to change configurations, in order to compare with the model. It is assumed that nodes are configured to receive one packet per Beacon Period ($\boldsymbol{b}_p$), which corresponds to one pingslot. Therefore, the input for the algorithm at this stage, is assumed to be $\boldsymbol{S}_{max}$. The value of $\boldsymbol{S}_{max}$ can be calculated according to (12), knowing the payload size of the packets and SF (to obtain the ***ToA***), duty cycle, and Beacon Period Size configured (which reflects in the beacon window $\boldsymbol{b}_w$):

$$S_{max} = \frac{b_w}{ToA + T_{DC}} \tag{12}$$

**TABLE 4. Design considerations based on payload size, duty cycle and beacon period.**

| Beacon Period | $b_w$ (s) | Payload | $ToA$ (ms) | $T_{DC}$ (s) | Max Usable Slots ($S_{max}$) |
|---|---|---|---|---|---|
| W64 | 58.88 | 50 bytes | 369.7 | 3.327 | 16 |
| W128* | 120.88 | | | | 32 |
| W256 | 250.88 | | | | 68 |
| W512 | 504.88 | | | | 136 |
| W1024 | 1018.88 | | | | 276 |

*W128 corresponds to standard LoRaWAN class B parameters

The values obtained allow to validate the analytical model presented in previous section III-B, which then can be used to train this decision-making element of the ABPC in future implementations. The gateway knows how many devices are connected and requiring downlink messages, and has predefined rules, depending on the application, to maintain certain levels of QoS, such as Packet Delivery Ratio (PDR) and energy, so the model can determine if there is a need to change the Beacon Period Size for those requirements. The gateway then updates the synchronization configuration in the nodes via the beacon packets at the start of each Beacon Period. This algorithm is summarised in the flow chart in Fig. 6.

### D. VALIDATION SCENARIO

The scenario proposed for tests and validation is set to an industrial floor plant of radius 2km and a realistic propagation channel model as introduced in [31], with a variable number of end devices ranging from 4 to 1104, depending on the selected network size due to the design considerations. This number of end devices is realistic for an industrial application and is derived from proportional values of $S_{max}$. Note that $S_{max}$ values shown in Table 4 are not always powers of 2 as the Beacon Period, so for instance a value of 4 times $S_{max}$ for the **W1024** results in 1104 nodes.

Higher values of Beacon Periods and Smax are not considered at this time because they would introduce to much waiting time between periods, but it could be explored in the future for delay tolerant network without any time restrictions.

The application server is generating downlink messages with a rate of 1 packet per beacon window per node (whichever the length of the Beacon Period), while end devices only put uplink messages (for uplink devices behave like class A) every hour. Table 5 shows a summary of used configurations.

In the scenario simulated, the number of nodes is fixed for 24 hours, after which the network size is increased (see parameter $N$ in table 5), which triggers the ABPC and reconfiguration of class B parameters, repeating the cycle to achieve 24h of statistics for each configuration.

To calculate the different energy consumption in the different configurations, a value of 11mA for radio consumption during receiving/listening time has been used, as extracted from Semtech chipsets datasheets. The real consumption on a device should also consider other elements such as Central Processing Unit (CPU), sensors, memory, etc. But those parameters are hardware and application related and are not impacted by the proposed solution which relates only to radio aspects. For reference, an example of battery lifetime is calculated assuming a common battery configuration of 3000 mAh

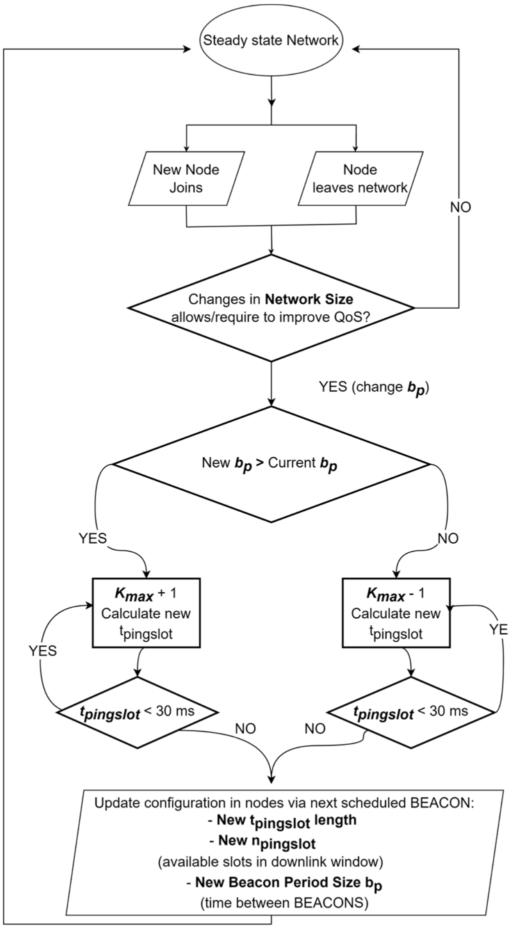

Steady state Network
New Node Joins
Node leaves network
NO
Changes in **Network Size** allows/require to improve QoS?
YES (change $b_p$)
New $b_p$ > Current $b_p$
YES
NO
$K_{max}$ + 1
Calculate new $t_{pingslot}$
$K_{max}$ - 1
Calculate new $t_{pingslot}$
YES
YES
$t_{pingslot}$ < 30 ms
$t_{pingslot}$ < 30 ms
NO
NO
Update configuration in nodes via next scheduled BEACON:
- New $t_{pingslot}$ length
- New $n_{pingslot}$
(available slots in downlink window)
- New Beacon Period Size $b_p$
(time between BEACONS)

**FIGURE 6. Flow chart diagram of the ABPC mechanism process.**

**TABLE 5. Scenario simulation configurations.**

| Parameter | Description | Value |
|---|---|---|
| $N$ | Number of devices | $S_{max}/4 \le N \le 4\, S_{max}$ |
| $k_{max}$ | Max. Periodicity | $5 \le k_{max} \le 10$ |
| $k$ | Periodicity | $0 \le k \le k_{max}$ |
| *pingPeriod* | Ping period for a given k | $n_{pingslot}/\ 2^k$ |
| *Beacon Period* | Duration of Beacon Period | [64, 128, 256, 512, 1024] |
| $\lambda$ | Arrival rate | $N \cdot pingPeriod$ |
| $SF$ | Spreading Factor downlink | 9 |
| $M_{down\_payload}$ | Message size | 50 Bytes |
| $ToA$ | Time on air | 369.7 ms |
| $DC$ | Duty cycle limit in g3 sub-band | 10% |

**TABLE 6.** Power related parameters.

| Parameter | Description | Value |
|---|---|---|
| $I_{TX/RX}$ | Radio current consumption during receiving/listening time | 11 mA |
| $I_{deep\_sleep}$ | CPU current consumption in deep sleep | 7 µA |
| $I_{CPU}$ | CPU current consumption active (during radio operation) | 50 mA |
| $ToA$ | Time on air to receive a normal message | 369.7 ms |
| $ToA_{beacon}$ | Time on air to receive beacons | 160 ms |
| $ToA_{empty}$ | Time on air listening for empty slots | 3.697 s |

(two AA alkaline batteries for instance), and an ESP32 CPU with the SX1276 LoRa radio chipset. For reference, the $ToA$ related are 369.7 ms for 50 Bytes data packets, 160 ms for beacons, and 30 ms for empty pingslots. A summary of power related parameters is shown in Table 6.

## IV. RESULTS AND DISCUSSION

To validate the ABPC solution proposed, the selected scenario is simulated under ns3, and compared to the analytical model. To represent the different configurations according to Beacon Period, figures' captions show a naming convention of W64 to W1024 as in Table 4. It is also relevant for the analysis of the results to highlight that the W128 configuration corresponds to the standard's proposition, so that is also a key configuration in the comparison.

First, a detailed analysis of event in the gateway buffer is presented, characterizing the operation of this element for the different configurations, showing **PDR** (packet delivery ratio, which is the percentage of received messages versus the expected, therefore the probability of successful transmission), and the **PLR** (packet loss ratio) due to the different type of events.

Then, on the End devices or nodes' part, results in terms of delay between consecutive downlink packets, PDR, and power consumption are obtained. The number of devices used is shown as in results as multiples of N, where $N$=$S_{max}$, and $S_{max}$ values can be observed in table 4. This is important when reading the figures, as values of $N$ in fact are different between Beacon Period configurations.

From the gateway's perspective, as shown in Fig. 7, the main problem detected in all cases is that packets are missing their transmission pingslots due to duty cycle limitations, followed by finding the scheduled pingslot being still in use by other message transmission (either downlink or uplink).

This can be seen as the series labeled as 'DC PLR', which describe packets discarded at gateway's buffer, are the more significant and grow considerably with the number of devices. Bars with dotted filling and colored border are PDR, while PLR series are solid color bars.

With regards to configurations, longer Beacon Periods suffer less from the duty cycle and occupied slot events, achieving a higher percentage of packets successfully transmitted, for the same amount of end nodes. See in Fig. 7 the highlighted values for number of nodes are 16 or 552, as $N$ depends on $S_{max}$(in those cases $S_{max}$ =16 or 276, see table 4 for relation with Period Beacon sizes) as example. The values of PDR for the analytical model, which are correspondent to the percentage of transmitted messages, is also shown in Fig. 7, so it can be seen that the model describes well the expected results.

As the model inputs are related to the traffic injected to the gateway, and this is related to the Beacon Period size, statistically the model only depends on the interarrival time of packets to the gateway and thus is dependent on $N$ but independent of beacon window size.

Regarding the nodes, increasing the Beacon Period size and the ping period increases the waiting time or delay, but it is worthwhile noting that this increase does not seem to be directly proportional (doubling the Beacon Period size does not automatically double the time between packets). This applies also to the number of packets received successfully on the nodes, as the results show that even when reducing the message generation rate (increasing ***pingNb***).

This supposes that for a Beacon Period of 256 s (W256), the gateway is trying to transmit half the amount of packets than for a Beacon Period of 128 s (W128), but in the node's end this does not translate in half the messages. It depends on the size of the network, as when the number of nodes grows, the delay between messages at nodes' end equalizes, as does the number of messages received. But on the other hand, the power consumption becomes lower with longer beacons windows.

In Fig. 8 it can seen how the delay increases and the PDR (using the values of the model) decreases when rising the number of devices. This increase in delay is all the more pronounced when W is higher, being almost negligible for W64, and very pronounced for W1024.

Regarding power consumption, the mean current $I_m$ per hour after 1 day has been obtained. The reason to use current is that voltage is a fixed value (typically 3.3V or 5V for battery powered devices). This can be seen as batteries show their capacity in mAh too. The calculation considers the number of packets received by each node, the time it takes to receive them, and the same for beacons. Then the rest of consumption comes from nodes enabling radio reception during the minimum time to listen to preambles and turning off if no data present in the channel.

Increasing the Beacon Period size automatically reduces the number of transmitted synchronization beacons, which means reducing the time an end device is expected to be listening, therefore it has a direct impact on power savings. As can be seen in Table 3, window sizes above (or below) 128 s result in longer pingslot durations higher than 30 ms. Considering the real minimum time needed, regardless of window sizes or $SF$, to detect the preamble of an incoming frame is 30 ms, from the point of view of the end device's radio, that is the amount of time it has to be listening. The altered duration of the pingslot length is relevant for synchronization so the gateway and end devices know when to start transmitting/receiving, but minimum active radio time

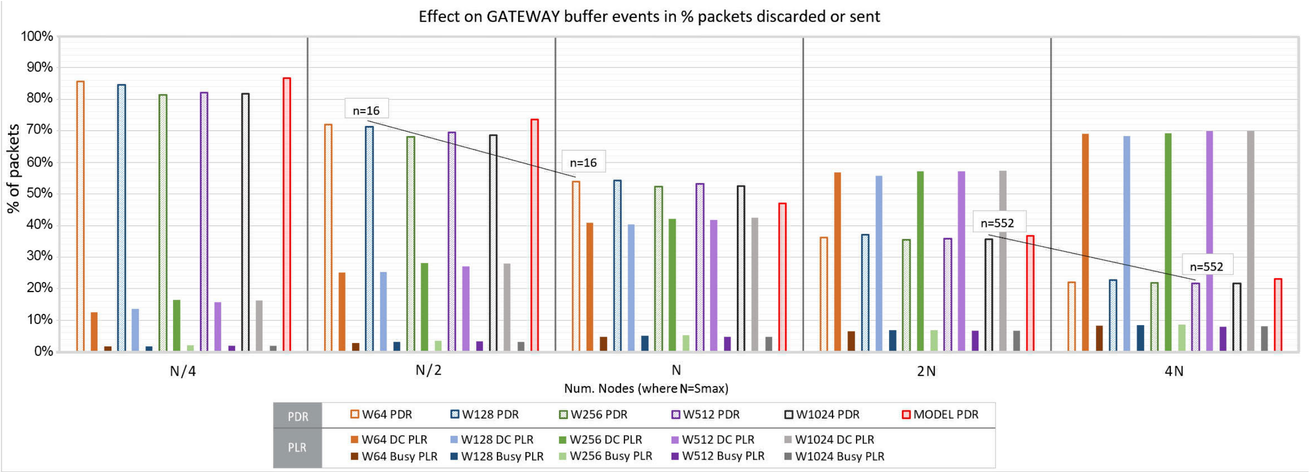


**FIGURE 7.** Impact of different event in the gateway buffer depending on window size and network size.

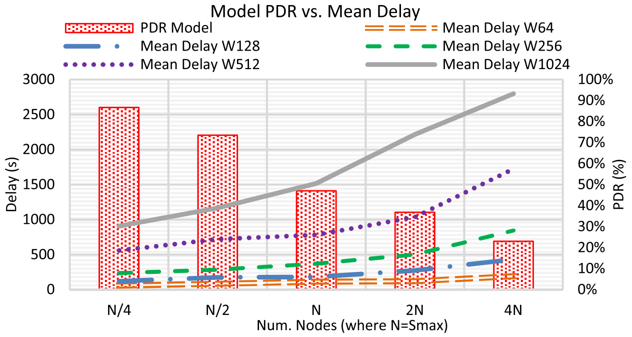


**FIGURE 8.** Mean network values for PDR and delay.

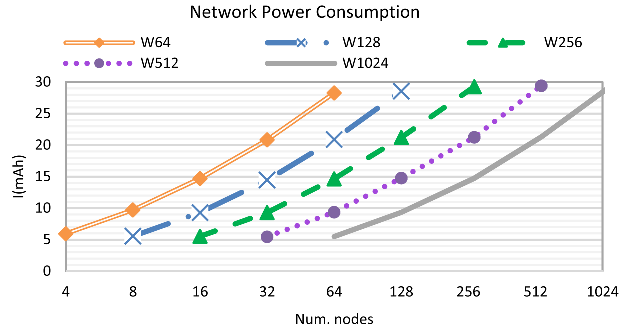


**FIGURE 9.** Overall network consumption for different combinations of network size and Beacon Period configurations.

when pingslot starts can be considered fixed in 30 ms for all configurations. Therefore, it does not impact negatively in power consumption.

On the other hand, increasing (or decreasing) the periodicity ($\boldsymbol{k}$) means each node wakes up more or less frequently, so longer periods that enable higher periodicity values achieve (at the cost of higher delays) longer battery life. This can be seen in Fig. 9 and 10.

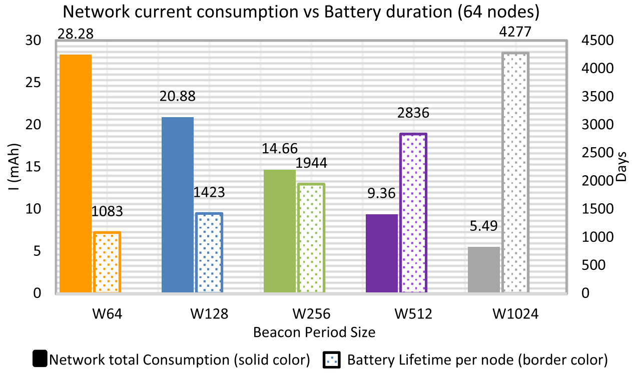


**FIGURE 10.** Network consumption for 64 nodes and battery lifetime per node depending on the Beacon Period size.

Finally, Fig. 10, which is obtained from a network with 64 nodes as example, also shows the duration of the batteries (3000mAh) on the hardware described in section III-D. This calculation assumes the CPU is in deep sleep state while radio is idle and activates when it is time to receive (depends on Beacon window size) or send messages (fixed to one per hour). The results show how relevant it can be to change Beacon Periods, if possible, as nodes can benefit from up to 200% more days of operation than with the standard. This battery duration drastically drops if hardware does not support deep sleep, but it is a feature that most manufacturers include in their products nowadays.

## V. CONCLUSION AND FUTURE WORK

The introduction of class B mode in LoRaWAN to support downlink messaging without needing end devices to trigger it and the rising interest in using these networks in industrial applications has shown that there are still optimization and scalability problems that can benefit from more flexible scheduling approaches. The results obtained from testing different configurations of Beacon Period sizes and periodicities, supported by the pingslot duration adaptation mechanism described, show that the proposed parameters by the standard can be further optimized according to the network size and the specifications or quality of service required by the application.

On one hand network size can impact greatly the performance of the network, especially for LoRaWAN class B and its scheduled operation, so in cases where scalability may be an issue it is worthwhile having the chance to adapt configurations to enhance network operation and maintain a certain level of PDR despite increasing delay.

On the other hand, other cases feature a reduced number of end devices, and it may be of interest to favor other key parameters exploiting that more resources in the gateway are available.

From the results, it can be seen that using smaller Beacon Period sizes benefits the delay between messages, and logically the number of total messages received. These parameters' tendency equalizes as the network grows, which means that for very large networks, delay and messages received are no longer directly influenced by the Beacon Period size unless this is also greatly increased, to the point that the network is useless because delay and packet loss are too high. The other relevant aspects, such as power consumption and PDR, follow the inverse logic. If the key characteristics required are longer battery life in end devices, then longer Beacon Periods achieve much better results, considering that large networks feature similar mean levels of delay and messages received per node.

Thanks to the model and the curves obtained for PDR, delay and energy, the gateway can infer in advance the behavior of the network according to end devices connected, and take well founded decisions on Beacon Period size configurations or even in blocking new devices to guarantee predefined requirements. The point where the adaptiveness is no longer beneficial can be calculated with these data and algorithm, and checked against user's requirements. For instance, a delay of one hour may be reasonable for configuration updates or non-critical telemetry requests, but other type of messages such as actuations would require a limit within seconds.

Future work is directed to implementing an automated mechanism in the gateway that can adapt to network changes (in size or in QoS requirements) following the knowledge derived from this work and use case limitations introduced by users, and its implications with gateway federation in multiple-gateway networks regarding gateway routing selection in the network server.

## REFERENCES

[1] S. Balaji, K. Nathani, and R. Santhakumar, "IoT technology, applications and challenges: A contemporary survey," *Wireless Pers. Commun.*, vol. 108, no. 1, pp. 363–388, Apr. 2019, doi: 10.1007/s11277-019-06407-w.

[2] W. Abdallah, S. Mnasri, N. Nasri, and T. Val, "Emergent IoT wireless technologies beyond the year 2020: A comprehensive comparative analysis," in *Proc. Int. Conf. Comput. Inf. Technol.*, Sep. 2020, pp. 1–5, doi: 10.1109/ICCIT-144147971.2020.9213799.

[3] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Exp.*, vol. 5, no. 1, pp. 1–7, Mar. 2019, doi: 10.1016/j.icte.2017.12.005.

[4] *Zero Defect Manufacturing Platform Project 2021*. Accessed: Mar. 7, 2023. [Online]. Available: https://www.zdmp.eu/

[5] (Jan. 2022). *KPN Activates Class B to Level up Its LoRaWAN Network With Groundbreaking Capabilities*. Accessed: Mar. 7, 2023. [Online]. Available: https://www.actility.com/kpn-activates-class-b-to-level-up-its-lorawan-network/

[6] K. Mikhaylov, J. Petäjäjärvi, and A. Pouttu, "Effect of downlink traffic on performance of LoRaWAN LPWA networks: Empirical study," in *Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2018, pp. 1–6, doi: 10.1109/PIMRC.2018.8580721.

[7] J. P. S. Sundaram, W. Du, and Z. Zhao, "A survey on LoRa networking: Research problems, current solutions, and open issues," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 371–388, 1st Quart., 2020, doi: 10.1109/COMST.2019.2949598.

[8] M. A. M. Almuhaya, W. A. Jabbar, N. Sulaiman, and S. Abdulmalek, "A survey on LoRaWAN technology: Recent trends, opportunities, simulation tools and future directions," *Electronics*, vol. 11, no. 1, p. 164, Jan. 2022, doi: 10.3390/electronics11010164.

[9] A. H. Jebril and R. A. Rashid, "A systematic literature review on downlink frames in LoRaWAN," *Comput. Electr. Eng.*, vol. 101, Jul. 2022, Art. no. 108006, doi: 10.1016/j.compeleceng.2022.108006.

[10] A.-I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, "Does bidirectional traffic do more harm than good in LoRaWAN based LPWA networks?" 2017, *arXiv:1704.04174*.

[11] A. Ivoghlian, K. I.-K. Wang, and Z. Salcic, "Application-aware adaptive parameter control for LoRaWAN," *J. Parallel Distrib. Comput.*, vol. 166, pp. 166–177, Aug. 2022, doi: 10.1016/j.jpdc.2022.04.023.

[12] D. Zorbas, "Improving LoRaWAN downlink performance in the EU868 spectrum," *Comput. Commun.*, vol. 195, pp. 303–314, Nov. 2022, doi: 10.1016/j.comcom.2022.09.001.

[13] J. M. Marais, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on the viability of confirmed traffic in a LoRaWAN," *IEEE Access*, vol. 8, pp. 9296–9311, 2020, doi: 10.1109/ACCESS.2020.2964909.

[14] M. Pasetti, E. Sisinni, P. Ferrari, S. Rinaldi, A. Depari, P. Bellagente, D. D. Giustina, and A. Flammini, "Evaluation of the use of class B LoRaWAN for the coordination of distributed interface protection systems in smart grids," *J. Sensor Actuator Netw.*, vol. 9, no. 1, p. 13, Feb. 2020, doi: 10.3390/jsan9010013.

[15] C. Zaraket, I. Dogas, and D. Kalyvas, "Open source LoRaWAN telemetry test bench for smart grid—A DLMS/COSEM implementation case study," in *Proc. AIP Conf.*, Aug. 2022, vol. 2437, no. 1, pp. 1–12, doi: 10.1063/5.0095471.

[16] G. Scaramella, G. C. Heck, L. Lippmann Junior, R. A. Hexsel, T. Santana, and V. B. Gomes, "Enabling LoRaWAN communication over Wi-SUN smart grid networks," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 4842–4847, doi: 10.1109/ICC45855.2022.9838959.

[17] D.-Y. Kim, J. Park, and S. Kim, "Data transmission in backscatter IoT networks for smart city applications," *J. Sensors*, vol. 2022, pp. 1–9, Nov. 2022, doi: 10.1155/2022/4973782.

[18] D. Zorbas and X. Fafoutis, "Time-slotted LoRa networks: Design considerations, implementations, and perspectives," *IEEE Internet Things Mag.*, vol. 4, no. 1, pp. 84–89, Mar. 2021, doi: 10.1109/IOTM.0001.2000072.

[19] J. Finnegan, S. Brown, and R. Farrell, "Evaluating the scalability of LoRaWAN gateways for class B communication in NS-3," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Oct. 2018, pp. 1–6, doi: 10.1109/CSCN.2018.8581759.

[20] H. E. Elbsir, M. Kassab, S. Bhiri, and M. H. Bedoui, "Evaluation of LoRaWAN class B efficiency for downlink traffic," in *Proc. 16th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2020, pp. 105–110, doi: 10.1109/WiMob50308.2020.9253405.

[21] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving reliability and scalability of LoRaWANs through lightweight scheduling," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1830–1842, Jun. 2018, doi: 10.1109/JIOT.2018.2815150.

[22] L. Chasserat, N. Accettura, P. Berthou. (Oct. 2022). *LoRaSync: Energy Efficient synchronization for Scalable LoRaWAN*. Accessed: Mar. 7, 2023. [Online]. Available: https://hal.laas.fr/hal-03694383v2

[23] V. Di Vincenzo, M. Heusse, and B. Tourancheau, "Improving downlink scalability in LoRaWAN," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7, doi: 10.1109/ICC.2019.8761157.

[24] D. Ron, C.-J. Lee, K. Lee, H.-H. Choi, and J.-R. Lee, "Performance analysis and optimization of downlink transmission in LoRaWAN class B mode," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7836–7847, Aug. 2020, doi: 10.1109/JIOT.2020.2994958.

[25] H. E. Elbsir, M. Kassab, S. Bhiri, and M. H. Bedoui, "Evaluation of LoRaWAN class B performances and its optimization for better support of actuators," *Comput. Commun.*, vol. 198, pp. 128–139, Jan. 2023, doi: 10.1016/j.comcom.2022.11.016.

[26] *Semtech Application Note AN1200.22. 2015; LoRa Modulation Basics*. Accessed: Mar. 7, 2023. [Online]. Available: http://wiki.lahoud.fr/lib/exe/fetch.php?media=an1200.22.pdf

[27] LoRa Alliance, Notice of Use and Disclosure. (2017). *LoRaWAN 1.1 Specification*. Accessed: Mar. 7, 2023. [Online]. Available: https://net868.ru/assets/pdf/LoRaWAN-v1.1.pdf

[28] LoRa Alliance. (2017). *LoRaWAN 1.1 Regional Parameters*. Accessed: Mar. 7, 2023. [Online]. Available: https://lora-alliance.org/wp-content/uploads/2020/11/lorawan-regional-parameters-v1.1ra.pdf

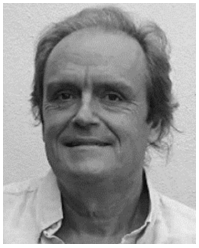
[29] S. Brooks, "Markov chain Monte Carlo method and its application," *J. Roy. Stat. Soc., D, Statistician*, vol. 47, no. 1, pp. 69–100, Mar. 1998, doi: 10.1111/1467-9884.00117.

[30] Y. Shiferaw, A. Arora, and F. Kuipers, "LoRaWAN class B multicast scalability," in *Proc. IFIP Netw. Conf.*, Jun. 2020, pp. 609–613. [Online]. Available: http://dl.ifip.org/db/conf/networking/networking2020/1570619875.pdf

[31] D. Todoli-Ferrandis, J. Silvestre-Blanes, and V. Sempere-Payá, "Robust downlink mechanism for industrial Internet of Things using LoRaWAN networks," *Electronics*, vol. 10, no. 17, p. 2122, Aug. 2021, doi: 10.3390/electronics10172122.

**DAVID TODOLI-FERRANDIS** was born in Valencia, Spain, in 1983. He received the first M.Sc. degree in telecommunications engineering from the Polytechnic University of Valencia, and the second M.Sc. degree in energy installations management and project's internationalization from Universidad Cardenal Herrera-CEU, Valencia.

He has worked in various research and development projects with the Communications Department, Polytechnic University of Valencia, and ITACA Institute, related to wireless sensor networks, WiMAX, and communication systems modeling. Since 2013, he has been with the Advanced Communications and Industrial Computing Research Group, Instituto Tecnológico de Informática, involved in wireless sensor networks and LPWAN systems, network modeling, and the Industrial Internet of Things (IIoT) for industry 4.0 applications, participating in several European projects. He has coauthored several publications related to wireless sensor networks and modeling.



**JAVIER SILVESTRE-BLANES** received the M.Sc. and Ph.D. degrees in computer architecture. In 1999, he joined the Polytechnic University of Valencia, as an Associate Lecturer with the Department of Computer Architecture, where he has been an Associate Professor, since 2010. He joined ITI, in 2009, in the research and development area (industrial computing, communications, and image processing). He has more than 32 papers in international conferences, 22 articles in journals, and eight book chapters, related to the sector of industrial communications, heterogeneous networks, multimedia networks, image processing, and computer vision. He is a member of the Industrial Electronics Society (IES), belonging to the IEEE, and the sub-committee FA10 Computer Vision and Human–Machine Interaction in Industrial and Factory Automation, belonging to the IES-IEEE Factory Automation Committee. Since 2005, he has been participating as a member of the organizing committee and the program committee of numerous international conferences of the IES-IEEE.

**VÍCTOR SEMPERE-PAYÁ** (Associate Member, IEEE) received the degree in industrial electronics and computer engineering and the Ph.D. degree in telecommunications engineering from the Polytechnic University of Valencia (UPV). Currently, he is an Associate Professor with the Department of Communications, UPV, where he teaches industrial communications and public access networks. He is also the Director of the Advanced Communications and Industrial Computing Research Group, Instituto Tecnológico de Informática. He has authored or coauthored more than 60 technical papers in journals and international conferences. He has managed more than 50 research and technological projects. His current research interests include factory communications, real-time communications, and quality of service (QoS) in networks. He has served as a program committee member for several conferences in the area of factory communications.

**SALVADOR SANTONJA-CLIMENT** received the M.Sc. degree in telecommunications engineering and in communications technologies, systems and networks from the Polytechnic University of Valencia. He has worked in various research and development projects with the Communications Department, Polytechnic University of Valencia, related to industrial communications, multimedia, telemetry, and ad hoc wireless protocols. He is currently the Project Manager and a Coordinator of the Advanced Communications Research Group, Instituto Tecnológico de Informática. He has participated in more than 20 research and development projects, involved in European projects related to Industry 4.0, IoT architectures, wireless protocols, telemetry, and energy management. He is the coauthor of 16 publications and one patent.