future internet

Article

Polling Mechanisms for Industrial IoT Applications in Long-Range Wide-Area Networks

David Todoli-Ferrandis [1], Javier Silvestre-Blanes [1,2,*], Víctor Sempere-Payá [1,3] and Salvador Santonja-Climent [1]

[1] Instituto Tecnológico de Informática (ITI), 46022 Valencia, Spain; dtodoli@iti.es (D.T.-F.); vsempere@upv.es (V.S.-P.); ssantonja@iti.es (S.S.-C.)
[2] Departamento de Informática de Sistemas y Computadores (DISCA), Universitat Politècnica de València (UPV) (ITI), 46022 Valencia, Spain
[3] Departamento de Comunicaciones (DCOM), Universitat Politècnica de València (UPV) (ITI), 46022 Valencia, Spain
* Correspondence: silbla@upv.es

Abstract: LoRaWAN is a low-power wide-area network (LPWAN) technology that is well suited for industrial IoT (IIoT) applications. One of the challenges of using LoRaWAN for IIoT is the need to collect data from a large number of devices. Polling is a common way to collect data from devices, but it can be inefficient for LoRaWANs, which are designed for low data rates and long battery life. LoRaWAN devices operating in two specific modes can receive messages from a gateway even when they are not sending data themselves. This allows the gateway to send commands to devices at any time, without having to wait for them to check for messages. This paper proposes various polling mechanisms for industrial IoT applications in LoRaWANs and presents specific considerations for designing efficient polling mechanisms in the context of industrial IoT applications leveraging LoRaWAN technology.

Keywords: LoRaWAN; low-power wide-area network (LPWAN); scalability; network optimization

check for updates

Citation: Todoli-Ferrandis, D.; Silvestre-Blanes, J.; Sempere-Payá, V.; Santonja-Climent, S. Polling Mechanisms for Industrial IoT Applications in Long-Range Wide-Area Networks. *Future Internet* **2024**, *16*, 130. https://doi.org/10.3390/fi16040130

Academic Editors: Zhihao Liu, Franco Davoli and Davide Borsatti

Received: 27 March 2024
Revised: 9 April 2024
Accepted: 10 April 2024
Published: 12 April 2024

1. Introduction

The industrial Internet of Things (IIoT) is rapidly transforming the way we operate and manage industrial assets. By connecting sensors, actuators, and other devices to the internet, IIoT can provide real-time data and insights that can be used to improve efficiency, productivity, and safety. As introduced in [1,2], the concept of functional safety is relevant in industrial systems. Functional safety is the ability of a system to perform its required safety functions in a reliable manner under all conditions, including normal operation, abnormal operation, and fault conditions.

One challenge is that the IIoT is a complex and heterogeneous environment, with a wide variety of devices and networks. This can make it difficult to ensure that all devices and networks comply with functional safety requirements. Another challenge is that the IIoT is a dynamic environment, with devices and networks constantly changing. This can make it difficult to maintain functional safety over time.

IIoT is a trending solution to collect data from a large number of devices, which as can be seen, comes with challenges. Polling is a popular way to gather data in industrial use cases and digitalization because it is simple and efficient. The central server polls each device at regular intervals, and the device sends back the data that it has collected. This can be completed over a variety of networks, including wired, wireless, and cellular. Polling has a number of advantages, including the following:

1. Simple: Polling is a simple and straightforward way to gather data. The central server does not need to know anything about the devices that it is polling, and the devices do not need to be able to communicate with each other.

2. Efficient: Polling can be an efficient way to gather data, especially if the devices are not generating a lot of data. The central server can poll the devices at regular intervals, and the devices only need to send back the data that has changed since the last poll.
3. Scalable: Polling can be scaled to accommodate a large number of devices. The central server can simply poll more devices at regular intervals.

However, polling also has some challenges to overcome, including the following:

1. Latency: Polling can introduce latency into the system. The time it takes for the central server to poll a device and for the device to send back the data can add up.
2. Battery drain: Polling can drain the batteries of the devices that are being polled. If the devices are polled too often, the batteries may not last as long.
3. Bandwidth usage: Polling can use up a lot of bandwidth, especially if the devices are requested by the polling master to generate data more frequently.

Polling has been used for decades to gather data from a variety of devices. This means that there are a lot of tools and resources available to help implement polling on a system. It is compatible with a wide range of devices. So, it can be used to gather data from a variety of devices, including sensors, actuators, and controllers. This makes it a flexible and versatile solution for gathering data in industrial use cases, which comprise applications such as environmental monitoring, asset tracking, or smart metering.

While low-power wide-area networks (LPWANs) offer long battery life, low data rates, and wide coverage, their limited bandwidth presents challenges for polling in industrial Internet of Things (IIoT) applications. This is particularly true for LoRaWAN, a popular LPWAN technology. Although LoRaWAN boasts long range, low power, wide availability, security, and private network deployment, its bandwidth constraints necessitate careful consideration when employing polling techniques. These challenges include the following:

1. Bandwidth overhead: polling can consume a significant amount of bandwidth, which can be a problem for LoRaWAN.
2. Latency: polling can introduce latency, as the central controller must wait for each device to respond before it can request data from the next device.
3. Power consumption: polling can also increase the power consumption of devices, as they must wake up from sleep mode to respond to polling requests.

LoRaWAN is better suited for a polling system than other LPWANs because it supports different medium access policies that enable a more efficient downlink communication, which is key to enable new solutions to reduce latencies, take advantage of multicast support to reduce overhead, and have better management of power consumption.

Within the LoRaWAN device classes, Class B and Class C have fundamental differences in their ability to receive messages and manage power. The choice between them depends on various factors related to the environment and specific application requirements. Class A focuses on uplink, where only the end device can initiate data exchange, so it is discarded for polling applications. The following explores scenarios where polling is more appropriate for each class.

Class B allows for asynchronous bidirectional communication, meaning Class B devices can receive messages at scheduled times known as "receive slots". This approach is useful in environments where there is a need for a balance between power savings and responsiveness. This is particularly beneficial in scenarios where devices need to communicate periodically, like in remote monitoring applications, or in general when traffic is low, as it helps conserve energy by minimizing unnecessary listening.

Class C, unlike Class B, allows for continuous message reception, partially sacrificing energy efficiency for increased responsiveness. Here, synchronous polling becomes an effective strategy in case environments where latency is critical, such as security systems or real-time asset tracking, or in situations where node density is high, and data traffic is constant.

Both classes can be used to implement a polling system. The choice between Class B and Class C in a LoRaWAN polling application depends on the nature of the application and power consumption requirements. Class B is suitable for environments where energy efficiency is crucial and data traffic is sporadic, while Class C is preferable in situations that require higher responsiveness and are willing to accept higher energy consumption.

This paper presents the challenges of polling for IIoT over LoRaWAN. Then, it proposes a number of techniques that can be used to facilitate the adoption of polling on this technology, given different requirements. Finally, we discuss the future of polling for IIoT over LPWANs. Abbreviations includes all abbreviations and their meaning for readers convenience.

2. Related Work

Polling techniques for collecting data have been a topic well studied in the literature for several decades, especially because they have been commonly used in wired industrial networks. Nevertheless, wireless networks are only recently gaining relevance in industrial applications, with the popularity of IIoT and massive digitalization. A number of research papers have been published on the topic of using polling systems in wireless industrial applications. Some of these papers focus on developing new polling protocols that are more efficient and reliable than traditional polling protocols. Other papers focus on developing new methods for managing polling traffic in wireless industrial networks. The following papers are some of the most recent and relevant papers on the topic of polling systems in wireless industrial applications.

Authors in [3] provide a comprehensive overview of frequency-domain contention and polling MAC protocols for IEEE 802.11 wireless networks. The paper discusses the advantages and disadvantages of these protocols, as well as their performance in different scenarios, which is a valuable resource for researchers who are interested in using polling systems in wireless industrial applications. The paper provides a good understanding of the different frequency-domain contention and polling MAC protocols that are available, as well as their performance characteristics. Following the IEEE 802.11-based approach, the article in [4] proposes a new rate selection algorithm for IEEE 802.11 industrial wireless LANs (WLANs). The algorithm, called Rate Selection for Industrial Networks (RSIN), is designed to minimize the error probability of polling packets, while considering the deadline imposed on packet delivery. It covers interesting topics for wireless polling but focuses on WLANs. Moreover, ref. [5] proposes a prototype implementation of a critical industrial wireless control system based on ultra-reliable low-latency communication (URLLC). The system uses a polling scheme to ensure that all sensors are read and that all actuators are updated within a tight deadline. This shows a polling application but addresses its uses in a 5G traffic profile and critical system requirements.

In [6], authors propose the Acyclic Traffic Scheduling Algorithm (ATSA), which is not specifically designed for polling traffic. However, ATSA can be used to schedule polling traffic by giving polling traffic a higher priority than other acyclic traffic. This will ensure that polling traffic packets are always transmitted before other acyclic traffic packets. The work described in [7] proposes a new Medium Access Control (MAC) protocol for OFDMA-based networks (as Wi-Fi or cellular) in cyber-physical systems (CPSs). The protocol, called Priority-Aware Frequency Domain Polling (PAFDP), is designed to improve the reliability and performance of CPS networks by using priority-aware polling and frequency diversity.

Authors in [8] propose a polling-based transmission scheme for industrial IoT applications. The scheme is designed to improve the uniformity of network traffic, which can lead to improved performance and energy efficiency. The paper begins by discussing the challenges of polling-based communication in industrial IoT applications. These challenges include the following: the need to collect data from a large number of devices, to collect data at regular intervals, and to minimize the amount of network traffic. The paper then proposes a polling-based transmission scheme that is designed to address these challenges, assuming the use of current industrial wireless network protocols such as ISA100.11a or Wireless HART. The scheme uses an algorithm to generate a schedule for polling the devices. The schedule is designed to minimize the variance in the time between polls, which can lead to improved uniformity of network traffic. Then, it presents an experimental evaluation of the proposed scheme using a simulation of an industrial IoT network. The results show that the proposed scheme can significantly improve the uniformity of network traffic. This led to improved performance and energy efficiency. Nevertheless, authors conclude that future work should focus on the integration of the proposed scheme with other network protocols and the evaluation of the proposed scheme in real-world industrial IoT networks.

These works describe new polling schemes that adapt wireless communications for industrial applications but are based on technologies that have been around for longer or ones requiring high bandwidth, such as Wi-Fi or 5G. Focusing on adapting IoT constrained devices, and especially on LoRaWAN as the candidate standard to use, authors in [9] discuss the challenges of implementing LoRaWAN scheduling and present a low-overhead synchronization and scheduling concept that is fully compliant with the LoRaWAN standard. The paper begins by discussing the scalability challenges of LoRaWAN, which is due to its Aloha-based MAC layer. Aloha is a random-access protocol that can lead to collisions, which can reduce the overall network performance. The paper then presents a low-overhead synchronization and scheduling concept that is designed to address the scalability challenges of LoRaWAN. The concept is based on the use of time synchronization and scheduling slots. Time synchronization is used to ensure that all devices in the network are synchronized to the same time reference. This is necessary for scheduling slots, as devices need to know when they are allowed to transmit data. The evaluation was conducted using a testbed of LoRaWAN nodes. The results of the evaluation showed that the proposed concept can significantly improve the PDR, especially under high network loads. The paper concludes by discussing the future work on LoRaWAN scheduling. The authors suggest that future works should focus on the following areas: scalability, security, and efficiency.

In [10], TRILO, or Traffic Indication-based Downlink Communication Protocol (for LoRaWAN), describes a downlink communication protocol that is designed to improve the energy efficiency of LoRaWANs, by using a beacon-based polling mechanism to inform end devices when they have downlink traffic. When an end device receives a beacon, it checks the number of downlink messages that are pending for it. If there are any downlink messages pending, the end device will open a ping slot to receive the messages. The TRILO protocol has been shown to improve the energy efficiency of LoRaWANs by up to 20%. This is because TRILO reduces the amount of time that end devices spend listening for downlink traffic. Nevertheless, the solution proposed only implements sequential polling on the premise that it helps avoiding collisions, disregarding the multi-channel capabilities of LoRaWAN gateways. Also, it does not consider Class B operations, which are already similar to the proposal the authors describe. While [11] proposes a lightweight scheduling solution for LoRaWAN, it focuses on overall network traffic management and necessitates the deployment of a custom MAC layer named RS-LoRA. This proprietary approach introduces backward compatibility issues with existing LoRaWAN devices adhering to the standard MAC protocol. The article in [12] proposes a solution specifically for LoRaWAN-based AMR (Automatic Meter Reading) systems. It addresses this challenge by introducing a polling algorithm for the Outage Management System (OMS) that aims to critically reduce the number of queries needed to identify system conditions during outages, but it does so by analyzing outages and restorations to confirm without polling all the AMRs in the system, so it is not really applicable to other applications.

In addition to the papers listed above, there are a number of other papers that have been published on the topic of polling systems in wireless industrial applications. A comprehensive review of all of the relevant literature is beyond the scope of this paper. However, the papers listed above provide a good starting point for researchers who are interested in this topic.

These papers demonstrate the growing interest in developing new and improved polling schemes for wireless industrial communication networks. However, there is still room for further research in this area, and the literature on the applications of polling specifically in LoRaWANs is not as extensive as other wireless solutions that have been around for longer. One promising area of research is the development of polling schemes that are adaptive to the dynamic conditions of wireless networks. This would allow polling schemes to optimize their performance for different traffic loads, channel conditions, and device types.

3. Materials and Methods

This section proposes different types of polling mechanisms for LoRaWAN in more detail. After introducing the basics of the technology, two polling alternatives for Class B and two for Class C are described. They can be used to perform polling in different environments in terms of time requirements, number of nodes, and consumption. Nevertheless, with the described polling systems, the majority of polling applications involving a LoRaWAN are covered; furthermore, these alternatives can be combined to adapt dynamically to environment and application changes.

3.1. Technology Overview and Operation in LoRaWAN

At the physical layer, the radio transceivers of the end devices use Semtech's long-range (LoRa) technology [13]. 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 CSS (Chirp Spread Spectrum), using linear frequency modulation chirp pulses with high bandwidth to encode information. These chirp pulses are sinusoidal signals varying in frequency over time, which determines the symbols that represent the information. The spreading factor (SF) determines the number of bits encoded in each symbol, with a 2^{SF} relationship. Valid SF values range from 7 to 12. Equation (1) shows how to calculate the symbol duration (Ts) in seconds based on SF and bandwidth (BW).

$$\mathrm{T_s} = \frac{2^{\mathrm{SF}}}{\mathrm{BW}} \tag{1}$$

According to Equation (1), increasing SF reduces the bit rate and extends the packet's time on air (ToA). This also increases power consumption because the radio interface stays active longer for data transmission. However, higher SF values enhance coverage range due to the increased ToA, making them more resistant to noise [14].

LoRaWAN, an open protocol defined by LoRa Alliance, operates on top of LoRa technology. A central network server coordinates all network devices (end nodes and gateways), including selecting the optimal gateway for each node. As illustrated in Figure 1, the LoRaWAN architecture utilizes application servers to forward information to other systems and networks using various application protocols.

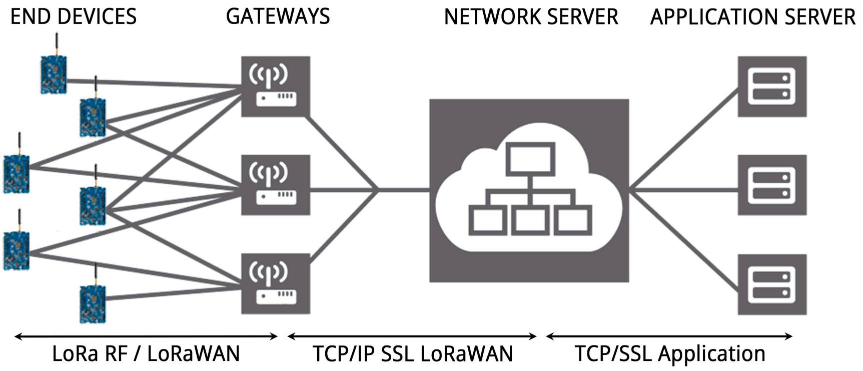


Figure 1. LoRaWAN architecture.

The latest version of the protocol released by LoRa Alliance is version 1.1 [15]. The gateways are connected to the network server through a conventional TCP/IP SSL network, while the end devices use LoRa to communicate with one or more gateways.

Another matter to keep in mind when sending and receiving messages in a LoRaWAN 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 [16].

Table 1. Duty cycle restrictions on EU bands.

Name	Band (MHz)	Limitations
G	8630–8680 MHz	EIRP < 25 mW—duty cycle < 1%
G1	8680–8686 MHz	EIRP < 25 mW—duty cycle < 1%
G2	8687–8692 MHz	EIRP < 25 mW—duty cycle < 0.1%
G3	8694–86965 MHz	EIRP < 500 mW—duty cycle < 10%
G4	8697–8700 MHz	EIRP < 25 mW—duty cycle < 1%

In the described work, the LoRaWAN Regional Parameters 1.0.2 Rev B is used to define duty-cycled limited transmissions to comply with the European Telecommunications Standards Institute (ETSI) regulations. In ETSI, most bands use a maximum duty cycle of 1%, but some use 0.1% and 10%. For any network, SF11 and SF12 are only allowed when the ADR (Adaptive Data Rate) mechanism is enabled. It is important to highlight that increasing the spreading factor (SF) by one almost doubles the ToA (for the same BW), which also directly impacts the duty cycle waiting time, as it is proportional to ToA.

For example, a single transmission on SF10 takes more time than six transmissions on SF7. This behavior also influences the maximum payload size for each SF.

The default transmission configuration recommended for the downlink is 869.525 MHz (G3 sub-band offering the highest DC) with SF9 and 125 KHz bandwidth [16]. Nevertheless, the default channel can be modified by gateways using MAC commands in accordance with the LoRaWAN Regional Parameters.

Regarding the MAC access layer, there are three classes of LoRaWAN nodes (Figure 2).

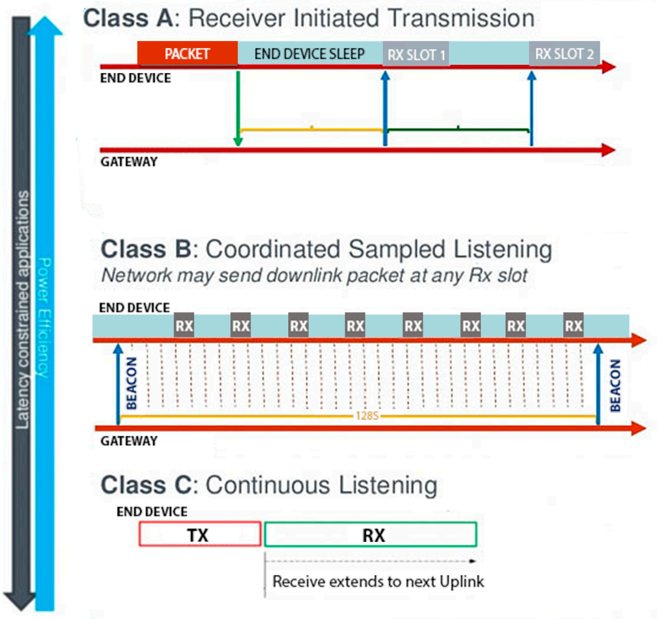

Class A: Receiver Initiated Transmission
PACKET
END DEVICE SLEEP
RX SLOT 1
RX SLOT 2
END DEVICE
GATEWAY
Class B: Coordinated Sampled Listening
Network may send downlink packet at any Rx slot
END DEVICE
RX
RX
RX
RX
RX
RX
RX
RX
BEACON
128S
BEACON
GATEWAY
Class C: Continuous Listening
END DEVICE
TX
RX
Receive extends to next Uplink
Latency constrained applications
Power Efficiency

Figure 2. LoRaWAN Classes A, B, and C.

1. Class A: This is the most basic class of the LoRaWAN node. It can only receive downlinks when it is actively sending an uplink, so it is not suitable for polling

applications. This means that Class A nodes have the best energy efficiency, but they also have the lowest data throughput.

2. Class B: This class works by periodically opening receive windows to receive downlink messages. These receive windows are called ping slots. The network server broadcasts a time-synchronized beacon periodically through the gateways, which is received by the end devices. The end devices then use this timing reference to schedule when ping slots are opened to receive potential downlinks from the network. The parameters and times shown in Figure 3 correspond to the recommendations found in the standard.

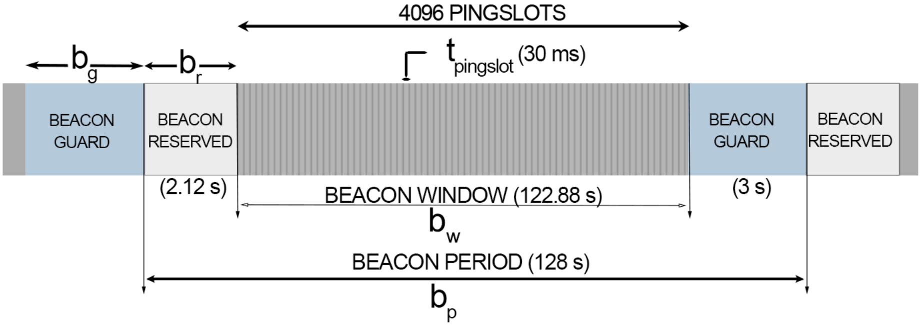


Figure 3. LoRaWAN Class B operation mode time parameters.

The number of ping slots (*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) by setting the number of ping slots usable by a node in each beacon period from 1 to 128 ping slots, as follows:

$$pingNb = 2^{7-k} \text{ where } 0 \leq k \leq 7 \quad (2)$$

The ping period is the delay between two ping slots and it is constant; moreover, it is calculated as follows:

$$pingPeriod = \frac{4096}{2^k} \text{ slots} \quad (3)$$

3. Class C: This class of LoRaWAN nodes can receive downlinks at any time (provided the sender complies with duty cycle regulations). This gives Class C nodes the highest data throughput, but it also has the lowest energy efficiency.

The choice of the LoRaWAN node class depends on an application's requirements. Class A nodes are best suited for applications where energy efficiency is critical, such as environmental monitoring. Class B nodes are a good compromise between data throughput and energy efficiency, and they are suitable for a wide range of applications. Class C nodes are best suited for applications where data throughput is critical, such as actuator control.

Finally, polling scheduling then depends on the network devices support for multicast messages, which was introduced by the LoRaWAN Remote Multicast Setup v1.0.0 Specification [17], which preferably needs nodes to support the LoRaWAN 1.1 Specification [15] in their firmware. As many devices are still only compatible with previous versions of this specification, the solutions proposed for polling consider both unicast and multicast support.

3.2. Polling on Class B

When using Class B for polling under the standard, operation collisions may happen frequently. As the NS randomly assigns slots, a node can start answering the downlink message when the GW has already started sending another downlink. Moreover, a node may be in the middle of an uplink, which started when the GW's radio was free for reception, when a downlink scheduled slot should open.

In summary, the GW's radio cannot receive and transmit at the same time, so when a downlink and uplink events concur, the one started first takes priority and the other event is missed (see Figure 4).

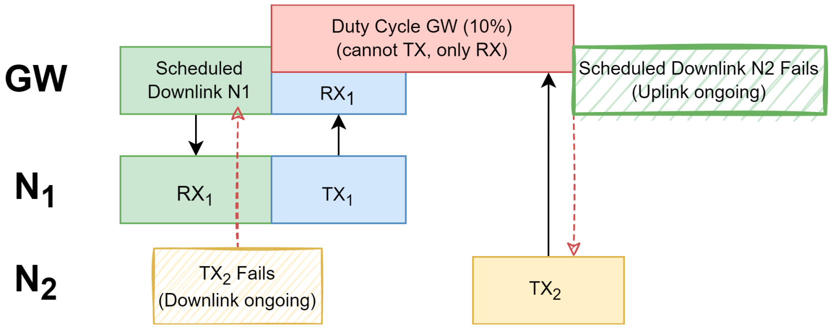


Figure 4. Failed TX/RX events due to shared radio unavailability.

To prevent this and allow nodes to transmit efficiently, they should only transmit right after, and only after, receiving a downlink. This ensures that the GW's radio cannot be in transmission mode as it needs to wait a given time due to duty cycle regulations, and therefore it will be listening. It also ensures that no other node will occupy the channel and cause a possible collision. In order to achieve this particular implementation of a polling system, LoRaWAN Class B requires some configuration changes and optimization.

Due to scalability issues in these types of networks, the standard Class B parameters of the beacon period, beacon window, and slot size are not fit for a usable polling system, as this introduces many packet losses and missed slots. In this case, the solution that enables a feasible polling system is achieved using the Adaptive Beacon Period Configurator (ABPC) for scalable LoRaWAN downlink applications. A detailed description of ABPC can be found in [18]. ABPC works by dynamically adjusting the beacon period of LoRaWAN gateways. The beacon period is the interval at which gateways transmit beacon frames. Beacon frames are used by end devices to synchronize their clocks and to learn about the available channels. ABPC can adjust the beacon period based on the following factors: the number of end devices in the network, the traffic load on the network, the channel conditions, battery constraints, or latency requirements.

ABPC increases the beacon period when the network is congested or when the channel conditions are poor. This reduces the overhead of beacon frames and improves the performance of downlink traffic.

ABPC decreases the beacon period when the network is not congested and when the channel conditions are good. This reduces the latency of downlink traffic.

The results showed that ABPC significantly improves the scalability and performance of LoRaWANs with a large number of end devices. In summary, the ABPC algorithm operates as follows:

1. Observe network conditions: The ABPC algorithm periodically observes the network conditions, such as the traffic density and the average message payload size. This information can be obtained from the network server.
2. Calculate the Class B beacon period: Based on the observed network conditions, the ABPC algorithm calculates the optimal beacon period and its derived ping slot length and *pingNb* and *k* (see Figure 5). The optimal beacon period is the value that balances the energy consumption of the end devices with the downlink latency and Packet Delivery Ratio (PDR).
3. Adjust beacon period: The ABPC algorithm adjusts the beacon period in the gateway. The updated beacon period and ping slot length are then broadcasted to the end devices.

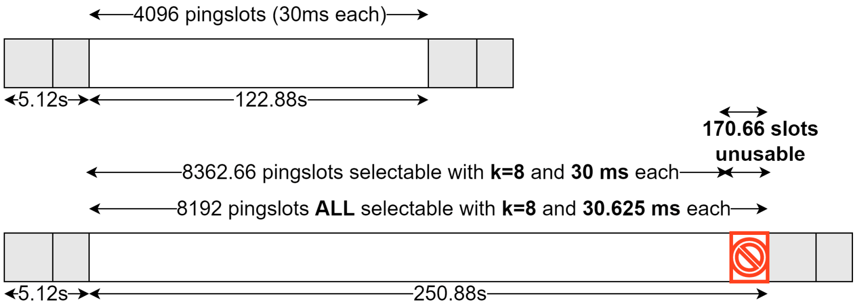

4096 pingslots (30ms each)
5.12s
122.88s
170.66 slots unusable
8362.66 pingslots selectable with **k=8** and **30 ms** each
8192 pingslots **ALL** selectable with **k=8** and **30.625 ms** each
5.12s
250.88s

Figure 5. ABPC selection of parameters for a 256 s beacon period instead of standard 128 s [17].

The ABPC algorithm is based on the observation that the performance of the LoRaWAN downlink is affected by several factors, including the following:

1. Traffic density: the higher the traffic density, either by number of devices or packets per device, the longer beacon period needed to support this traffic, which can increase the downlink latency.
2. Message payload size: a larger message payload means that the gateways need to transmit more data, which can increase the downlink latency and the probability of collision events (see Figure 4).

The ABPC algorithm addresses these factors by dynamically adjusting the beacon period based on the observed network conditions. The ABPC algorithm can be used to improve the performance of the LoRaWAN downlink in a variety of scenarios, including the following:

1. High-traffic scenarios: in high-traffic scenarios, the ABPC algorithm can reduce the downlink latency by adjusting the beacon period downwards.
2. Low-traffic scenarios: in low-traffic scenarios, the ABPC algorithm can improve the energy efficiency of the end devices by adjusting the beacon period upwards.
3. Scenarios with variable traffic density: in scenarios with variable traffic density, the ABPC algorithm can automatically adjust the beacon period to meet the changing demands of the network.

A scheduler running this algorithm on the network server allows knowing in advance the performance of a Class B network and can increase or decrease the beacon period to respond to different network sizes while maintaining the application requirements, such as PDR or latency. For a polling system to work, downlink messages must be received by nodes to enable transmissions, so ABPC is used to ensure that the PDR is > 90% by adapting the beacon period.

Figure 6 shows the polling system for Class B in unicast mode for a typical beacon period of 128 s. Note that the duration of the ping slot and the beacon period, after ABPC calculates the optimal values for this configuration, can differ from the standard.

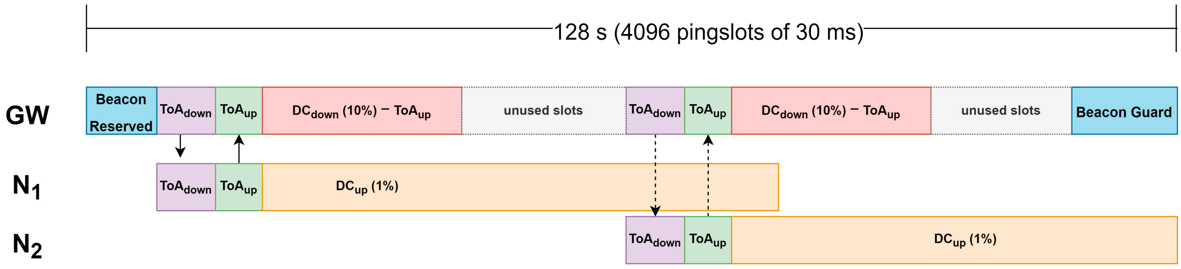


Figure 6. Solution with unicast polling on Class B.

Due to the pseudo-random channel access introduced by Class B ping slot offset selection at the beginning of each period (see Equations (4) and (5)), a PDR of 100% cannot be ensured.

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

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

A critical issue to address when using Class B is the possibility of having a node polled before its duty cycle guard time after the previous transmission is over. This can happen when the ToA and duty cycle time are longer than the beacon guard (b_g) plus the beacon reserved (b_r), which is 5.12 s, and the algorithm selects one of the first slots in the window. So, an additional step in the ABPC scheduler is added to calculate the required beacon guard time to ensure the slot will not be missed due to this problem. Taking the example of Figure 6, if one of the nodes is polled just before the beacon guard, even if it can correctly receive the downlink polling message at the start of a new period (because the gateway uses G3 sub-band enabling it to use 10 times more airtime), the node is unable to respond because it needs to wait until its duty cycle limitation is over.

Hence, the scheduler running the ABPC needs to calculate the optimal beacon guard time to prevent this from happening (6), so the correct beacon period may be longer than 128 s. This is a required modification to the ABPC in order to support polling applications.

$$b_g = DC_{up} - b_r \tag{6}$$

If nodes and GW have multicast support, then downlink messages can be used to trigger the polling simultaneously on a group of nodes (as shown in Figure 7 with an example of 4 nodes in a 128 s period, before the calculation of optimal beacon guard time), which then transmit at the same time but on different channels; the number of these channels can be as large as supported by the GW and defined as n_{ch} from now on. This results in multicast groups of size n_{ch}.

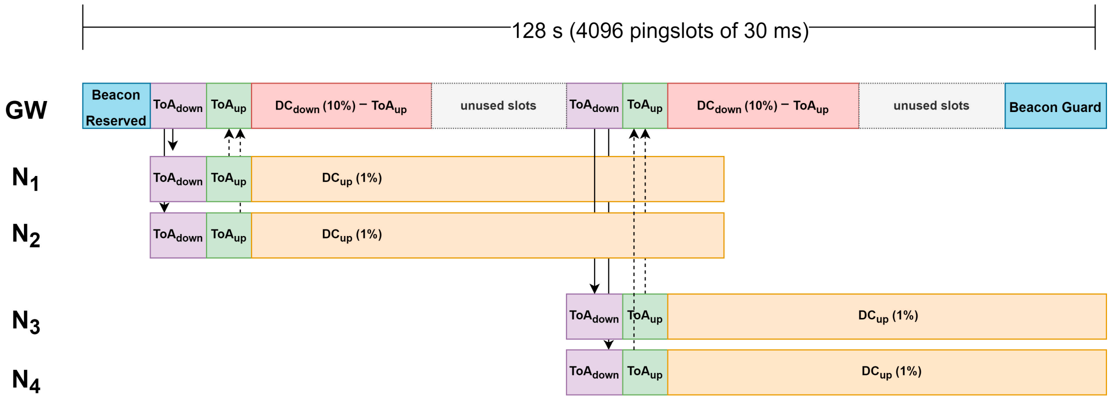


Figure 7. Solution with multicast polling on Class B.

Note that the possibility of using the synchronization beacon at the start of each period, regardless of unicast or multicast, is discarded. Modifying these beacons could break the compatibility for standard Class B nodes and other downlink messages that are non-polling-related.

3.3. Polling on Class C

By using Class C nodes, the overhead of control messages of Class B and the randomness of scheduling inside a beacon window can be overcome. Also, the polling period can be better adapted to the exact number of nodes in the network, disregarding the strict timing parameters of Class B (see Figures 8 and 9), which requires beacon periods to be powers of two. This mode of operation can be positive for nodes that are mains-powered and thus does not need to save battery by sleeping, which reduces the problems of synchronization.

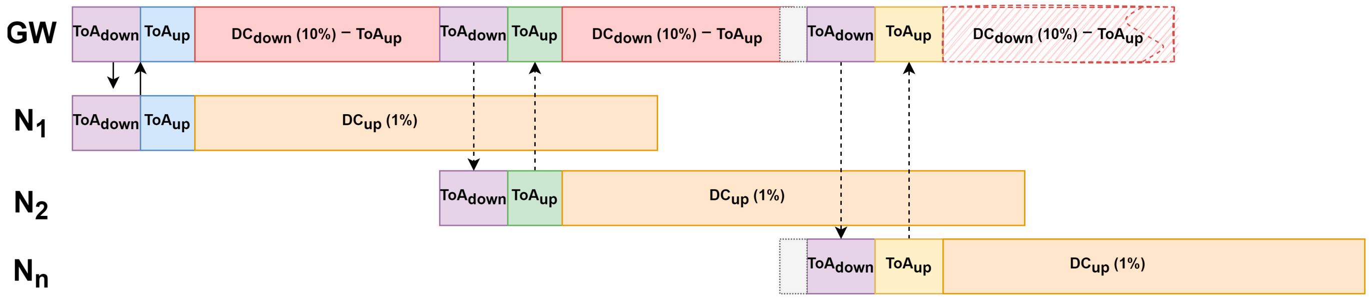
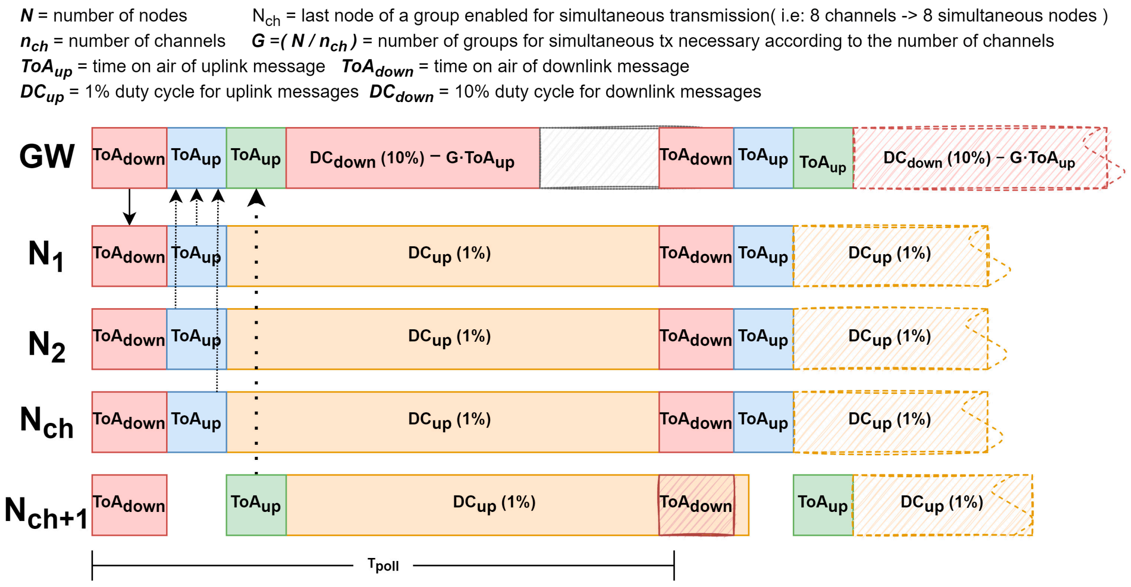


Figure 8. Solution with unicast polling on Class C.


N = number of nodes N_{ch} = last node of a group enabled for simultaneous transmission(i.e: 8 channels -> 8 simultaneous nodes)

n_{ch} = number of channels   *G =(N / n_{ch})* = number of groups for simultaneous tx necessary according to the number of channels

ToA_{up} = time on air of uplink message ToA_{down} = time on air of downlink message

DC_{up} = 1% duty cycle for uplink messages DC_{down} = 10% duty cycle for downlink messages

Figure 9. Solution with multicast polling on Class C.

The proposed polling mechanisms could be used to implement via software sleeping times in Class C nodes, which would help save energy if needed, but would break compatibility with regular Class C operation in case a downlink out of the scope of the polling system occurs. That is main reason why Class B is still interesting in terms of energy efficiency.

Again, if nodes and GW have multicast support, groups of size n_{ch} nodes can be polled simultaneously and respond at the same time but on n_{ch} different channels. In Figure 9, node N_{ch+1} represents a node that does not fit into the first group of n_{ch} devices that can be polled at the same time.

If the polling time T_{poll} is defined as the time between polling downlink messages from the gateway to the nodes, it can be calculated as in (7) for any case, as follows:

$$T_{poll} = \begin{cases} 10{\cdot}ToA_{down}\ , \ DC_{up} < G{\cdot}DC_{down} \\ 10{\cdot}ToA_{up}, \ \ DC_{up} \geq G{\cdot}DC_{down} \end{cases} \tag{7}$$

4. Results and Discussion

Different factors should be considered when choosing a polling mechanism for a particular application, based on the results obtained for each alternative. The message size was fixed to 51 bytes, which is LoRaWAN packet's maximum payload size for SF10. It is enough to contain several sensor measurements and a timestamp.

SF for downlink is also fixed to SF9 following the standard's recommendations as it provides a good compromise between time on air (which impacts coverage and resilience to noise) and data rate. The simulation also assumes the gateway has an eight-channel radio, which is the most extended option off-the-shelf. Table 2 shows all the related parameters.

Table 2. Scenario and simulation configuration.

Parameter	Description	Value
N	Number of devices	1. . .100
SF_u	Spreading factor uplink	SF7–10
SF_d	Spreading factor downlink	SF9
$M_{down\_payload}$	Message size (up/down)	51 Bytes
DC_d	Duty cycle downlink limit in g3 sub-band	10%
DC_u	Duty cycle uplink limit in g1 sub-band	1%
n_{ch}	Number of channels available in GW	[1-8-16-32]
$I_{TX/RX}$	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_{beacon}	Time on air to receive beacons	160 ms
$ToA_{downlink}$	Time on air listening for empty slots	328,704 ms

4.1. Class B Polling Results

Class B polling in wireless networks introduces some randomness due to the way slots are selected for nodes at the beginning of each beacon period. This mechanism, designed to improve interference resilience, means that polling periods in Class B represent maximum theoretical latencies rather than deterministic intervals.

With the described ABPC mechanism over the Class B scheduler, the maximum polling time is derived from the beacon period, but it can be shorter. Table 3 shows the calculated beacon guard duration, according to (6), using the modified ABPC.

Table 3. Beacon guard duration to adapt to uplink ToA.

SF	Beacon Guard Duration (s)
SF7	0
SF8	17
SF9	31
SF10	60

Figure 10 shows stepped or discrete evolution in the max polling time duration when network size changes. This behavior is directly related to how the Class B scheduler configures beacon period durations, proportional to powers of 2 and fixed for intervals of network sizes. As expected, using the multicast solution reduces the maximum polling time by a factor of n_{ch} against the unicast. This can be seen in better detail in Figure 11, which shows the impact of n_{ch} in the maximum polling latency for a fixed network size of 100 nodes.

The ratio in maximum latency between devices with different SFs becomes more significant when more channels are available. This is another point to consider when deciding to operate with a given number of channels and when having variable SFs.

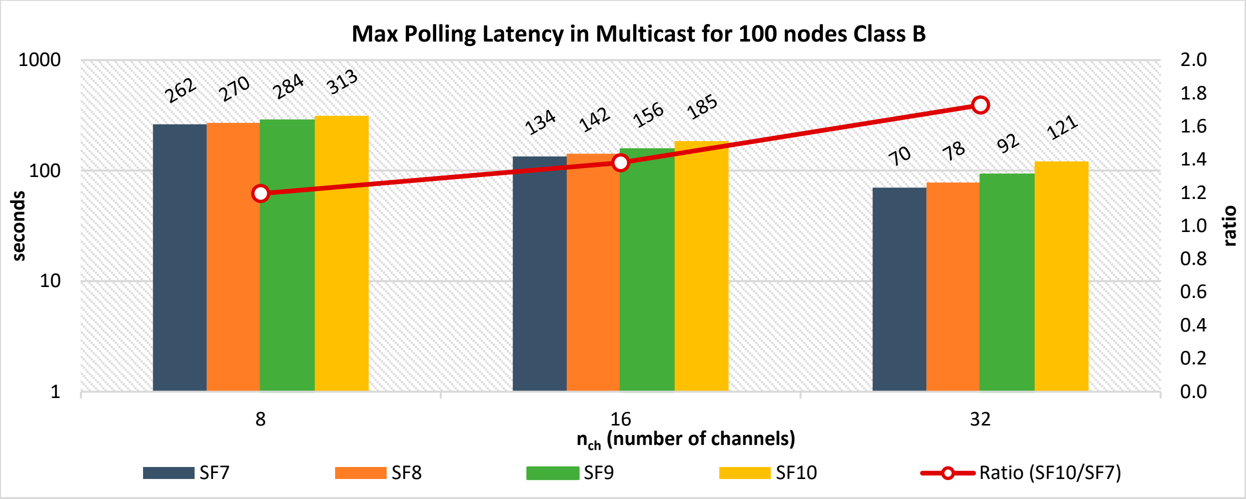


Figure 10. Max latency for Class B unicast and multicast (eight channels) polling depending on network size.

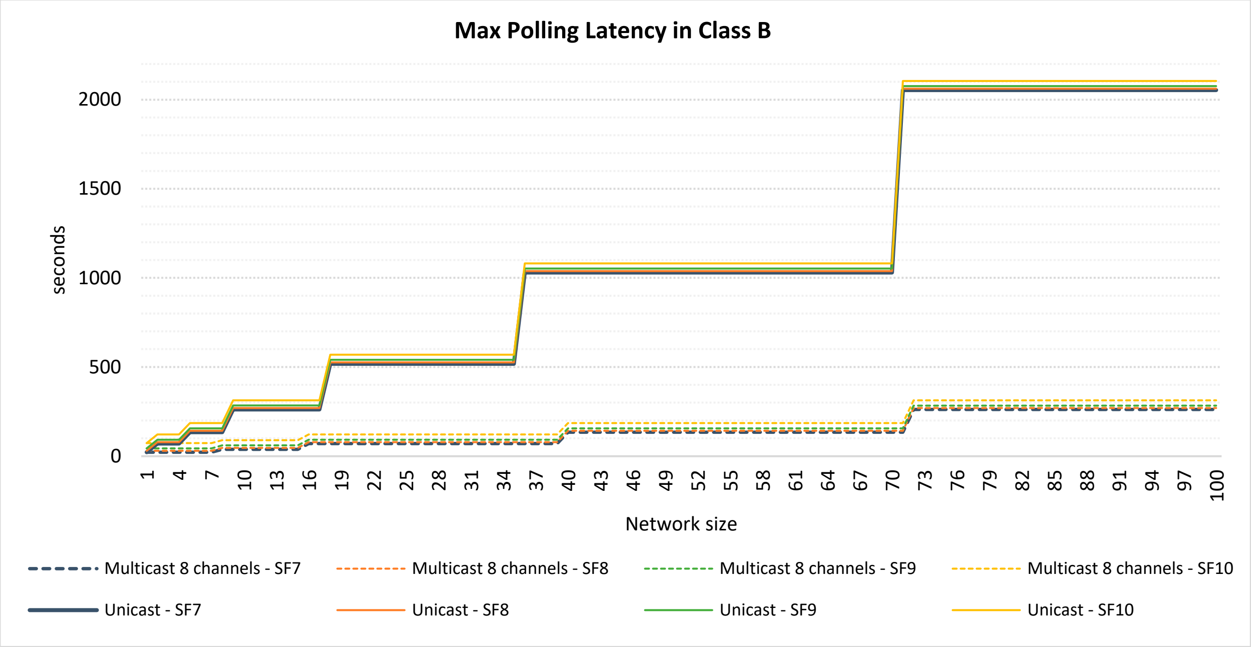


Figure 11. Impact of number of channels in polling delay for a multicast network of 100 nodes in Class B.

4.2. Class C Polling Results

The random behavior in downlink slot allocation found on the Class B operation is not an issue in this case, so the results in Figure 12 show the exact duration in seconds of the polling period. Because there are no possible idle times between scheduled slots (as introduced by the Class B scheduling algorithm), and no restriction on minimum period to guarantee PDR, the calculated periods are not discrete and they show a lineal increase proportional to the network size.

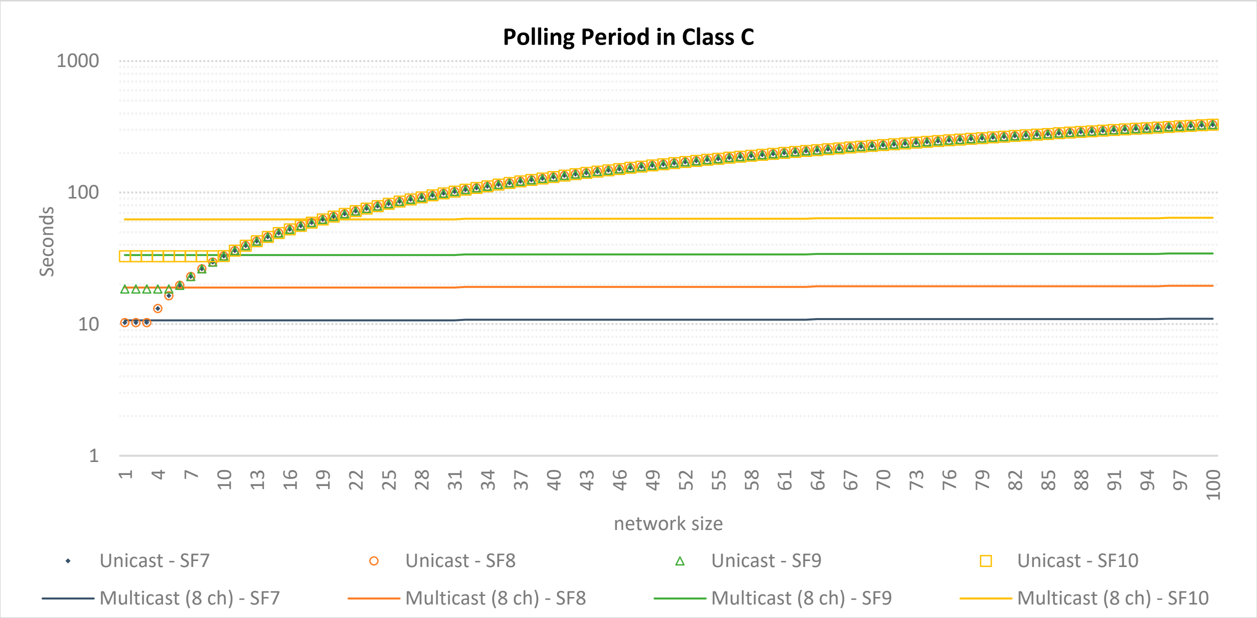


Figure 12. Period in seconds for Class C unicast and multicast (eight channels) polling, depending on network size.

It can also be seen that, for network sizes over 10 devices, the impact of the SF on uplink messages is not relevant, because as there are more nodes to poll from the gateway, the limiting factors are the ToA and DC of the gateway.

Channel-wise, the impact of varying n_{ch} in this multicast polling on Class C is shown in Figure 13. In this case, the impact of using more again greatly reduces latency, with the reduction being proportional to the one seen in Class B. As Class C is from the lower latencies achieved at the start, the multicast Class C polling solution with n_{ch} of 32 is the fastest data collection configuration tested, achieving a latency of 11 s, which can be considered fast enough for some critical IIoT applications, with sensors that have slow responses, such as temperature or humidity for instance. The ratio in latency between devices with different SFs based on the number of channels available is greater than in Class B polling, although it decreases when more channels are available.

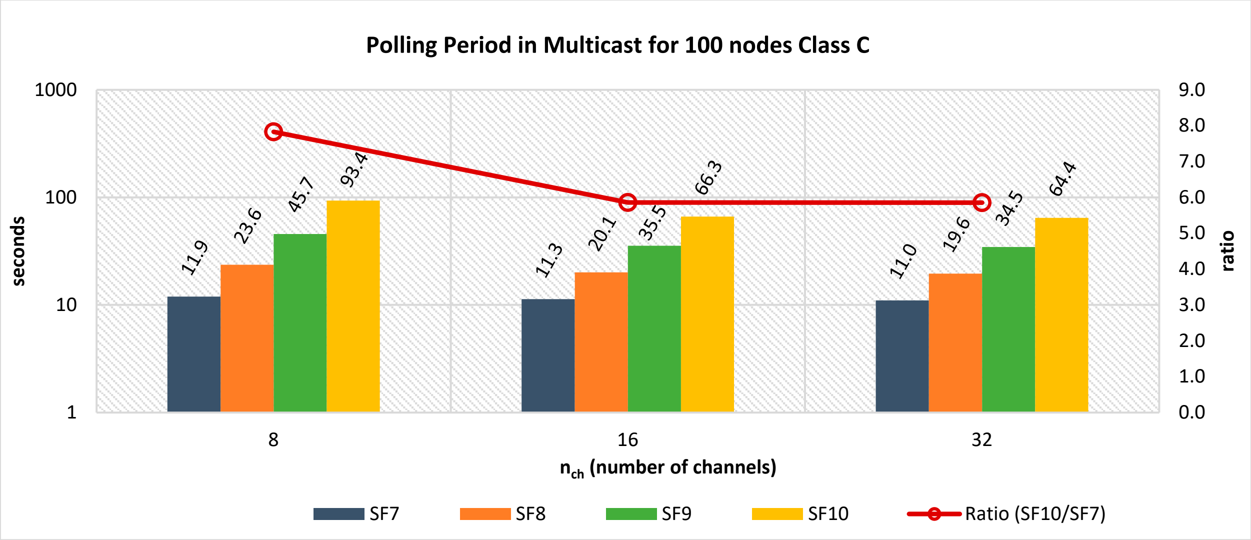


Figure 13. Impact of number of channels in polling delay for a multicast network of 100 nodes in a Class C.

4.3. Energy Consumption Analysis

Based on the capacity of the selected 3000 mAh battery, Figure 14 shows the expected operating lifetime of a node, depending on the class and SF used. On the one hand, as expected, Class C polling requires much more energy because the nodes' radio is constantly active, which also means that the use of a different SF or n_{ch} is not relevant. This results in duration of around 2 days, considering only radio and CPU consumption (real duration will depend on usage of sensors, screen, buttons, or other added components to the device).

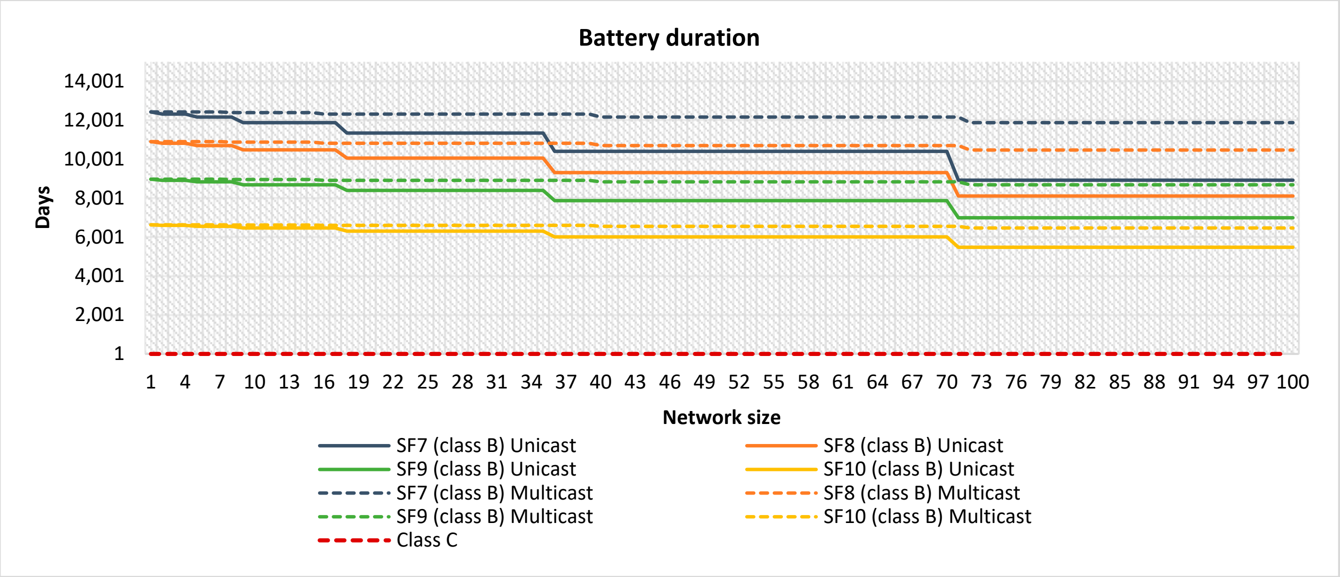


Figure 14. Proposal of battery durations for Class B and Class C polling systems.

On the other hand, the possible configurations on Class B can be translated into battery durations from 15 to 34 years, values that are not realistic due the energy consumed by other device components or even to the natural degradation of battery materials. For reference, various device manufacturers specify battery durations of up to 2 or 3 years when network traffic is around 24 uplink packets per day in Class A.

In summary, the relevant result is the ability to extend the operation time of a node drastically by using Class B polling configurations, which can ensure several months of unattended service.

5. Conclusions and Future Work

As LPWANs become more widespread and as their bandwidth increases, polling will become a more viable option for collecting data from a large number of devices. Additionally, the development of new techniques for reducing the bandwidth overhead of polling will make it even more efficient.

Focusing on the LoRaWAN standard, up to now, applications were mainly focusing on uplink traffic. Trying to export polling as the data acquisition mechanism finds the challenge to optimize the downlink traffic, which is required to send the polling messages to nodes, while guaranteeing that uplink messages will arrive in time. This optimization becomes harder when network size or traffic load may be changing constantly, with periods when new nodes connect, and old nodes run out of batteries, alarm events are detected and trigger more frequent messaging, to provide some examples.

To enable a polling system over LoRaWAN that can adapt to these requirements and ways of operation, configuring nodes with Class B or Class C was proposed and tested, with each option with unicast or multicast scheduling. The results obtained can help understanding the limits of this technology in terms of latency (or polling periods) to collect data and battery lifetime information. This information enables the network server to organize the network to fulfill different application requirements: when latency is not a hard limitation, polling in Class B offers the best energy/latency tradeoff, ensuring a high PDR and many months of operation, while polling in Class C offers unbeatable shorter latencies at the cost of requiring being mains-powered, or lasting only a few days, or less, whilst connected. The multicast option shows better results overall, as expected; however, this study considers also unicast configurations as an option for legacy devices or cheaper gateway options (the more channels available in the interface, the more expensive the device). In conclusion, the decision between LoRaWAN Class B and Class C is contingent upon the specific demands of the application and the imperative considerations of power consumption. Class B stands out as the optimal choice for scenarios where meticulous energy management is paramount and data transmissions occur intermittently. Its ability to schedule receive slots ensures efficient use of power resources, making it particularly well suited for applications like remote monitoring where periodic data reporting is the norm.

On the other hand, Class C emerges as the preferred option in contexts where instantaneous responsiveness takes precedence over prolonged battery life. Its continuous polling capability facilitates real-time communication, making it indispensable for applications demanding low latency, such as security systems and dynamic asset tracking in fast-paced environments with consistently high data traffic.

An optimal solution to benefit from both classes and polling modes relies on the possibility of changing modes on the fly. The network server can force nodes to operate in one class or another, depending on the conditions, switching between Class B and Class C. Ultimately, the coexistence of LoRaWAN Class B and Class C is a testament to the protocol's versatility, allowing for tailored solutions based on the unique requirements of diverse IIoT polling applications. As the landscape of IIoT continues to evolve, the sensible selection of LoRaWAN classes and polling mechanisms will play a pivotal role in optimizing the balance between energy efficiency and real-time responsiveness in the evolving realm of wireless connectivity.

For future work, testing over a network deployed in a representative scenario with industrial requirements and providing situations for polling class switching will help validate the obtained results in a continuous scenario and characterize better the battery behavior in real hardware acquiring real data.

Author Contributions: Methodology, J.S.-B., V.S.-P. and S.S.-C.; software, D.T.-F.; validation, J.S.-B., V.S.-P. and D.T.-F.; formal analysis, D.T.-F.; writing—original draft preparation, D.T.-F.; writing—review and editing, D.T.-F., J.S.-B., V.S.-P. and S.S.-C.; supervision, J.S.-B. and V.S.-P. All authors have read and agreed to the published version of the manuscript.

Funding: The research leading to these results was funded by grant nos. PID2021-123168NB-I00 MCIN/AEI/10.13039/501100011033 by ERDF—"A way of making Europe" and by the Horizon Europe Framework Programme of the European Commission under grant agreement no. 101058589 "AI Powered human-centred Robot Interactions for Smart Manufacturing (AI-PRISM)".

Data Availability Statement: The data are available upon request from the corresponding authors. Requests for data access will be evaluated on a case-by-case basis, and data will only be shared with researchers who have a legitimate need for the data and who agree to protect the privacy of this research. To request access to the data, please contact the corresponding authors and provide a brief description of your research project and how you plan to use the data.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

5G	Fifth generation of wireless cellular technology
ABPC	Adaptive Beacon Period Configurator for scalable LoRaWAN
ADR	Adaptive Data Rate
AMR	Automatic Meter Reading
AS	Application server
ATSA	Acyclic Traffic Scheduling Algorithm
b_g	Beacon guard time in Class B slot frame
b_r	Beacon reserved in Class B slot frame
BW	Bandwidth
Class A	MAC access mode in LoRaWAN, focused on uplink
Class B	MAC access mode in LoRaWAN, with scheduled downlinks
Class C	MAC access mode in LoRaWAN, with radio always enabled for reception
CPS	Cyber Physical System
CSS	Chirp Spread Spectrum, the LoRa radio modulation
DC	Duty cycle, depending on regional fair radio access policies
DR	Data rate
ETSI	European Telecommunications Standards Institute
IEEE	Institute of Electrical and Electronics Engineers
IIoT	Industrial Internet of Things
IP	Internet protocol
ISA100.11a	Wireless Systems for Industrial Automation, Process Control, and Related Applications by the International Society of Automation
LPWAN	Low-power wide-area network
LoRaWAN	Long-range wide-area network
NS	Network server
MAC	Medium Access Control
mMTC	Massive Machine Type Communications
OFDMA	Orthogonal Frequency-Division Multiple Access
OMS	Outage Management System for AMR networks
PDR	Packet Delivery Ratio
RSIN	Rate Selection for Industrial Networks
SF	Spreading factor
TCP	Transmission Control Protocol
ToA	Time on air
TRILO	Traffic Indication-based Downlink Communication Protocol for LoRaWAN
T_s	symbol duration
URLLC	Ultra-Reliable Low-Latency Communications
Wi-Fi	Wireless Fidelity, set of standards for WLANs
Wireless HART	Wireless Highway Addressable Remote, an WLAN industrial standard
WLAN	Wireless Local Area Network

References

1. Peserico, G.; Morato, A.; Tramarin, F.; Vitturi, S. Functional Safety Networks and Protocols in the Industrial Internet of Things Era. *Sensors* **2021**, *21*, 6073. [CrossRef]
2. Fernandes, P.; de Lemos Dinis, V.M.; Briones Peñalver, A.J.; Madeira Esteves, F.J. Functional Safety as a critical success factor to industry 4.0. *Procedia Comput. Sci.* **2022**, *204*, 45–53. [CrossRef]
3. Al-Mefleh, H.; Al-Kofahi, O. Frequency-domain contention and polling MAC protocols in IEEE 802.11 wireless networks: A survey. *Comput. Commun.* **2018**, *129*, 1–18. [CrossRef]
4. Tramarin, F.; Vitturi, S.; Luvisotto, M. A dynamic rate selection algorithm for IEEE 802.11 industrial wireless LAN. *IEEE Trans. Ind. Inform.* **2017**, *13*, 846–855. [CrossRef]
5. Xu, C.; Yu, H.; Zeng, P.; Li, Y. Towards Critical Industrial Wireless Control: Prototype Implementation and Experimental Evaluation on URLLC. *IEEE Commun. Mag.* **2023**, *61*, 193–199. [CrossRef]
6. Hoyningen-Huene, J.V.; Mueller, A. Scheduling acyclic traffic in cyclic and deterministic communication systems. In Proceedings of the 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS), Trondheim, Norway, 31 May–2 June 2017; pp. 1–9. [CrossRef]

7. Zheng, M.; Lin, J.; Liang, W.; Yu, H. A priority-aware frequency domain polling MAC protocol for OFDMA-based networks in cyber-physical systems. *IEEE/CAA J. Autom. Sin.* **2015**, *2*, 412–421. [CrossRef]
8. Igarashi, Y.; Nakano, R.; Wakamiya, N. A Polling-Based Transmission Scheme Using a Network Traffic Uniformity Metric for Industrial IoT Applications. *Sensors* **2019**, *19*, 187. [CrossRef] [PubMed]
9. Garrido-Hidalgo, C.; Haxhibeqiri, J.; Moons, B.; Hoebeke, J.; Olivares, T.; Ramirez, F.J.; Fernández-Caballero, A. LoRaWAN Scheduling: From Concept to Implementation. *IEEE Internet Things J.* **2021**, *8*, 12919–12933. [CrossRef]
10. Oh, Y.; Lee, J.; Kim, C.-K. TRILO: A Traffic Indication-Based Downlink Communication Protocol for LoRaWAN. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 6463097. [CrossRef]
11. Reynders, B.; Wang, Q.; Tuset-Peiro, P.; Vilajosana, X.; Pollin, S. Improving Reliability and Scalability of LoRaWANs through Lightweight Scheduling. *IEEE Internet Things J.* **2018**, *5*, 1830–1842. [CrossRef]
12. Samuhasilp, J.; Pora, W. Development of an Automatic Meter Reading and Outage Management System using LoRaWAN Technology. In Proceedings of the 2018 IEEE 5th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA), Songkhla, Thailand, 28–30 November 2018; pp. 1–4. [CrossRef]
13. Semtech Application Note AN1200.22. 2015. LoRa Modulation Basics. Available online: http://wiki.lahoud.fr/lib/exe/fetch.php?media=an1200.22.pdf (accessed on 23 October 2023).
14. Bartolín-Arnau, L.; Todoli-Ferrandis, D.; Sempere-Payá, V.; Silvestre-Blanes, J.; Santonja-Climent, S. LoRaWAN Networks for Smart Applications in Rural Settings. *IETE Tech. Rev.* **2023**, *40*, 440–452. [CrossRef]
15. LoRa Alliance, Notice of Use and Disclosure. 2017. LoRaWAN 1.1 Specification. Available online: https://net868.ru/assets/pdf/LoRaWAN-v1.1.pdf (accessed on 23 October 2023).
16. LoRa Alliance, LoRaWAN 1.1 Regional Parameters. 2017. Available online: https://lora-alliance.org/wp-content/uploads/2020/11/lorawan-regional-parameters-v1.1ra.pdf (accessed on 23 October 2023).
17. LoRaWAN Remote Multicast Setup Specification v1.0.0. 2018. Available online: https://lora-alliance.org/wp-content/uploads/2020/11/remote_multicast_setup_v1.0.0.pdf (accessed on 23 October 2023).
18. Todoli-Ferrandis, D.; Silvestre-Blanes, J.; Sempere-Payá, V.; Santonja-Climent, S. Adaptive Beacon Period Configurator for Scalable LoRaWAN Downlink Applications. *IEEE Access* **2023**, *11*, 83627–83638. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.