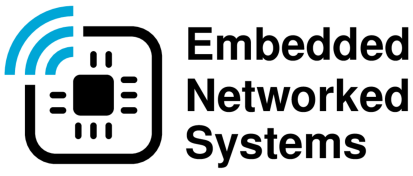
Delft University of Technology
Master's Thesis in Embedded Systems

LoRaWAN Class B Multicast: Scalability

Yonatan Woldeleul Shiferaw



TU Delft Delft University of Technology

LoRaWAN Class B Multicast: Scalability

Master's Thesis in Embedded Systems

Embedded and Networked Systems Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Yonatan Woldeleul Shiferaw
yoniwt@gmail.com

19th September 2019

Author
Yonatan Woldeleul Shiferaw (yoniwt@gmail.com)
Title
LoRaWAN Class B Multicast: Scalability
MSc presentation
26th September 2019

Graduation Committee

Dr. ir. Fernando Kuipers (Chair)	Delft University of Technology
Dr. ir. Jos Weber	Delft University of Technology
Dr. ir. Jos Adema	KPN

Abstract

LoRaWAN is amongst the most prominent LPWAN (Low Power Wide Area Networking) technologies and provides an operation mode, called Class B, where devices can be reached with limited latency. One of the most important features of LoRaWAN Class B is multicast, which allows a single transmission to be shared by multiple multicast group member end-devcies. This enables key IoT functionalities such as FOTA (Firmware Over The Air) and other applications that need to transfer multiple packets to a set of end-devices. This thesis will delve into LoRaWAN Class B multicast and study its scalability – the most important behavior to study in all IoT technologies. An ns-3 module for LoRaWAN Class B multicast has been extended and used for in-depth analysis and drawing scalability insights.

The study is divided into two major categories: multicast group scalability and multicast member scalability. Beacon blocking, overhearing and collision, and average throughput performances are studied as part of the group scalability problem, and different techniques are suggested to avoid beacon blocking. On the other hand, under multicast member scalability, the study covers multicast capacity (the main focus), Class A coexistence with Class B, and multicell multicast scalability. To support the duty cycle limited gateways in multicast transmission and improve the multicast capacity, an enhancement to the LoRaWAN Class B multicast protocol called ping-slot relaying has been introduced and evaluated. It is compatible with the standard LoRaWAN Class B and does not need any change on the network-server.

Acknowledgment

All my praise and thanks goes to my Lord, my King and my savior Jesus Christ without whom not a single day would have been as meaningful as today. He is and will remain to be my greatest treasure, because life was never the same after knowing him personally.

I want to thank my supervisor, Dr. ir. Fernando Kuipers, for his advise and guidance. He is calm, patient and very flexible in his approach. He identifies the weaknesses and the strengths of his students and help them work accordingly.

I would like to extend my sincerest thanks to Dr. ir. Jos Adema and Dr. ir. Jos Weber for taking there time to be in my thesis committee. I would also like to thank several people who helped me and motivated me during my thesis. Peirre Dufour has helped me during the initial brainstorming. My colleague Alexander van Gessel was never reluctant to offer technical help in times of need. My friend Anup Bhattacharjee was always eager to listen to my ideas and engage in fruitful discussions.

I can't thank enough my friend Samuel Aklilu Mezmure, who is more of a brother, for all his support. He was emotionally involved with this project even as I am. He has rejoiced in my success and engaged in my toil.

I am really grateful for the continues prayer and encouragement of Muluken Amare in various times of need.

Let me express my gratitude to my father Woldeleul Shiferaw, my mother Tigist Abera and all my family for their support. As always, they have been concerned and was eagerly waiting for the completion of this work.

Finally, many thanks to Delft Global initiative for the Scholarship which has enabled me to undertake my study in TU Delft.

Yonatan Woldeleul Shiferaw

Delft, The Netherlands
19th September 2019

Contents

Chapter 1

Introduction

1.1 Motivation

The Internet of Things (IoT) allows to connect myriad of devices in an effort to collect information and take action with minimal human interaction. According to the World-Economic-Forum, IoT is a promising game changer for sustainability and 84% of current IoT applications address UN's sustainable development goals (SDGs) [1].

Following the innovation of a communication technology called LoRa with a range that extends to kilometers, LoRaWAN became one of the leading standards for low-power IoT device connectivity. It finds its application in smart cities, smart industries, smart utilities, smart homes, smart healthcare, smart agriculture and other smart sectors where battery power and range are desired over data-rate. The LoRa-Alliance – the technology alliance behind this standard – has more than 500 members and is expected to grow in the coming years [2].

With a rise in interest for Firmware Over The Air (FOTA), which allows IoT end-devices to be reconfigured, updated, or even re-purposed remotely, the ability to support such feature became incumbent for the long-term success of a communication technology that is targeted towards IoT. For massive number of end-devices this requires a specific ability for a communication technology to be able to deliver an identical packet to the desired end-devices simultaneously – called multicasting. Otherwise, as the number of end-devices increases – which has been the case for IoT end-devices – the amount of time spent in transferring a sequence of packets to all end-devices will grow enormously. On top of FOTA, multicast is also important for several applications that benefit from simultaneous transmission to multiple end-devices: street lighting, gas and water distribution valves, and pagers are a few examples of such applications.

By the same token, LoRaWAN has to support multicasting for its success as an IoT standard. Since LoRaWAN operates in free spectrum called the

ISM band and is subject to transmission regulations, such a feature is not only useful but also indispensable. In line with this, LoRaWAN provides 2 additional operation classes – Class B and Class C – in the protocol definition that support multicasting besides its default and mandatory class of operation, Class A. Among these classes, only one class – called Class B – is suitable for battery-operated devices, which IoT end-devices typically are.

1.2 Problem Statement / Thesis Objective

LoRaWAN offers 3 different operation classes which are targeted for different applications depending on their latency and energy requirements: Class A, Class B, and Class C. Class A allows end-devices to save the most energy by allowing them to wake up only when they have to send a packet. The downside of this class is that it is not possible to send a packet to an end-device unless the end-device sends an uplink – incurring unpredictable downlink delay. Class C is the other extreme where a network can always send a downlink to an end-device, but requires the end-device to always listen except when it is sending uplinks. This makes class C not suitable for battery-operated devices. The remaining class – Class B – provides a fair trade-off between these two classes. It offers a predictable downlink latency while allowing the end-device to still save energy. Class B accomplishes this by synchronizing the end-devices and the LoRaWAN network and allowing the end-devices only to wake up and check for downlinks periodically. If there are no ongoing downlinks, the end-device again goes to sleep until the next period to save energy even more. This renders Class B suitable for multicast applications.

Nevertheless, until now, there hasn't been any research on LoRaWAN Class B multicast and there only has been little work on LoRaWAN Class B in general. The authors believe that a work on LoRaWAN Class B multicast will open doors for future research in improving the protocol, while at the same time offering insights for industries and operators that use the technology. The goal of this thesis will be exploring various challanges of LoRaWAN Class B multicast scalability. This breaks down to two main challanges: group scalability and member scalability.

The first challenge, multicast group scalability is similar with a Class B unicast[1] end-device scalability problem, except the performance is now shared with all member end-devices of a group. In this thesis, we will study the beacon-blocking behaviour, overhearing and collision problem, and average group throughput. Our study will start by first re-evaluating beacon blocking for all ping-slot periodicities and all relevant data-rates in Europe. We will then look into the overhearing and collision problem. Finally, we will see the average throughput performance as the number of groups grow.

[1] In unicast a transmission targets a single receiver as opposed to multicast.

The second challenge of multicast scalability is member scalability. As the number of end-devices in a multicast group increases, the success of a multicast transmission will degrade. This is because the packet loss that the end-devices experience varies with space – as the channel typically varies with space – which all together may require a huge proportion of already transmitted packets to be retransmitted although they are received by some of the members. This problem gets worse as the number of end-devices increases or simply as the members of a multicast group scale. In this thesis, we will explore how we can exploit the scaling of the multicast members to ameliorate the degradation of the multicast success using a ping-slot relaying technique that does not require a change in the network-server. In addition, we will also see how member scaling affects the issue of LoRaWAN Class B coexistance with LoRaWAN class A along with different suggestions to resolve potential conflicts. We conclude by briefly explaining the multicell multicast member scaling problem which happens when a multicast group has member end-devices in more than one cell.

1.3 Contributions

The following are the key contributions of this thesis:

- We develop a LoRaWAN Class B multicast module for ns-3.
- We study LoRaWAN Class B multicast group scalability with the help of the simulator. To be more specific:
  - We re-evaluate beacon blocking behavior for all ping-slot periodicites as number of multicast groups (unicast end-devices) scale.
  - We suggeste techniques that can be used to prevent beacon blocking.
  - We analyze the overhearing and collisions as the multicast group (unicast end-devices) scales.
  - We analyze the average throughput per group (per end-device in case of unicast) as the multicast group (unicast end-devices) scales.
- We study LoRaWaN Class B multicast member scalability. To be more specific:
  - We simulate and evaluate the LoRaWAN Class B multicast capacity.
  - We propose and evaluate a LoRaWAN Class B multicast compatible technique called ping-relaying to enhance LoRaWAN Class B multicast when a transmitting gateway is in its time-off period.

- We discuss the problem of Class B coexistence with Class A in detail.
- We discuss different options to resolve the the conflict between the Class A uplinks and Class B ping-slots.
- We look into the multicell multicast member scaling problem, which is common for multicast groups that have members in different cells.

1.4 Organization / Thesis Outline

The rest of the thesis is organized in 5 chapters. Chapter 2 will give background information on LoRaWAN, LoRaWAN Class B and LoRaWaN Class B multicast. Related works in the context of LoRaWAN Class B will also be discussed in this chapter. Chapter 3 will explain about the ns-3 LoRaWAN Class B multicast module. Chapter 4 will be discussing about multicast group scalability and delve into the beacon blocking behaviour, beacon blocking prevention techniques, overhearing and collision performance as well as average group throughput. Chapter 6 will discuss multicast member scalablity and explore the multicast capacity and a technique that uses the unused ping-slots to enhance the capacity. It will also look into the LoRaWAN Class B coexistance with Class A along with some possible conflict resolution methods for the coexistence. The chapter concludes with the multicell multicast member scaling problem. In the final chapter, chapter 6, the we will summarize our discussions and indicate a direction for future work.

Chapter 2

Background and related work

This chapter provides background information on LoRaWAN and multicasting. First, the LoRaWAN protocol is briefly explained with a special emphasis on LoRaWAN Class B, which is the focus of this thesis. Then, multicasting is explained in context of LoRaWAN, again concentrating on LoRaWAN Class B. The chapter is concluded with related work on LoRaWAN Class B.

2.1 LoRaWAN

LoRaWAN (Long Range Wide Area Network) is an LP-WAN (Low Power Wide Area Network) MAC (Medium Access Control) and higher-layer protocol that operates on top of LoRa (Long Range) – a physical-layer protocol developed by Cyclo and acquired by Semtech [3]. LoRa is derived from chirp spread spectrum modulation [4] and it has a higher link-budget compared to conventional narrow-band modulation [5]. This makes it possible for LoRaWAN to operate in a star topology with a range that extends to kilometers.

The LoRaWAN network-architecture consists of end-devices (nodes), gateways (concentrators), network-server and application-servers as shown in Fig. 2.1 [6]. The gateways provide the RF interface between the network-server and the nodes; they forward received uplink packets from the nodes to the network-server and send downlink packets to the nodes. The network server implements the LoRAWAN protocol and provides the interface to the application servers.

LoRaWAN operates in the unlicensed spectrum. This spectrum is subject to regulations and transmissions should therefore adhere to these regulations. In line with this, [7] specifies the operation of LoRaWAN for different regions. In Europe, LoRaWAN operates in the EU433 band (433.05 - 434.79

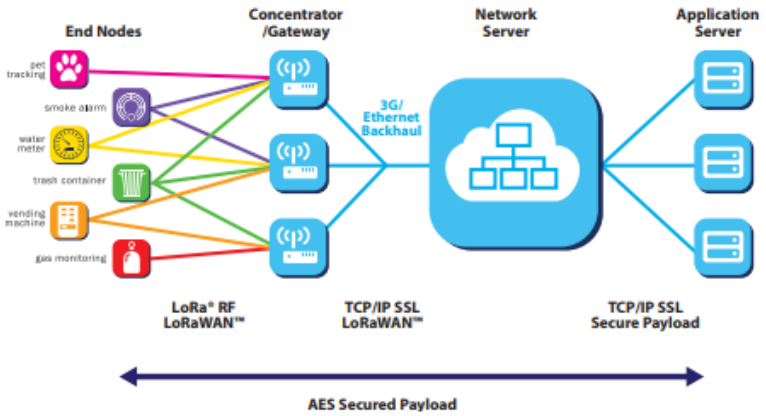


Figure 2.1: LoRaWAN Network Architecture [6].

MHz) and in the EU863-870 band (863 - 870 MHz), where EU868-870 is the most common band. Table 2.1 gives the duty cycle and the transmit power regulations that apply to the EU868-870 band, as defined by ETSI [8] [9] for 125KHz bandwidth.

Band	Edge Frequencies	Duty Cycle	Max Power
g (Note 7)	865-868MHz	1% or LBT AFA	+6.2dBm /100kHz
g1	868-868.6MHz	1% or LBT AFA	14dBm
g2	868.7-869.2MHz	0.1% or LBT AFA	14dBm
g3	869.4-869.65MHz	10% or LBT AFA	27dBm
g4	869.7-870MHz	No Requirement	7dBm
g4	869.7-870MHz	1% or LBT AFA	14dBm

Table 2.1: ETSI regulations for different sub-bands.

The LoRaWAN protocol has three basic classes of operation: **Class A**, **Class B** and **Class C**. The fundamental difference in these classes of operation is in how the gateways send downlink packets to the nodes; the way in which the nodes send uplink packets to the gateways is essentially the same, as both Class B and Class C are extensions of Class A [10]. These downlink and uplink packets carry application-payloads or MAC-commands. A **MAC-command** is a special payload provided by LoRaWAN for smooth operation of remote nodes: for connecting over-the-air, for remotely changing parameters, for retrieving required information and for sending confirmations [10]. The following sections will briefly explain the operation of the three LoRaWAN classes along with some of the MAC commands they use.

2.2 LoRaWAN Class A

Class A is a mandatory mode of operation for all LoRaWAN nodes; a switch to the other operation modes is only allowed after entering this mode. It is the most energy-efficient mode of operation of all the LoRaWAN classes. Such efficiency is accomplished by allowing the device to wake up only when it needs to send an uplink. As one can see from Fig. 2.2, to accommodate downlinks, the device opens two receive windows – namely Rx1 and Rx2 – with Rx_Delay_1 (typically of one second) and Rx_Delay_2 (typically of two seconds) delay respectively, which can be adjusted using the RXTiming-SetupReq/Ans MAC command. If the network-server has additional data queued for the node, it can request the node to send an uplink as soon as possible by placing one in the current downlink's FPending field [10].

Although class A is the most energy-efficient mode of operation for most sensor applications, it has the worst performance when it comes to downlink latency. The latency between two consecutive downlinks is approximately the same as two consecutive uplinks from the node – which is not under the control of the network-server.

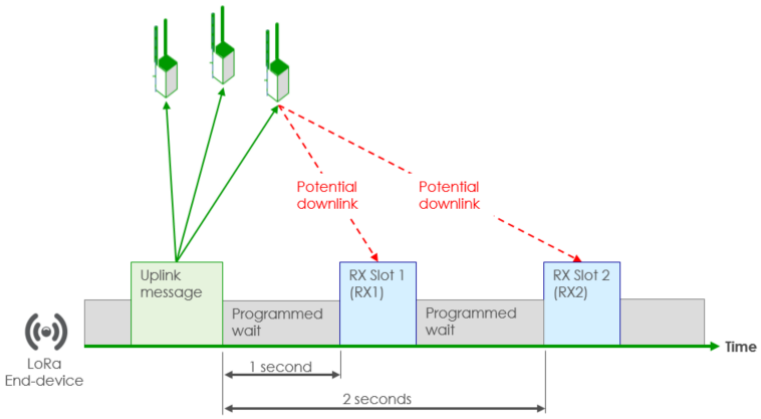


Figure 2.2: Class A operation [11].

2.3 LoRaWAN Class C

Class C, as opposed to class A, is the most power hungry mode of operation – but with the least downlink latency. As can be seen from Fig 2.3, in this mode of operation, the Rx2 window is always open except when it sends an uplink and when it opens the Rx1 window following the uplink [10]. This mode of operation therefore leverages its power, to guarantee a reception window at any time except when the device is sending an uplink.

As a result, this mode is suitable for an application with a strict timing requirement and direct-powered nodes such as smart IoT Actuators.

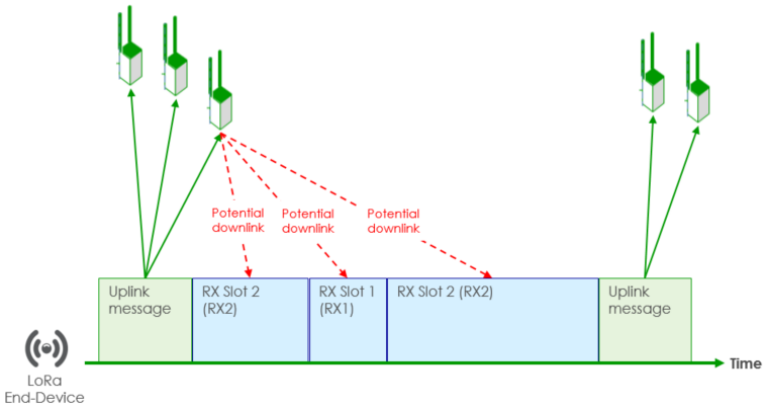


Figure 2.3: Class C operation [11].

2.4 LoRaWAN Class B

The final mode of operation, class B, is the most modest of all as it offers a predictable downlink latency with a moderate energy consumption. It is ideal both for battery powered devices and for energy harvested devices if most of the communication involves downlinks that have bounded latency requirements.

As one can see from Fig. 2.4, class B does not require a node either to send an uplink for each downlink – unlike class A – or to open a continuous receive window – unlike class C. Instead, class B defines periodic receive windows on the end-devices called **ping-slots** [10] to allow the network-server to send downlinks without the need to wait for uplinks. However, such a communication is not possible if the gateways and the end-devices are not synchronized; that is why the network-server sends periodic downlink packets known as **beacons** [10], which accomplishes this synchronization.

2.4.1 Beacon

Beacons are time-synchronized packets transmitted by gateways and received by the end devices every beacon-period (which is 128 seconds and marked as Beacon_period in Fig. 2.5) on the g3 sub-band by default [7]. As synchronization is at the heart of class B operation, a node is allowed to switch to Class B only after it searches and locks a beacon. To save

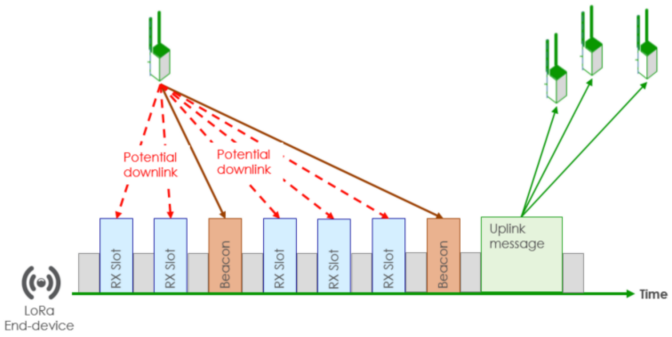


Figure 2.4: Class B operation [11]

energy of the nodes in the search, the LoRaWAN specification provides a DeviceTimeReq/Ans or BeaconTimingReq/Ans MAC command, depending on the LoRaWAN version[1], which facilitates the process.

In order to avoid a conflict between an end-device's uplink transmission and its beacon reception, a beacon is preceded by a guard time (3 seconds), marked as the Beacon_Guard in Fig. 2.5, during which a ping slot can not be placed [10]. The calculation of this guard time is done in such a way that the longest downlink reception will not overlap with the beacon reception. Furthermore, a time is also reserved for the beacon (2.12 seconds), marked in Fig. 2.5 as Beacon_Reserved, in order to account for possible jitter and future extensions. All together, the remaining time in a beacon-period that is available for ping-slots is 122.880 seconds which is referred to as beacon-window (marked as Beacon_window in Fig. 2.5) in the specification.

The received beacon packets contain a 17 byte payload which uses 4 bytes for GPS time, 7 bytes for gateway-specific information and 4 bytes for CRC (2 bytes for the GPS time and 2 bytes for the gateway-specific information). Fig. 2.6 depicts this packet as defined by the LoRaWAN specification [10]: the time field[2] contains the time elapsed – in seconds – since GPS epoch[3], the gateway specific information in most of the cases contains information that corresponds to the GPS coordinates of the transmitter antenna. Separating the CRC for the GPS time and the gateway-specific information allows the nodes to synchronize with the gateways in conditions where the gateway-specific information field is corrupted, by decoding the GPS time separately.

[1]BeaconTimingReq/Ans is used for LoRaWAN 1.0.2B or below, whereas the DeviceTimeReq/Ans is for the versions above LoRaWAN 1.0.2B.

[2]As the time field is 4 bytes long the counter will wrap back to zero when it reaches an integral multiple of 2^{32}.

[3]00:00:00, Sunday 6^{th} of January 1980.

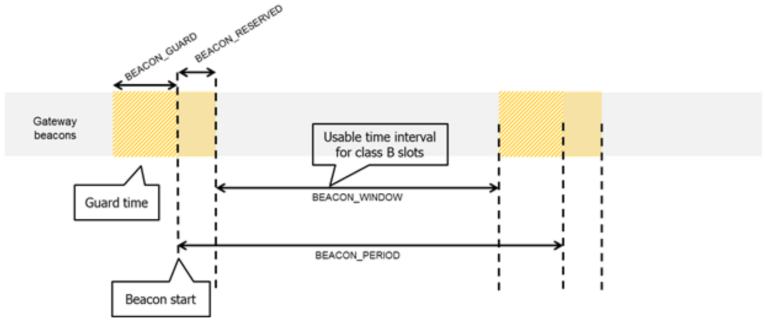


Figure 2.5: Beacon timing [10].

Size (bytes)	2 to 5	4	2	7	0 to 3	2
BCNPayload	RFU	Time	CRC	GwSpecific	RFU	CRC
	Compulsory common part			Optional part		

Figure 2.6: Beacon payload format [10].

2.4.2 Ping Slot

Once a node has successfully locked a beacon, it switches to Class B. That means it starts opening periodic reception windows – ping-slots – to receive downlinks from synchronized gateways. After switching to Class B, the end-devices open these windows for 8 symbols[4] for each locked beacon[5] and goes back to sleep if no preamble is detected although the slot is reserved for 30 ms (**slotLen**). If a preamble is detected during these windows, end-devices will continue to receive until the end of the packet. Fig. 2.7 shows an example of downlink packet reception via the last ping-slot reception window.

The duration between two consecutive ping slots (**pingPeriod**) is found by first expressing the beacon-window in-terms of slots (which will be 2^{12} slots) and then dividing it by the number of pings in a beacon period (called **pingNb**) as shown in equation 2.1. This is important for the slot randomization technique that will be explained in section 2.4.4.

$$pingPeriod(slots) = 2^{12}/pingNb \tag{2.1}$$

The pingNb in-turn is derived from **ping-slot periodicity** (or simply **periodicity**), which is a three bit number that goes from 0 to 7, as shown in equation 2.2. In order for the network server to know when the ping slots

[4]This is not stated in the specification, but extracted from LoRaMac-node firmware [12].

[5]See section 2.4.3 for beacons.

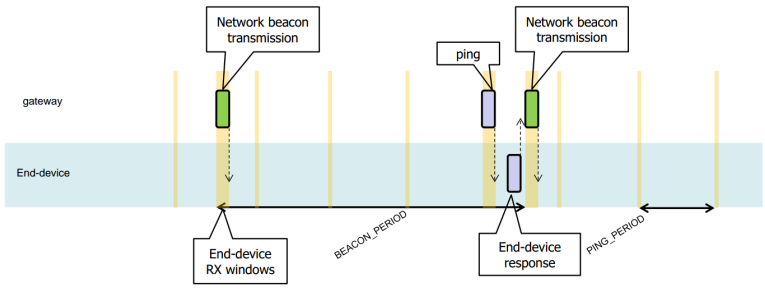


Figure 2.7: A gateway sending a downlink at the final ping slot of a Class B node which has a ping-slot-periodicity of 5 [10].

are opened, the periodicity is communicated to the network server using the PingSlotInfoReq/Ans MAC command.

$$PingNb = 2^{7-Periodicity} \tag{2.2}$$

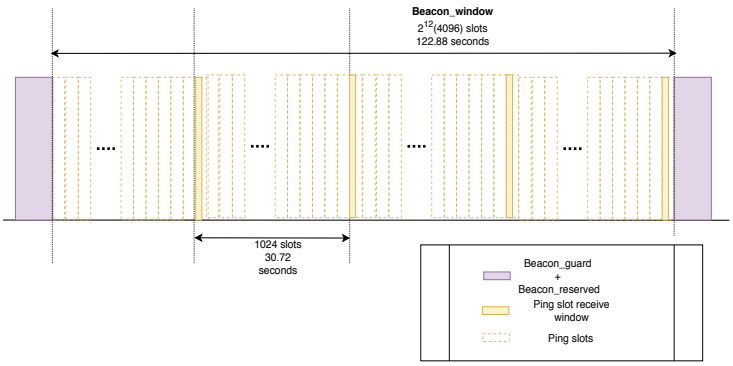


Figure 2.8: Ping slot placement example for a ping slot periodicity of 5 which translates to pingNb of 4 and pingPeriod of 1024 slots (30.72 seconds).

The default transmission configuration used for the ping-slots is 869.525MHz (g3 sub-band) with SF[6] 9 and 125KHz band-width [7]. If desired, the default channel can be modified by gateways using the PingSlotChannelReq/Ans MAC command in accordance with the LoRaWAN Regional Parameters [10].

[6]SF stands for spreading-factor. It is the LoRa physical layer parameters which is the ratio between symbol rate and chip rate [13]. The higher the spreading factor, the longer a transmission takes but the more resilient it is towards noise and interference as well as travels longer distance. On the other hand, the smaller the spreading factor, the shorter a transmission takes but is much more susceptible to noise and interference as well as covers shorter range. In the EU868-870 band, LoRaWAN uses minimum spreading-factor of 7 (SF7) and maximum spreading-factor of 12 (SF12).

2.4.3 Beacon-less Operation and Minimal Beacon-less Operation Time

Once a node has locked a beacon and has switched to class B, it is not obliged to immediately switch back to class A if it starts occasionally losing beacons. Instead, it can use its internal clock from the previous successful beacon reception to open the receive windows for the coming ping and beacon packets. However, as losing beacons will result in internal clock drift – depending on the ppm of the node – the LoRaWAN specification requires the nodes to progressively expand both the ping-slot windows and the beacon windows, to compensate the drift, as shown in Fig. 2.9. This operation is called the "beacon-less" operation and can continue for minimal beacon-less operation time (120 minutes), after which it should switch back to class A if the situation persists [10]. This operation is always reset if a beacon is successfully received.

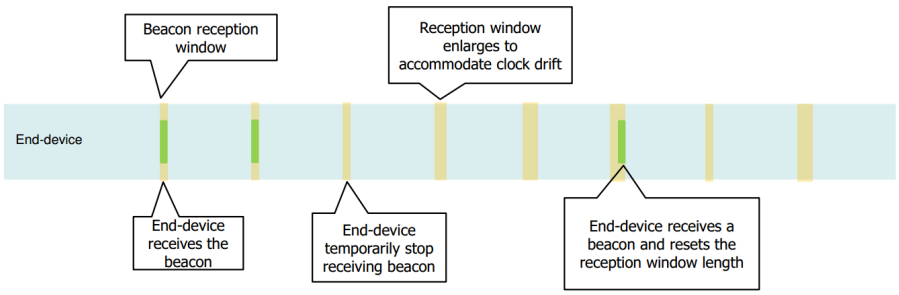


Figure 2.9: Beacon less operation [10].

2.4.4 Overhearing and Systematic Collisions

Collision and overhearing are among the major problems that hinder LoRaWAN Class B performance: collision hinders a successful packet reception and overhearing drains a node's battery. Collisions occur when two or more gateways are trying to send a Class B downlink to nodes that reside in their intersecting vicinity. Depending on the time offset and the power difference between the interfering signals, such collision can lead to reception error [14] [15]. This is not a one time incident, but continues at each beacon period as long as they have the same periodicity and configuration. Hence, this is called a systematic collision. On the other side, overhearing happens when different Class B nodes open their corresponding reception windows at the same time; the packet that is sent for one node will be dropped by the others only after the receiving nodes decode the packet – which wastes energy for

each packet that is received and yet discarded. The effects of these two problems are even more aggravated as the payload length and the SF increase. In order to alleviate this problem, the LoRaWAN specification uses a technique called slot-randomization. In this technique, a random offset (**pingOffset**) is calculated as a function of both the beacon time (for the corresponding beacon period) and the end-devices devAddr[7] and added to the start of the first ping for each beacon period. This will minimize the collision between different devices[8], because their devAddr varies. Furthermore, the systematic collision or overhearing is alleviated because the beacon time for each beacon period is different.

In order to align the reception slots with the ping transmissions both the end-device and the network-server calculate the pingOffset for each beacon period as shown in equation 2.5 using AES encryption [10].

$$Key = \texttt{0x00000000000000000000000000000000} \tag{2.3}$$

$$Rand = aes128\_encrypt(Key, beaconTime|DevAddr|pad16) \tag{2.4}$$

$$pingOffset = (Rand[0] + Rand[1] \times 256) \mod pingPeriod \tag{2.5}$$

Fig. 2.10 shows an example of ping slot randomization for two end-devices with DevAddr1 and DevAddr2 during two different beacon periods with BeaconTime1 and BeaconTime2 respectively. As can be seen from the figure, although the two end-devices have the same class B configuration, the ping slot randomization technique avoided the overlap that would otherwise happen.

2.5 Multicasting in LoRaWAN Class B

Multicast is a communication technique which can be used to simultaneously deliver a packet to multiple destinations. In wireless communication, this is a matter of aligning targeted end-devices to listen at the same time [16] as well as share the keys and addresses [10] required to receive a transmitted packet due to its broadcast nature. In LoRaWAN, such synchronized listening can be accomplished either via Class C[9] or Class B because both classes allow time-constrained and server-initiated downlinks [10].

In order to achieve multicast in LoRaWAN Class B, all the nodes in a multicast-group[10] should listen and decode the multicast transmissions at

[7] Or multicast devAddr in case of multicast.

[8] Or multicast groups in case of multicast.

[9] Since class C end-devices are always listening and they are not suitable for battery operation we only consider class B end-devices for this thesis.

[10] A multicast group is a virtual target that contains a set of devices to which the multicast transmission is targeted.

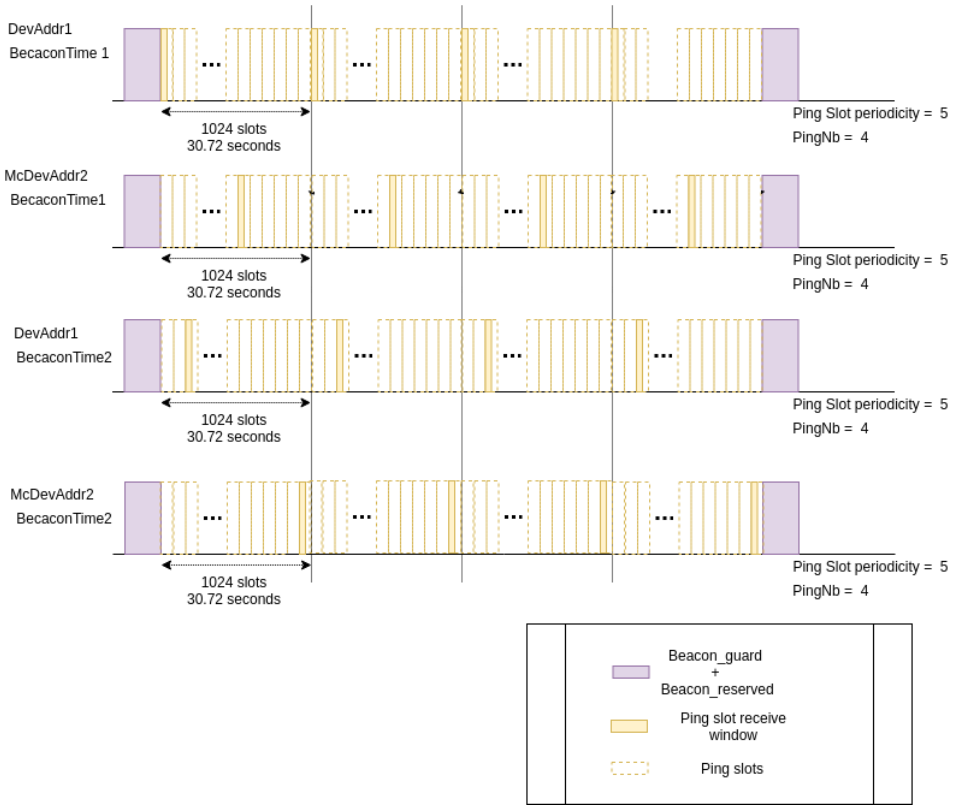


Figure 2.10: Ping slot placement for two end-devices with device address devAddr1 and devAddr2 whose pingOffset is calculated to be 0 and $pingPeriod - 6$ for $BeaconTime1$ and whose pingOffset is calculated to be 2 amd $pingPeriod - 1$ for $BeaconTime2$.

the same time. This can be realized by assigning the nodes in a multicast-group with the same configuration[11], address[12], and security keys[13], as synchronization is already achieved using beacons. Because all the end-devices in a multicast group share the same devAddr (called McAddr [17]), they will also have the same pingOffset which alleviates the overhearing and collision problem with other multicast groups or unicast devices that have the same configuration. Furthermore, a node can have multiple multicast addresses and security keys with the same configuration to allow it to participate in more than one multicast group; a node can check each received packet against the stored multicast address with corresponding security keys to identify the targeted multicast group.

[11]The multicast nodes should use the same SF(Spreading factor), channel, bandwidth, and ping-slot periodicity for reception.

[12]The nodes should have the same devAddr called multicast address (McAddr) [17]).

[13]See [17] to see how the security keys are derived.

As LoRaWAN is limited by regional duty cycle restrictions multicasting allows gateways to utilize the duty cycle efficiently when identical transmission is targeted to multiple nodes. This also helps in decreasing the average waiting time for nodes that would otherwise have to be served separately with unicast. However, compared to LoRaWAN Class B unicast, LoRaWAN Class B multicast introduces some restrictions. To avoid the resulting collision, the LoRaWAN specification [10] restricts the use of MAC commands and acknowledgments which require all the receiving nodes in a multicast group to respond at the same time. Furthermore, [10] does not offer support for the creation of multicast groups and securely distributing keys and addresses. This leaves the challenge of setting up a multicast group, starting a multicast session as well as distribution of multicast packets to member end-devices' application layer.

In line with this, the LoRa Alliance released two application layer specifications [17] and [18] for over the air creation of multicast groups and distribution of multicast packets respectively. [17] specifies application layer control messages which are sent in unicast for requesting information about the configured multicast groups in the end-devices, for distributing keys and address required for the end-devices to join a desired multicast group and for informing the end-devices of an upcoming multicast session along with the length of the session. [18] on the other hand specifies application layer commands that are sent in unicast mode to setup the end-devices for an impending multicast session as well as commands that could be sent in multicast mode for checking the status of a multicast session during transmission and for transmitting the multicast fragments. To minimize the resulting collision between the end-devices' responses for the multicast messages that require response – such as checking the status of an ongoing multicast session – [18] defines a random delay that each device should wait before responding to a multicast request. This random delay goes to a maximum delay defined by the session when setting up the end-devices called **BlockAckDelay**. BlockAckDelay is selected in such a way that it accounts for the number of end-devices that are in the multicast group as well as their geographic distribution.

Combining the MAC layer specification [10] and the application layer specifications [17] [18] for LoRaWAN Class B multicast, Fig. 2.11 shows an example on the process of creating a multicast group, setting up a multicast session, distributing the multicast fragments[14] and checking the status of the session.

[14]A multicast fragment packet is a packet among other packets that make up the whole data that is desired to be transmitted to an end device. During the setting of a multicast session the number of fragments that will be transferred in the session will be communicated along with expected time to be spent in delivering all the packets.

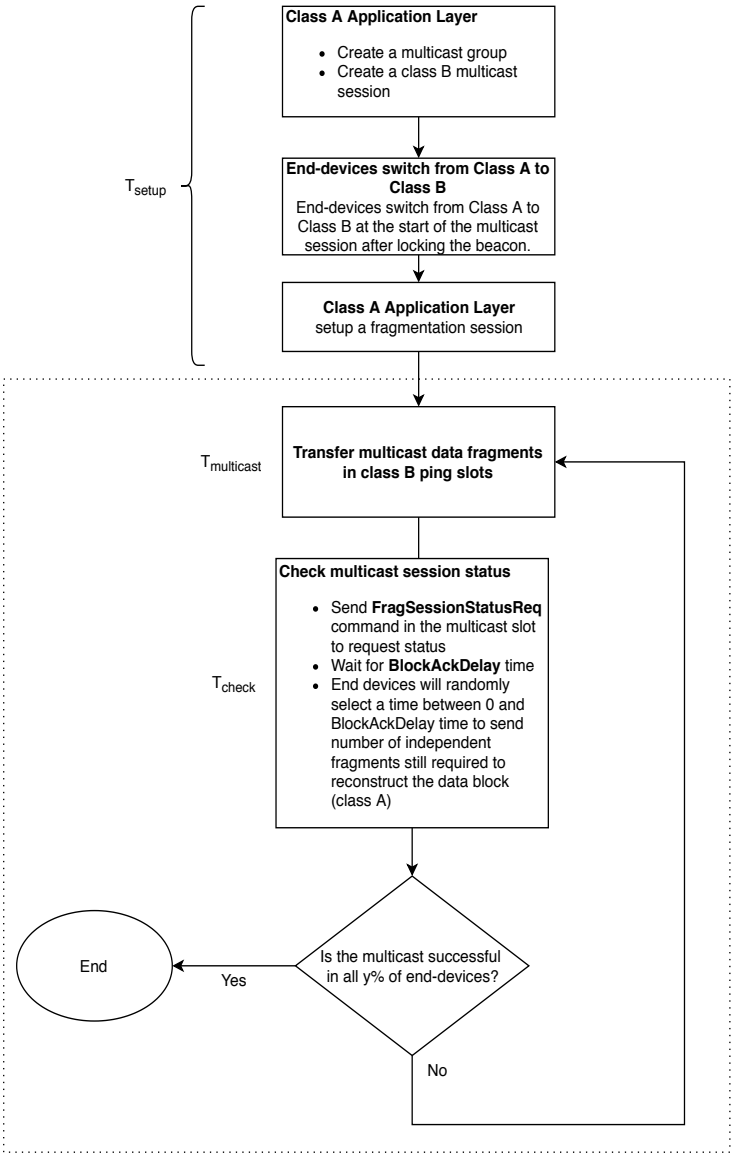


Figure 2.11: A flow chart for transferring data in multiple fragment packets to N end-devices via Class B multicast, where y% of the N end-devices should succeed for the multicast session to end.

2.6 Related work on LoRaWAN Class B

To the best of the author's knowledge there are only few works on LoRaWAN Class B unicast and *this thesis is the first work on LoRaWAN Class B multicast* – or on LoRaWAN multicast in general.

Delobel et al. [19] analyzed the effect of number of channels, number of

end-devices and data-rate[15] on the delay of confirmed class B downlinks, by using a markov chain. The results show that both data-rate and number of end-devices significantly affect the delay. Increasing both the number of sub-bands and number of channels per sub-band significantly reduced the delay when the number of competing nodes increased. On the other hand, for lower data-rates which have an increased delay, only increasing the number of sub-bands contributed to decreasing this delay.

Delobel et al. [19] also identified 3 limitations in the LoRaWAN class B standard: it is not possible for gateways to use all the ping slots of end-devices due to the transmission duty cycle limits imposed by ETSI, the specification does not restrict Class A transmissions during the ping-slots and beacon times (beacon reserved and beacon guard times), and the specification fails to specify a delay between confirmed downlinks and their corresponding acknowledgements. *The work, however, lacks a detailed explanation about the problem of Class A transmission collision with the ping-slots. Furthermore, an assumption was made in the work to postpone the uplink transmission whenever there is a conflict between uplink packet and class B ping-slots. In this thesis we will explore the Class A uplink conflict with Class B under the title of Class B coexistence with Class A and show how the ping-slot contributes to the level of the conflict. We will also discuss why restricting the class A transmissions during ping slots is not a plausible solution and in the worst cases could lead to starvation, where the end-devices can not send any uplink. Other alternatives, along with their pros and cons, that could rather be used to deal with the conflict between class A and class B ping-slots will be discussed.*

Finnegan et al. [20] did a simulation study on the scalability of LoRaWAN Class B unicast end-devices both in the g1 and g3 band with the help of ns-3 simulator. According to the results from the simulation, most of the performance losses in the g1 band were due to the limited transmission duty cycle that the class B gateway has. As the number of end-devices increases, although you have limited transmission queue size and duty-cycle on the gateway, the packets that are queued for the individual class B end-devices will increase – leading to packet loss. On the other side, on the g3 band, the results show that the reduced packet delivery ratio as the number of end-devices increased were due to the blocking of the beacon transmissions when the sub-band is already used for ping downlinks. *However, for drawing conclusions on packet delivery ratio on the g3 band, the work assumed that class B downlink packets are not sent if the preceding beacons are blocked. Furthermore, the effect of the ping slot periodicity on scalability and beacon*

[15]In LoRaWAN data-rate is defined by the combination of spreading-factor (SF) and bandwidth. Larger data-rates have higher physical layer bit rate, while smaller data-rates have lower physical layer bit rate. In the EU863-870 band, for 125 KHz bandwidth, the data-rate goes from DR0 to DR5 which corresponds to a spreading-factor that goes from SF12 down to SF7.

blocking is not considered in the study. This thesis, although it does not directly deal with class B unicast scalability, will address class B multlicast group scalability whose behaviour is similar with that of unicast – only that the performance losses will be shared by all the end-devices in a multicast group – while considering the beaconless operation mode (which can go up to 2 hours and allows for delivery of ping packets without beacons). *We will closely investigate the effect of ping-slot periodicity and data rate as well as the number of multicast groups on the overall beacon delivery performance. We will also evaluate overhearing and collision rate as well as average throughput, as multicast groups (end-devices in case of unicast) scale, which is missing in the current research.*

Xueying Yang et. al. [21] studied the security vulnerabilities in LoRaWAN. LoRaWAN Class B has been explored as part of the study. This thesis, however, will not look into the security of LoRaWAN Class B unicast or multicast.

Chapter 3

Simulator for LoRaWAN Class B multicast

3.1 Existing simulators

ns-3 is a discrete-event network simulator written in C++ with an option for Python binding [22]. It allows to write simulation scripts both in C++ and Python. There have been various researchers who wrote a LoRaWAN library for ns-3. Floris Van den Abeele et al. [23], Brecht Reynders et al. [24] and Davide Magrin et al. [25] have developed a LoRaWAN Class A module.

Joseph Finnegan et al. [20] extended the LoRaWAN Class A module developed by Floris Van den Abeele et al. [23] to include Class B in order to study the scalability of LoRaWAN Class B. Nevertheless, *features like LoRaWAN Class B multicast and the beaconless operation mode are missing in the module.*

3.2 Integrating LoRaWAN class B to ns-3

In this thesis, although we started with [23] and [20], we decided to shift to [25]. The frequent bugs that we faced and the complexity of the module were some of the reasons why we did this shift.

Following the shift, we extended the work of [25], in order to realize an ns-3 module that supports LoRaWAN Class B multicast. This makes the extended module also the first in simulating LoRaWAN Class B multicast. Furthermore, the inclusion of beacon-less operation mode allows the end-devices to behave in a way that is defined by the LoRaWAN specification when they fail to receive beacons occasionally.

3.2.1 Software Architecture

Our software architecture adapts the architecture used by [25] and applies the necessary modifications and additions to realize a module that supports LoRaWAN Class B multicast. Figure 3.1 shows the overall architecture of the module. Parts of module which are modified and extended are colored in red. As both LoRaWAN Class A and Class B essentially share the physical layer and only differ in the MAC layer operation, the physical layer and the channel part of the module are not modified.

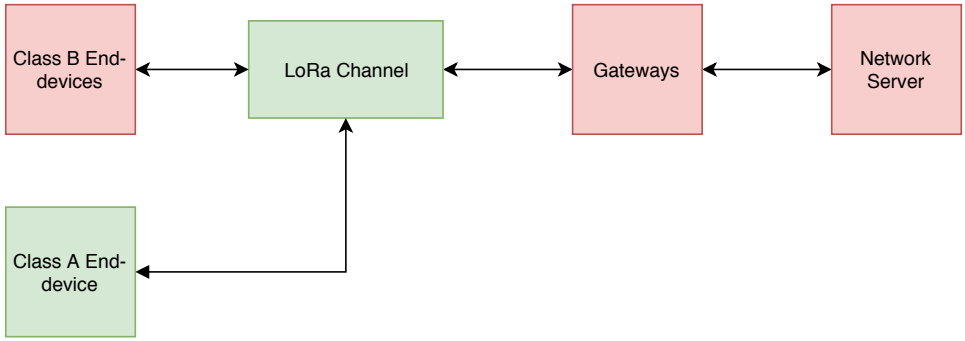


Figure 3.1: The overall architecture for the LoRaWAN Class B module.

In the following section we will delineate basic modifications and additions in the overall architecture. Around 10,877 lines of code were added to the original module. The actual code can be found in the authors github repository[1] and could be consulted for further understanding and replication of results.

3.2.2 End-device Class B application

The end-device Class B application is installed on each end-device in order to drive the MAC. It is responsible for initiating a switch to Class B and also sending Class A uplinks when required. It receives the information about the MAC status as well as the Class B downlinks through callbacks. It is also informed through callbacks when an end-device LoRaWAN MAC fails to lock a beacon or loses it after locking one. In such cases, it is required to re-initiate a switch to Class B until the configured number of trials set is reached.

3.2.3 End-device LoRaWAN MAC

Figure 3.2 show the expanded architecture of the end-device. In the architecture, the end-device LoRaWAN MAC is where both the Class A and

[1] `https://github.com/yoniwt/lorawan-private`

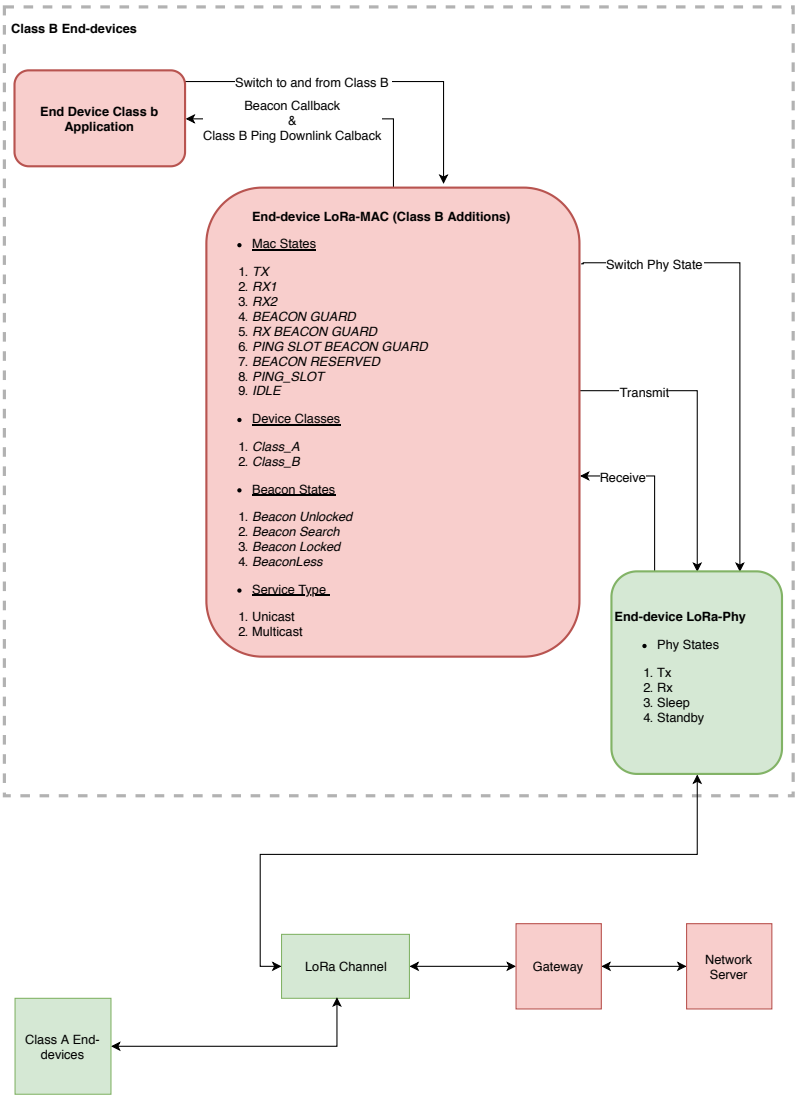


Figure 3.2: Expounded view on the end-device architecture.

Class B logic are implemented. Although Class A was already available on the module that we adapted, we still had to write the Class B part as well as the necessary transitions between Class A and Class B. What is represented in the figure is only the additions and the modifications.

As an end-device can not operate in Class B without a beacon, the beacon search process is the first thing that happens as soon as the Application layer requests for a switch to Class B. The transition of the beacon state on each beacon time, starting from the first request to switch to Class B is given in Figure 3.3. As shown in the figure, the end-device LoRaWAN MAC will

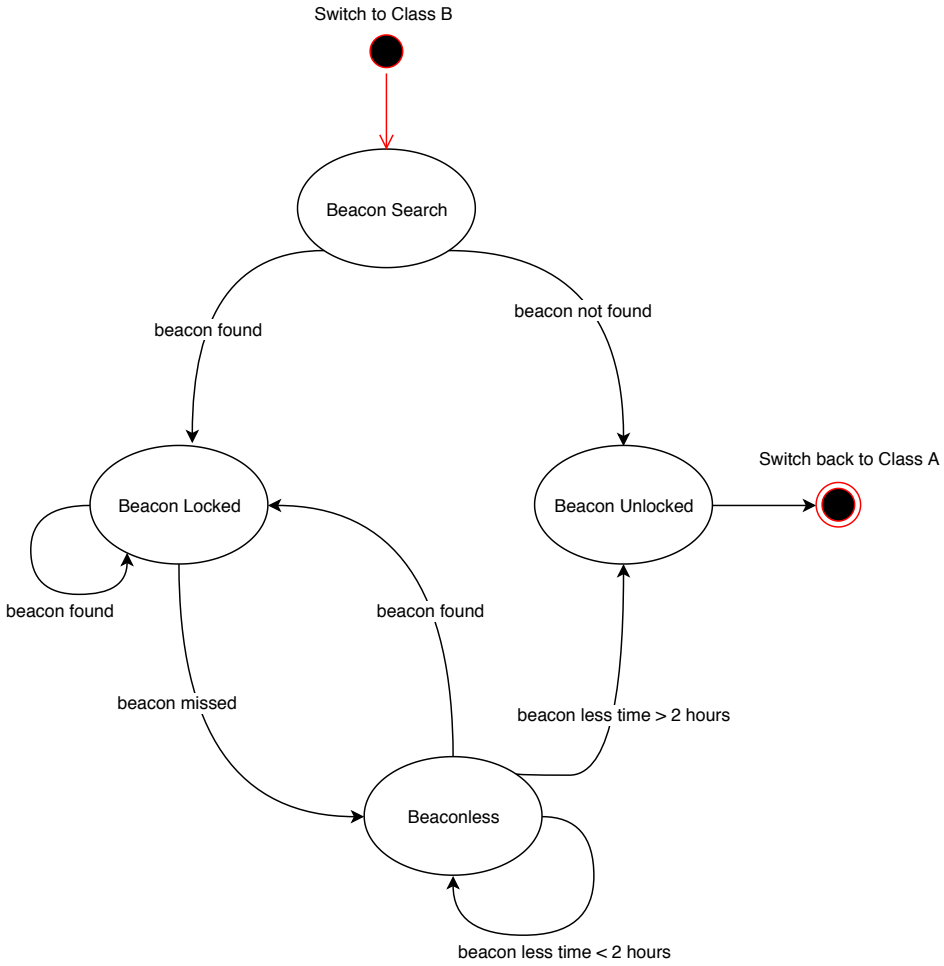


Figure 3.3: The beacon state diagram.

switch back to Class A if a beacon is not recieved for more than the minimal beaconless operation time (2 hours).

While the end-device is in the Class B state, it has to periodically open ping receive windows as well as the receive windows for beacons. And the MAC layer is responsible for opening these windows, as well as deal with other activities such as transmissions and receptions (both Class A Rx1/Rx2 receptions and ping receptions). The MAC states listed in Figure 3.2 are the states which the MAC layer switches to depending on these activities and are described as follows:

- TX: the end-device is currently transmitting.
- Rx1: the end-device has opened the Rx1 window.

- Rx2: the end-device has opened the Rx2 window.
- Beacon guard: the end-device is currently in the beacon guard period.
- Rx beacon guard: a reception that is already started near the beacon guard, has to finish in the beacon guard. This state makes sure that the end-device will switch to beacon guard at the end of the reception.
- Ping-slot beacon guard: is a state indicating that the ping slot is opened for 8 symbols. This state makes sure that the end-device will switch to beacon guard at the end of the slot.
- Beacon reserved: this state indicates the end-device is in the beacon reserved state. Depending on whether a beacon is received or not in this state, the beacon state makes the corresponding transition as shown in Figure 3.3.
- Ping slot: here the end-device opens a ping-slot that is 8 symbols long, in order to detect a preamble. If a ping downlink is detected, it will switch to Rx mode and start decoding, otherwise it will close the ping and go to idle state until the next ping slot time.
- Idle: is a state where there is no activity.

3.2.4 Gateway LoRaWAN MAC

The gateway is the radio interface between the network-server and the end-devices. It forwards Class A uplink packets to the network-server via forwarder and transmits back Class A downlinks, reply packets or an acknowledgements to the end-device. The expanded view of the gateway model is given in Figure 3.4. As we can see from the figure, the gateway LoRaMAC – colored in red – was the only part of the gateway that required a modification to realize a gateway that supports LoRaWAN Class B beacon transmission.

According to the specification, the LoRaWAN beacon preamble should start with 10 unmodulated symbols [10]. This is different from all other LoRaWAN transmissions which require 8 unmodulated symbols by default. Therefore, the necessary change was applied, on the gateway and the communication between the gateway and the network-server, to be able to support beacon transmission.

3.2.5 Network-Server

The LoRaWAN logic of the LoRaWAN Network is implemented on the network-server. The gateway just provides a radio interface to reach the end-devices in its vicinity. As we can see from Figure 3.4, in this model, the

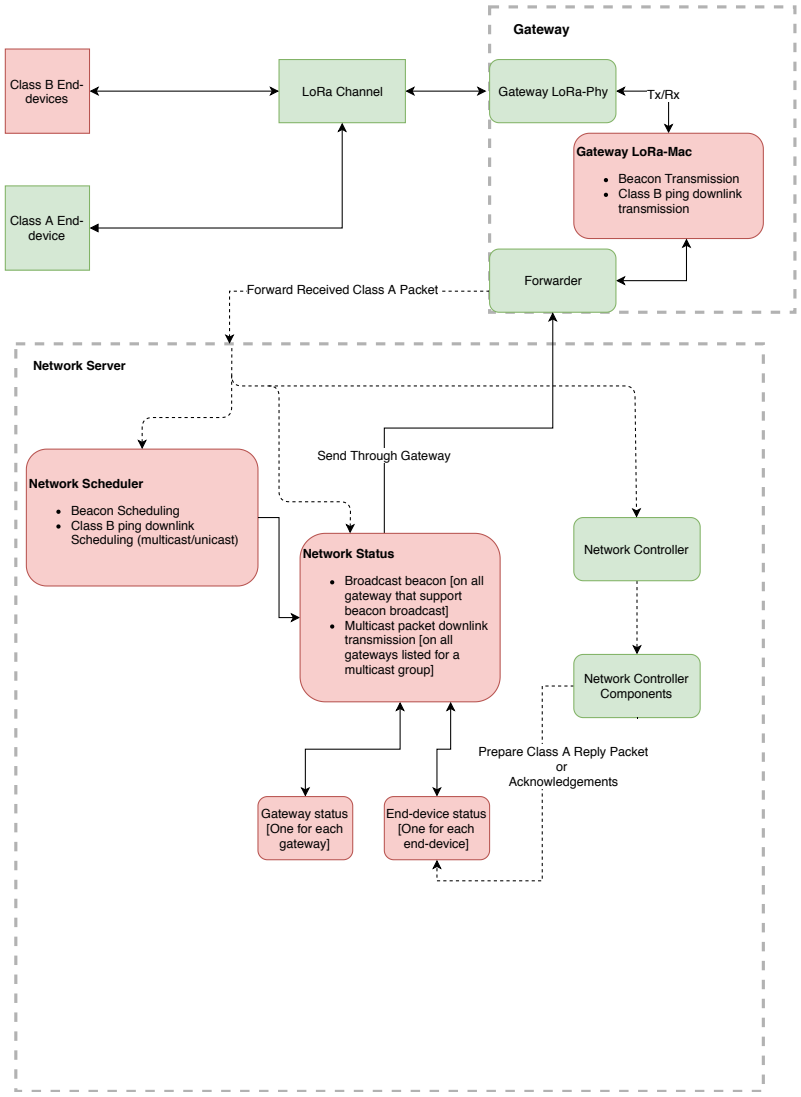


Figure 3.4: Expounded architecture view of the gateway and the network-server.

network-server accomplishes this by using other parts – Network-Scheduler, Network-Status and Network-Controller – which work together in order to realize the set of procedures required both for Class A and Class B operation. The required additions were made on the Network-Scheduler and the Network-Status to accommodate Class B unicast and multicast features.

The Network-Scheduler which is used to schedule Class A Rx1 and Rx2 window downlinks is now equipped with a functionality to schedule beacon broadcasts and Class B multicast and unicast ping donwlinks. Once con-

figured for LoRaWAN Class B, the Network-Scheduler will record the start of the simulation time as a GPS epoch and schedule the beacon broadcasts accordingly. Following each beacon broadcast, the Network-Scheduler schedules Class B ping downlinks both for unicast and multicast transmission, after extracting the necessary information for all the end-devices through their corresponding end-device-status object. Each scheduled transmission may or may not take place depending on whether the required gateways have enough duty cycle to transmit, which can be inquired via the gateways-status object for each gateway. If there is enough duty cycle for transmission, the Network-Scheduler will send the downlink packets through the Network-Status.

The Network-Status has four main tasks: maintaining an updated information of the end-devices' status and the gateways' status, retrieve the reply packets for Class A uplinks for the Network-Scheduler, get a reply gateway address to reach a target end-device, and finally act as an interface to reach those gateways assigned for transmission. In order to realize LoRaWAN Class B, we added beacon broadcast and Class B multicast functionalities on top of the four tasks it already had. Therefore, the Network Scheduler can simply initiate a broadcast and all the remaining stages to broadcast through the beacon-transmitting gateways will be dealt by the Network-Status. Furthermore, the only thing that the Network-Scheduler should do to multicast to a group is just pass the packet along with the group address to the Network-Status, which will complete the rest of the process required to send through the assigned gateways.

To identify bottlenecks, the network server is currently configured to send downlinks at the maximum throughput possible. That means the Network Scheduler will use all the pings slots possible after scheduling the slots for all the multicast groups.

3.3 LoRaWAN Class B multicast performance Analyzer

Required performance metrics and activities in the end-device mac-layer, application layer, and network-scheduler are exposed to the analyzer using traces for further analysis. The performance-analyzer at this point only analyzes class B multicast performance. Simulations of Class B unicast[2] is possible by creating a multicast group with one end-device or by modifying the analyzer for unicast end-devices. As the performance-analyzer already provides access to most of the internal behaviours, additional metric calculations could be done by extending it.

[2]Because LoRaWAN Class B multicast is restricted from the use of MAC commands and confimred Class B downlinks, only those simulations that do not require an exchange of MAC commands and transmission of confirmed Class B donwlinks can be studied.

3.4 Limitations of the module

The extended Class B module has the following limitations:

- Although both Class B unicast and multicast are fully implemented in the end-device, there are few additional steps required to fully realize Class B unicast communication from the Network-Server side. However, Class B unicast end-devices can still be simulated by using a Class B multicast group with a single member. This is enough for generalizing our study on beacon blocking, overhearing and collisions as well as throughput for Class B unicast end-devices which involve unconfirmed downlinks.
- A scheduling policy for Class A reply and Class B ping downlinks is still missing in the Network-server. Hence, confirmed uplinks are not supported together with the Class B end-devices in the Netwok-Server. This will not affect our result as confirmed uplinks are outside the scope of this work.
- The conflict resolution between Class A uplinks and Class B ping-slots as well as between Class A uplinks and beacon-reserved period is not yet fully implemented. This conflict is analyzed in Section 5.2 and conflict resolution techniques are also explored.
- Initial MAC exchanges in Class A to get the GPS time of the gateway is not modeled in the current model. We assume that such exchange has already has happened and register the GPS time from the simulator time. If desired, the exchange can be modeled by using a Class A network, as the exchanges are done before switching to Class B. This, however, is outside the scope of this thesis.
- The clock jitters are not yet modeled. In this thesis, we assume that the end-devices have sufficient PPM[3] that allows them to continue to operate in the beacon-less operation mode by widening their reception windows. The effect of end-devices' PPM on PRR (Packet Reception Rate) is outside the scope of this thesis.

3.5 Simulation script

Several helper classes are extended in order to allow users to create a class B network easily.

- The helpers provide APIs (Application Programming Interface) to activate a beacon transmission and class B downlink transmission on a group of gateways.

[3]PPM which stands for Parts Per Million is used to measure end-devices' clock accuracy.

- The helpers can be used to create a multicast groups by passing the number of end-devices and the number of multicast groups desired.
- The helpers also support manual creation of multicast groups, if desired.
- For the creation of the multicast groups, the helpers will additionally require the user to put the Class B parameters to be used.

These helpers along with the ns-3 helpers and the helpers that already exist for Class A, can be used to create a desired network and analyze the performance. Finally the performance-analyzer can used to to calculate desired performance metrics in the simulation.

For this thesis most simulations are run for 2 hours, unless otherwise stated. This corresponds to the minimal beaconless operation time and hence will give a good estimate on the performance of the end-devices that has once switched to Class B.

Chapter 4

Multicast group scalability

In this chapter, we will explore the performance of LoRaWAN Class B mutlicast as the number of multicast groups increase. In particular, we will be discussing beacon blocking, overhearing and collision as well as the maximum throughput per group. Since in LoRaWAN a multicast group is a set of LoRaWAN Class B end-devices that share the same downlink configuration and keys, Class B unicast can be considered as a special case of class B multicast with a group each containing one member. This, however, is after abstracting out Class B unicast confirmed downlinks and MAC command exchanges which LoRaWAN Class B multicast does not posses and which we are not going to cover in this study. Therefore, the results in this chapter can be applied to LoRaWAN Class B unicast end-device scalability.

All the simulations for this section are done with a cell radius of 7.5Km where all the nodes are randomly distributed with a log-normal fading channel. For all transmissions, the maximum packet-size allowed for each data-rate is used, unless otherwise stated.

4.1 Beacon Blocking

The default channel for both beacon transmissions and Class B ping downlink transmissions is the 869.525 MHz which is in the g3 sub-band as shown in Table 2.1. Although, this sub-band has higher duty cycle (10%) than the other sub-bands, it still has a time-off duration where it is restricted from transmission. This means that Class B downlink transmissions closer to the beacon reserved, could result in blocking of an impending beacon transmission.

Finnegan et al. [20] has looked into this problem as part of a Class B unicast scalability study in the g3 band and indicated the following behaviour:

- Lower data-rates (DR0-DR2) could lead to beacon blocking while the beacon-guard period is enough to prevent higher data-rates (DR3-DR5) from blocking.

- The beacon blocking increases for DR0-DR2 as the number of end-devices (number of groups in our case) increases. And according to the results, DR0 scales less with number of end-devices compared to DR1 and DR2.
- If a beacon is not sent, Class B ping downlinks will be hindered from being transmitted which will impact the packet delivery ratio (PDR).

The study, however, did not include the effect of both the ping slot periodicity as well as the downlink packet size. Furthermore, the beaconless operation mode, as described in Section 2.4.3, is not considered which would have prevented the beacon blocking from directly affecting the packet delivery ratio (PDR). *In our study, using our simulator, we have found that the relationship between the number of nodes and the beacon blocking ratio for different data rates is not straightforward and that it depends hugely on the ping slot periodicity.*

Beacon blocking happens if two conditions are fulfilled: (1) if the combination of the ping slot periodicity, ping offset, packet size and data-rate result in ping slots whose transmission will follow a time-off duration that includes the beacon reserved time, and (2) if the gateway did not have a previous transmission that blocked it from using these ping slots.

Figure 4.1 shows a heatmap for the number of beacons skipped out of 56 beacons generated, where the maximum packet size is used for all the data-rates for class B ping downlink transmissions. The heatmap shows that there is no clear relationship between the number of groups and beacon-blocking: sometimes it increases with the number of end-devices, sometimes it decreases and some times it remains constant. This is because beacon blocking only depends, as we discussed, on the probability of having a ping reception near the beacon guard. This means increasing the number of end-devices will simply raise the probability of having more ping slots occupied. Nevertheless, at the same time, the probability of having an earlier transmission that will block the gateway from using those beacon-blocking ping-slots also increases. This introduces some randomness in the beacon-blocking behaviour.

Despite the random nature of the beacon blocking, however, the following important observations can be made from Figure 4.1.

1. *It is clear from Figure 4.1 that even lower ping slot periodicity could give rise to a systematic*[1] *beacon blocking.* This ranges to 90% beacon

[1]The term systematic refers to a situation that happens continuously. The term is taken from the specification [10], which is used there for continuous collision between gateways as described in Section 2.4.4.

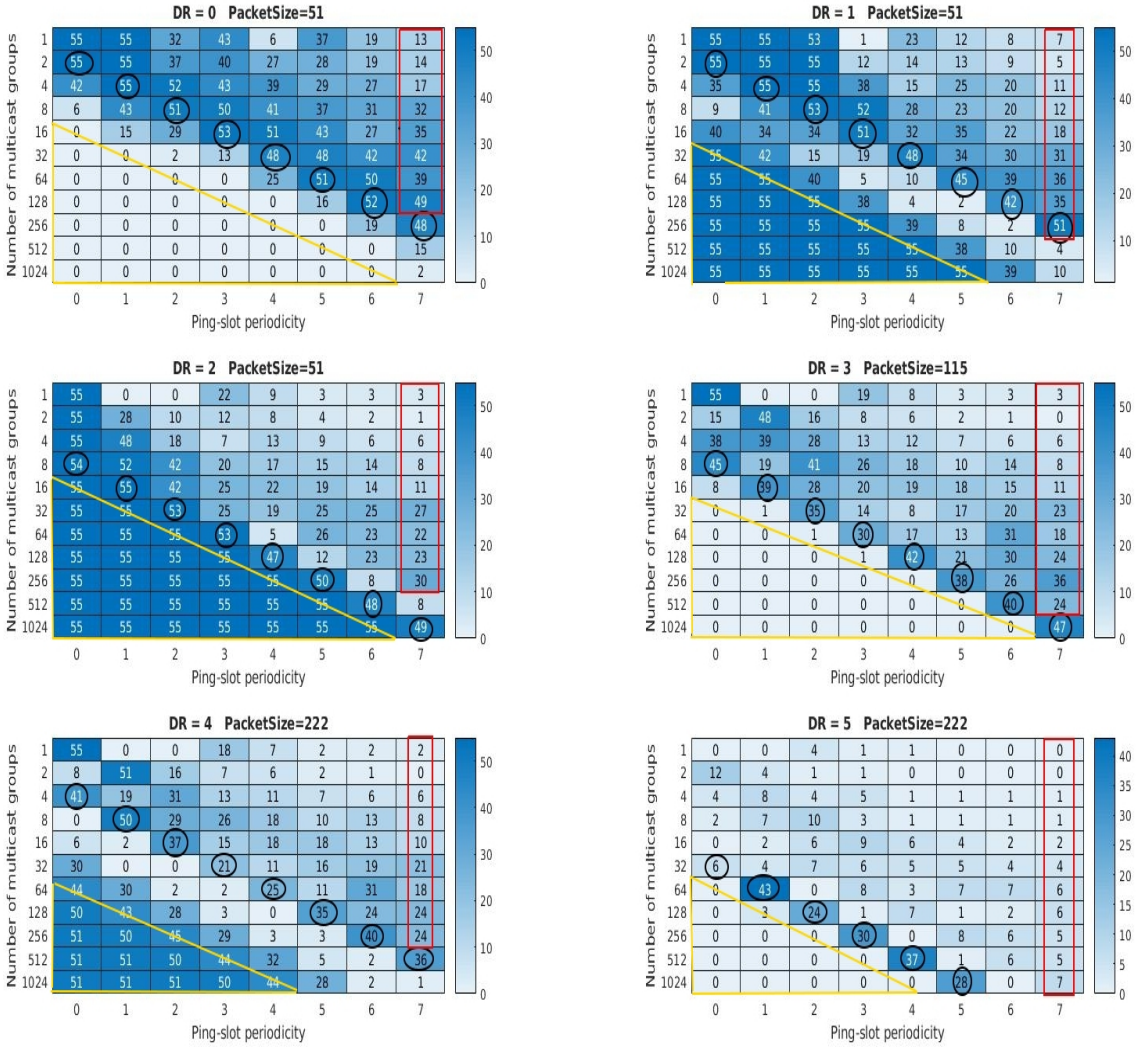


Figure 4.1: Heatmap for the number of beacon blocked out of 56 beacons generated.

blocking for DR0, DR1, DR2, DR3 and DR4. Even for DR5, the fastest data-rate, more than 50% beacons could be blocked for some combination of ping-slot periodicity and number of nodes. Table A.1 lists the number of ping-slots that could result in beacon blocking for different ping-slot periodicity and data-rates. *Therefore, the beacon guard is not enough to protect beacon blocking for any of the data-rates if the maximum packet size is used, hence proper beacon guarding mechanism is indispensable.*

2. *As the number of groups scale, the number of beacons blocked will start to saturate.* At this stage, you have high probability of having at least

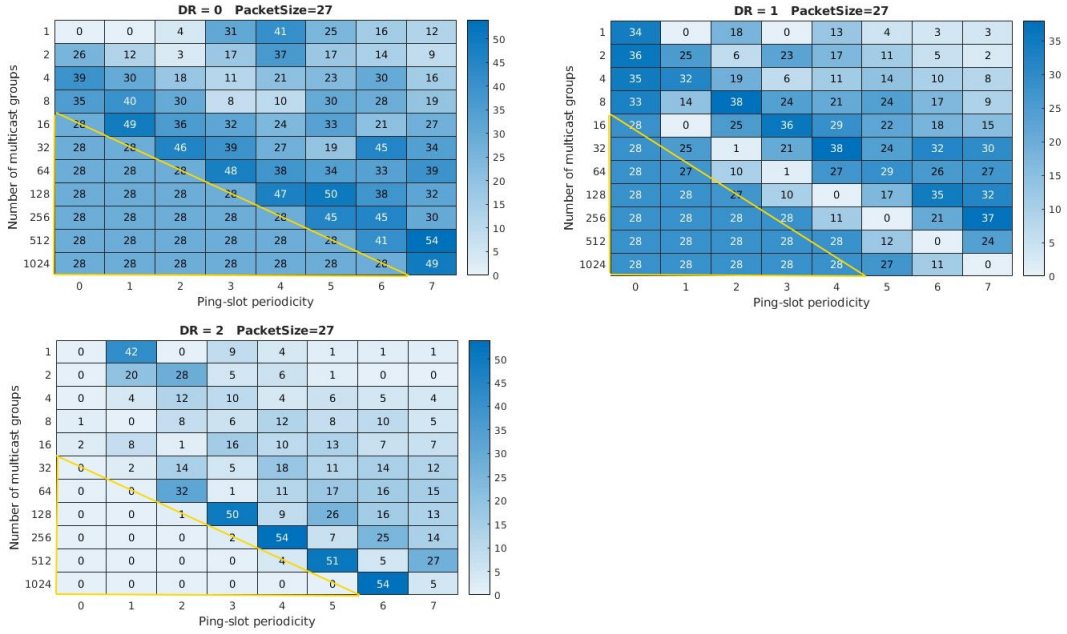


Figure 4.2: Heatmap for the number of beacon blocked out of 56 beacons generated when the downlink packet size is 27 bytes.

one group that opens a ping-slot on each slot, throughout the beacon window. This will result in a pattern in the network-server where it starts sending out ping downlinks always as soon as the time-off duration ends. And this behaviour will be the same no matter how many groups you further add, hence saturation. Because lower ping-slot periodicities have several number of ping-slots per beacon period, saturation begins for lower ping-slot periodicities with few number of groups compared to higher ping-slot periodicities. The yellow triangle in Figure 4.1 inscribes the saturation values for beacon blocking for each data-rate.

3. *As we can see from Figure 4.1, this saturation could be the highest (100%) or the lowest (0%) beacon-blocking*: DR0, DR3 and DR5 show a better condition where smaller to no beacons are blocked, whereas DR1, DR2 and DR4 experience the worst blocking. If we concentrate on DR0, DR1 and DR2, where DR0 have the highest air-time, we would not have expected that DR0 will experience the lowest beacon beacon blocking on saturation. The explanation for this is that DR0 had an air-time long enough that earlier transmissions prevented it from using the later ping-slots that lead to beacon blocking. This behaviour can change if the packet size is decreased. Figure 4.2 shows the heatmap for beacon blocking when the packet size is reduced to

27 bytes. As we can see in the figure, when the packet size is reduced, DR0 starts having an increased beacon blocking at saturation and DR2 experienced a decreased beacon blocking at saturation. *Therefore, the saturation value of beacon blocking depends not only on the data-rate and the ping-slot periodicity, but also on the length of the transmission packets. Hence, irrespective of the ping-slot periodicity, having a beacon guarding technique will avoid the risk of having high beacon-blocking saturation values.*

4. For higher ping-slot periodicites, you can see that the number of beacons blocked increases with the number of groups, before saturation is reached. This pattern is highlighted in red in Figure A.1.

As some of the above observations have indicated, a proper mechanism to guard the beacons is indispensable. Failing to guard the beacon in the midst of systematic beacon blocking will lead to an exponential increase in energy consumption and eventually force the end-devices to quit Class B operation. *In the following sections, we will explore alternatives that can 100% prevent beacons from being blocked, when a single gateway is used both for beacon and Class B ping downlink transmissions.*

4.1.1 Extending the beacon guard in the network-server

In this solution, the beacon guard is extended so that it includes the time-off duration as well. However, as the maximum air-time is around 24.66 seconds, which corresponds to DR0, putting such a guard will degrade the efficiency of the network-sever. In the worst case, if a ping-slot periodicity of 7 is used, the gateway will refrain from using all last 24 ping-slots, which were safe for higher data-rates.

A better alternative is to *use a separate beacon guard for each data-rate so that the maximum air-time for each data-rate will not cause beacon blocking.* Table 4.1 shows the corresponding beacon guard for each data-rate, that can be used to avoid beacon blocking.

Table 4.1: Gateway beacon-guard in addition to the end-device beacon guard in order to avoid beacon blocking

DR	Max air-time (ms)	Time-off (ms)	Beacon-guard (ms)
DR0	2465.792	22192.128	24657.920
DR1	1314.816	11833.344	13148.160
DR2	698.368	6285.312	6983.680
DR3	676.864	6091.776	6768.640
DR4	655.872	5902.848	6558.720
DR5	368.896	3320.064	3688.960

4.1.2 Limiting the maximum payload size for each ping-slot

The solution in Section 4.1.1 blocks all transmission including smaller packet sizes that will not result in beacon blocking during the extended beacon-guard. Table A.1 lists the number of pings that could result in beacon blocking for different data-rate and periodicity combinations. This result in unnecessary queuing of small sized packets that are scheduled to be transmitted in those last ping-slots.

A better utilization of the ping-slots can be accomplished by limiting the maximum payload size for each data-rate, rather than restricting the use of the ping-slots by setting a fixed guard. This is basically modifying the maximum payload size given in the specification [7]. To calculate the maximum packet size which will not lead to beacon blocking, we first start by putting a requirement that a downlink transmission should not be longer than the beacon-guard (3 seconds), together with its time-off duration. Therefore, according to Equation A.5, the denominator ($T_{packet}+9T_{packet}-beaconGuard$) should be less or equal to zero to guarantee that the gateway will be available for the beacon transmission, where T_{packet} stands for the air-time of the packet and *beaconGuard* stands for the beacon-guard duration. This can be solved using Equation A.1, Equation A.2, Equation A.3 and Equation A.4, and which result in Equation 4.1 when combined, where BW stands for band-width. Equation 4.1 can then be solved for maximum PL (payload length) that would not cause beacon blocking. The LoRaWAN application payload can then be calculated by subtracting the LoRaWAN header (13 bytes) from the PL [10].

$$\max(\lceil\frac{8PL-4SF+44}{4SF}\rceil\times 5,0)\leqslant\frac{beaconGuard*BW}{10\times 2^{SF}}-20.5 \qquad (4.1)$$

In Table 4.2, the maximum LoRaWAN payload that will not cause beacon-blocking is calculated using Equation 4.1 and listed along with the corresponding combined air-time and time-off duration. Nevertheless, we can see

Table 4.2: Maximum LoRaWAN packet size that can be used for each data-rate to minimize beacon-blocking and the corresponding combined air-time and time-off duration

Data-rate	**LoRaWAN payload size (bytes)**	**AirTme+Time-off**	Remark
DR5	174	2.972	<beacon guard
DR4	85	2.975	<beacon guard
DR3	31	2.877	<beacon guard
DR2	1	2.888	<beacon guard
DR1	0	5.775	>beacon guard
DR0	0	11.551	>beacon guard

from the table that DR0 and DR1 extend beyond the beacon guard and

cause beacon blocking without any application layer payload, only with the headers. That means it is impossible to get the combined airtime and time-off duration of DR0 and DR1 below the beacon guard-time (3 seconds). For DR2, although it is possible to use 1 byte and prevent beacon blocking, it will be inefficient to put such restrictions to transmissions. *Hence, this technique is not useful for lower data-rates.*

An improved solution is, therefore, to set the maximum packet size to be used for each ping-slot index according to the ping-slot periodicity. The ping-slot to be used by the network-server will then be calculated based on the transmission packet size. This avoids beacon blocking while dynamically restricting the ping-slot usage based on the packet size of the transmission. That means, *rather than limiting the combined air-time and time-off duration not to exceed 3 seconds for all the ping-slots, earlier ping-slots can be used for packets that have longer air-time.* For example, Equation 4.1 can be modified for the ping-slot that is just before the last ping slot as given in Equation 4.2 to accommodate longer packets.

$$\max(\lceil \frac{8PL - 4SF + 44}{4SF} \rceil \times 5, 0) \leqslant \frac{(beaconGuard + pingPeriod) \times BW}{10 \times 2^{SF}} - 20.5 \tag{4.2}$$

We can generalize this, for each ping-slot with a ping slot index N (where N goes from 0 to pingNb-1), as given in Equation 4.3.

$$\max(\lceil \frac{8PL - 4SF + 44}{4SF} \rceil \times 5, 0) \leqslant \frac{(beaconGuard + (pingNb - 1 - N) \times pingPeriod \times 0.030) \times BW}{10 \times 2^{SF}} - 20.5 \tag{4.3}$$

From the equation, we can see that earlier ping-slots in a beacon window with a smaller ping-slot index (N), increase the right side of the inequality, accommodating longer packets. On the other side, ping-slots closer to the beacon guard, hence with greater ping-slot index (N), are restricted to shorter packet size. *This sets a virtual beacon-guard that varies with ping-slot index (N) for each data-rate.* This enhances the utilization of the ping-slots for lower ping-slot periodicities. Figure B.1, in Appendix B, shows the maximum packet size supported for each ping-slot for different data-rates and periodicities.

4.1.3 Limiting the maximum payload size for each ping-slot and ping-offset

As we can see in Appendix B, although the technique in Section 4.1.2 allows – depending on the transmission packet size – a better utilization of ping-slots, its performance will start to degrade as the ping-slot periodicity increases. This is because the maximum packet size for each ping-slot is calculated for

the highest ping-offset. In the worst case, that is ping-slot periodicity 7, it will fallback again to the maximum payload size given in Table 4.2. This will block the network-server from sending any packet to a multicast group that is using DR0 or DR1, as well limit DR2 to maixmum allowed packet size of 1 byte. Furthermore, the other data-rates are also limited to a small packet size, while they could have carried much longer packet size if the ping-offset is considered.

The final modification of this beacon blocking algorithm takes into consideration the ping-offset in which a packet will be transmitted. *This allows any transmission at any slot, to be transmitted as long as it will not lead to beacon blocking.* The modified equation that takes into account the ping-offset is given in Equation 4.4. The network-server solves this inequality before every Class B ping transmission to make sure that the transmission will not lead to beacon blocking. If such a transmission will lead to beacon blocking, the network-server will postpone it to a ping-slot after beacon transmission.

$$\max(\lceil\frac{8PL-4SF+44}{4SF}\rceil\times 5,0)\leqslant((beaconGuard+((pingNb-1-N)\times pingPeriod\times 0.030)+((pingPeriod-1-pingOffset)\times 0.030))\times BW)\times\frac{1}{10\times 2^{SF}}-20.5 \tag{4.4}$$

This resembles a packet-scheduler that makes sure that no beacon is blocked and all packets that do not result in beacon blocking are transmitted. The disadvantage of this algorithm is that the network-server has to compute this before every transmission and hence adds computational overhead to every transmission. To cop with this, the values can be pre-computed and stored in a lookup table, so that the network-server can simply resolve the ping-slot periodicity, ping-offset and data-rate into the maximum packet size allowed in a particular slot. Figure 4.3 shows the maximum packet size which can be transmitted at each slot in a beacon window without blocking beacons.

4.2 Overhearing and collision

LoRaWAN uses slot randomization in order to alleviate overhearing and systematic collision among end-devices that have similar class B configuration including ping-slot periodicity as we have seen in section 2.4.4. Nevertheless, as the number of multicast groups increases the slot randomization technique will fail to accommodate all the groups without an overlap. Since the ping-slot offsets are distributed over a ping period, the ping slot periodicity will therefore have an impact on group (or end-device in case of unicast) scalability. Figure 4.4 shows the ratio of average overheared packets per group over

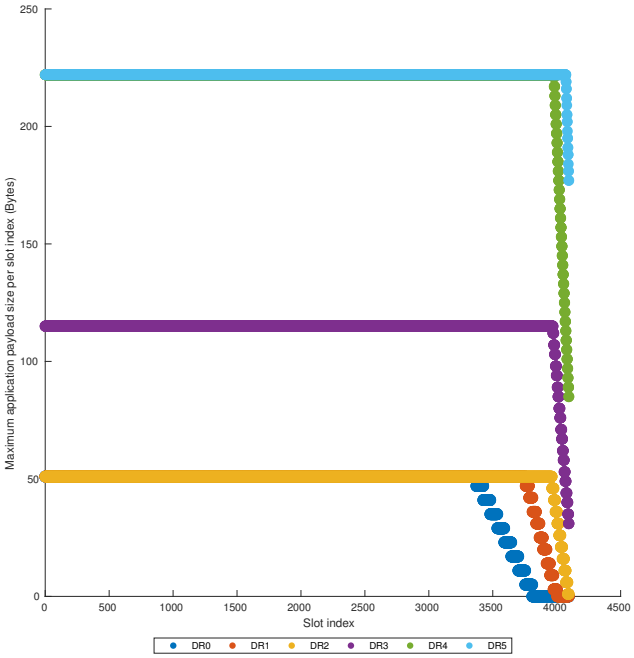


Figure 4.3: Maximum packet size for each slot in a beacon window.

received packets (overhearing ratio) for each ping slot periodicity. These overheard packets will lead to increased energy consumption for each multicast group in the presence of a single gateway, and could result in packet loss by creating interference in case of multiple gateways. Although capture effect might help for end-devices closer to the gateway that is serving them, the interference could be significant enough for the others. This can have a huge impact on multicast group scalability.

As we might expect, for all data rates, we can see from the figure that *lower ping-slot periodicities are less prone to collision and overhearing compared to higher ping-slot periodicities, and hence scale better*. This is because higher ping-slot periodicities have larger ping-periods that can accommodate more multicast groups (unicast end-devices) with less overlap. On the other hand, for lower periodicities, the number of ping-offset decreases with ping-period, which increases the overlap among multicast groups (unicast end-devices).

The other parameter that impacted the overhearing ratio, as we can observe from Figure 4.4, is data-rate: higher data-rates experienced less overhearing ratio compared to lower data-rates. The main reason for this has to do with the fact that lower data-rates have higher transmission air-times

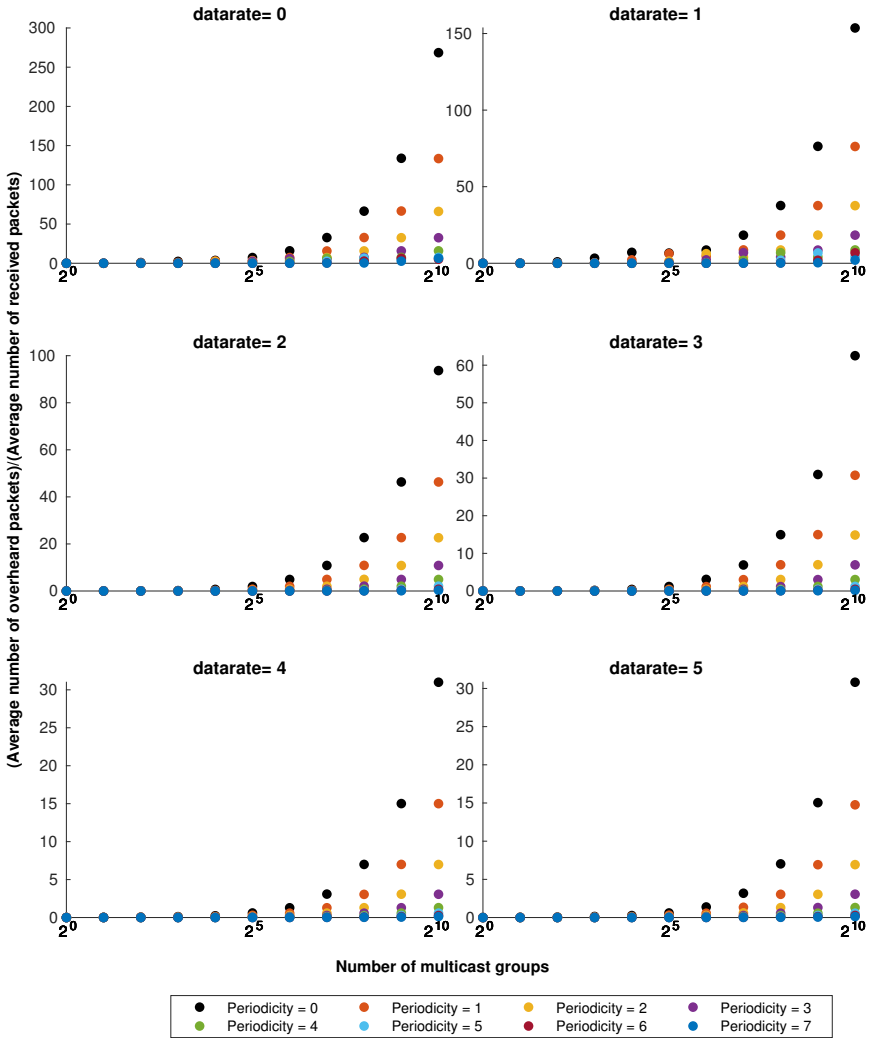


Figure 4.4: Number of overheard packets over number of packets received (overhearing ratio) as the number of groups increases. *Pay attention to the scale of the plots for ease of comparison.*

compared to higher data-rates. That means, for lower data-rates, a single transmission has higher probability of overlapping with multiple ping-slots. This causes a single transmission to be overheard by more groups than what the higher data-rates could have caused. On top of this, since LoRaWAN transmissions are limited by duty cycle, longer air-times of the lower data-rates will be followed by longer time-off duration, which decreases the frequency of packet reception. *Together, the decrease in the number of packets*

received and the increase in overheard packets, raises the overhearing ratio for the lower data-rates.

Therefore, an increased number of multicast groups or unicast end-devices has to be compensated by smartly choosing a good combination of data-rate and ping-slot periodicity. If a cell is experiencing bad channel condition for most multicast groups or unicast end-devices, selecting a lower data-rate has to be compensated by decreasing the ping-slot periodicity.

4.3 Throughput and maximum application server transfer rate

Finnegan et al. [20] showed that as the number of Class B end-devices increase there will be a drop in PRR (packet reception ratio), because of queuing of packets. In real world scenario, however, the network operator has control over the maximum frequency by which application-servers send downlink packets. Therefore, such queuing of packets can be avoided by setting the maximum throughput by which application-servers send downlink packets.

For our study, we preferred throughput over PRR for scalability study, which then can be used to set the maximum throughput for application servers. With the help of the simulator we extended, we analyzed the average throughput experienced by multicast groups (or end-devices in-case of unicast) as the number of groups increase. The results of the simulation are shown in Figure 4.5 and the following observations can be made from the figure with respect to group scalability (or end-device scalability in-case of unicast).

- *For all ping-slot periodicities and data-rates increasing the number of groups, on average, decreases the throughput experienced by each group.* This is an expected result as a single gateway can only serve one group at a time and each added group will share the throughput of that same gateway. This is further worsened by the fact that the gateway is restricted to the 10% duty cycle restriction of g3 band. Although using multiple gateways will help, the improvement is again limited by the collision and overhearing effect discussed in Section 4.2.

- *Higher data-rates scale better than lower data-rates.* This is mainly because higher data-rates can accommodate longer packet sizes and are followed by shorter time-off duration compared to the lower data-rates.

- Although DR5 has the shortest air-time and time-off duration, we can see DR4 had a better throughput with smaller number of groups – especially, for ping-slot periodicity of 3 and above. This is because,

for higher ping-slot periodicity with smaller number of groups, *the advantage that DR5 offers over DR4 is insignificant compared to the link-budget it trades-off. Hence, the packet loss due to the reduced link-budget hampers the overall throughput.*

- *Ping-slot periodicity has a significant impact on throughput.* Because the higher ping-slot periodicities offer smaller number of ping-slots per beacon window for the network-server to send downlinks, their throughput is much lower than the lower ping-slot periodicities which open ping-slots more frequently. And as we can see from the figure, *the difference between the higher and the lowest ping slot periodicity is greater than 200 Bps for higher data-rates.*

- Lastly, we can see that the lower data-rates (DR0, DR1 and DR2) show relatively very small improvement in throughput with a decrease in ping-slot periodicity when compared with the higher data-rates. This is accounted to the longer air-time that *low data-rates* have compared to higher data-rates, and hence *can not use all the available ping-slots although the number of ping-slots is increased.* That is the reason why you see the gap between the higher data-rate and lower data-rate widened as the ping-slot periodicity increases.

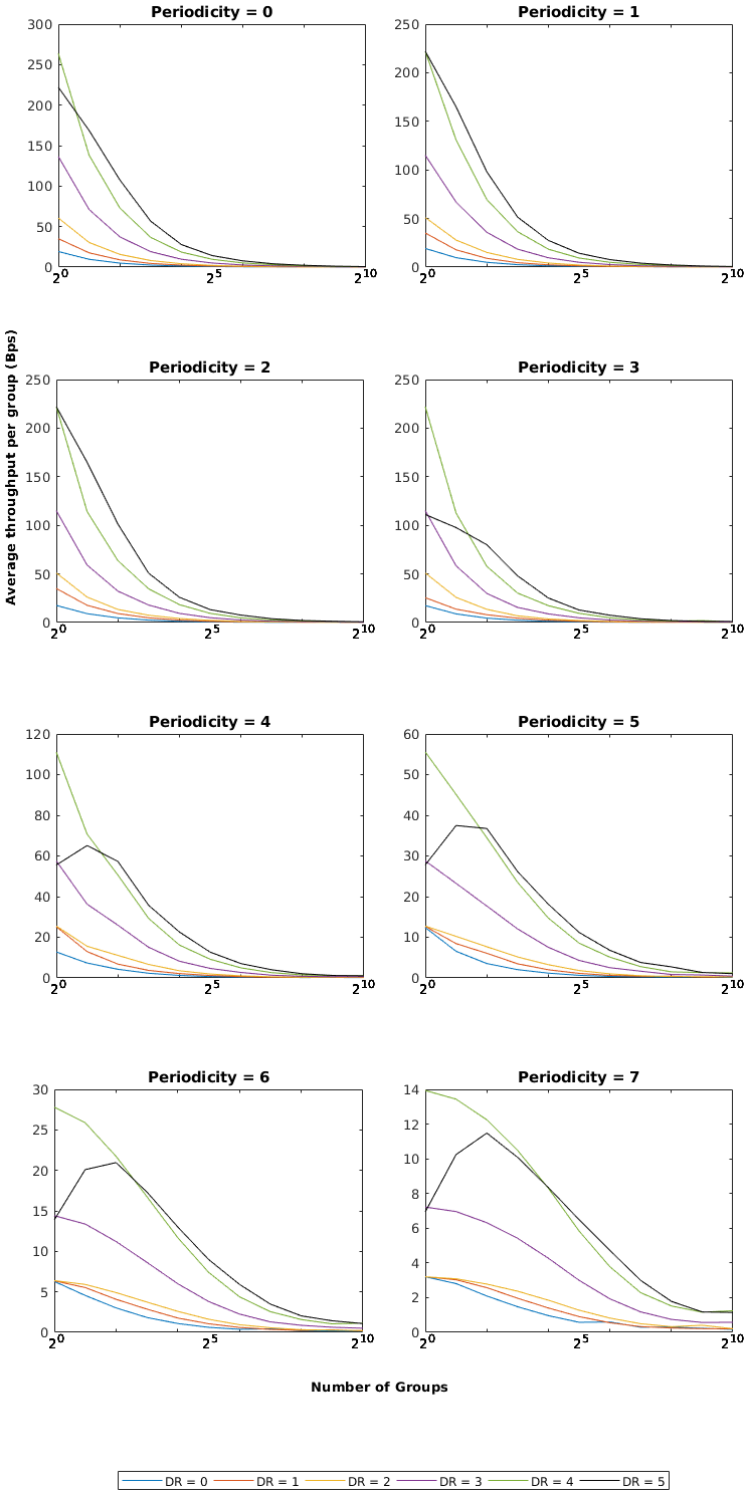


Figure 4.5: Average throughput per group as the number of multicast groups increases. *Pay attention to the scale of the plots for ease of comparison.*

Chapter 5

Multicast member scalability

In this chapter, we turn our attention towards multicast member scalability, focusing only on a single multicast group. Together with the study we conducted on group scalability in Section 4, it will give a holistic view of LoRaWAN Class B multicast scalability as a whole. We will, first, focus on multicast capacity and how we can use the unused ping-slots to improve it. Then we will address, the Class A coexistence with Class B issue on the end-device side and discuss different conflict resolution alternatives. We will end this chapter, by discussing multicell multicast scaling challenge.

5.1 Multicast capacity

In an ideal channel without fading, multicast can enable a support for infinite number of end-devices because a single transmission is shared with all the targets (also called multicast group members). Nevertheless, because a real channel experiences fading, a transmission will not always reach all its destinations. This sets a limit to how many end-devices you can have within a multicast group, because such losses have to be detected and recovered if they exceed the threshold set by their corresponding application.

In Figure 2.11, we have shown an example flowchart for a LoRaWAN multicast session, where a data is transferred in multiple packets. In the flow chart, we see that after a transmission, the network-server will send an application layer command to the member end-devices to report their status within a BlockAckDelay time. As the number of end-devices increases, especially with lower PRR, the number of end-devices that will respond in the BlockAckDelay time will increases, which will either increase the rate of collision between the responses or require the BlockAckDelay time to be extended, prolonging the total multicast session time in both cases. Furthermore, as the number of end-devices with lower PRR increases, the frequency at which the network-server retransmits packets increases, which leads the it to spend most of the time allocated to that multicast group retransmitting.

We therefore define, for this section, the multicast capacity with the PRR distribution of the member end-devices. If the probability of end-devices with higher PRR increases, the multicast can accommodate more members, as the amount of retransmissions required is low. This means the multicast has higher capacity. On the other hand, if the probability of end-devices with lower PRR increases, the number of members that a multicast can support decreases, as such devices will lead to more retransmissions. This means the multicast has lower capacity.

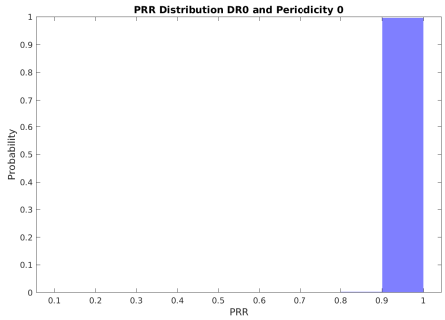
To estimate the multicast channel capacity for LoRaWAN Class B multicast, we used the simulator we extended and run a multicast session for 2 hours with different data-rate and periodicity. The member end-devices were distributed randomly over a 7.5 km cell radius, where the gateway is placed in the center. For modeling the multipath fading we used the m-Nakagami model which is among the famous stastical multipath channel models. It offers a more general distribution than both Reigligh and Recian, and fits well with experimental results [26]. When $m = 1$, Nakagami reduces to Rayleigh distribution and with $m < 1$ it models fading that exists in real scenarios and is worse than Rayleigh distribution [26].

This simulation was done for 10 different seeds and the resulting PRR distribution is averaged for the end-devices that locked a beacon and hence participated in the multicast transmission[1]. Figure 5.1 shows the averaged PRR distribution (from now on will be referred to as PRR distribution) for DR0, DR3 and DR5 with $m = 1$ for the Nakagami model.

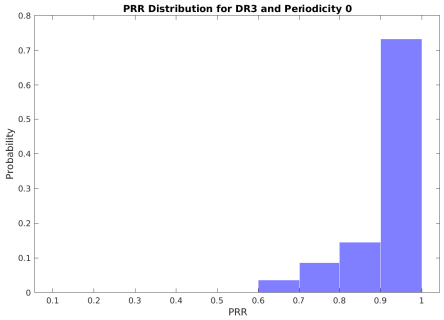
The result shows that the capacity of the multicast channel decreases as we increase in data-rate. DR5 has only 50% member end-devices having a PRR greater than 90%, and DR0 has a high probability that all the member end-devices will receive transmitted packets with a PRR that is greater than 90%. Nevertheless, we should keep in mind that lower data-rates have low throughput compared to high data-rates and hence will take longer time to complete a multicast session. Therefore, for smaller number of end-devices, using higher data-rate and retransmitting the lost packets might give a lower multicast session time. As the member end-devices scale, however, because the number of end-devices that experienced a PRR of less than 90% are more than 50%, the amount of time spent on retransmission will start outweighing its higher throughput. That means, it will start to take more time than the lowest data-rates. Therefore, *lower data-rate became a better choice for multicast transmission as the number of member end-devices scale.*

Now, if we re-plot the PRR distribution for Nakagami $m = 0.5$ as shown in Figure 5.1, we witness that the multicast capacity is significantly reduced

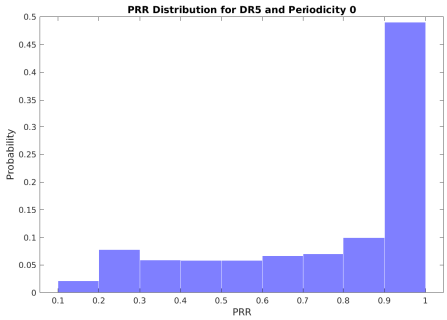
[1]For this study, end-devices that never locked a beacon were not considered in the PRR distribution as they will not open a reception window for the multicast transmissions anyway. From our simulations we found that lower ping-slot periodicites were prone to such issues, rather than higher ping-slot periodicities. This is because of the high probability of beacon-blocking for low ping-slot periodicities as discussed in Section 4.1.



(a) DR0 and periodicity 0.



(b) DR3 and periodicity 0.



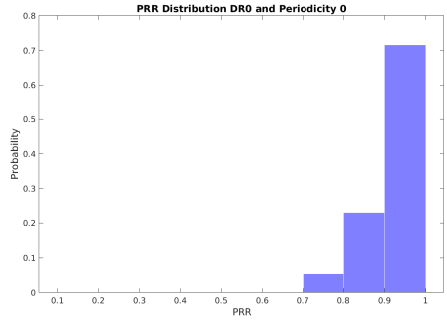
(c) DR5 and periodicity 0.

Figure 5.1: PRR distribution of LoRaWAN Class B multicast for Nakagami $m = 1$ fading model.

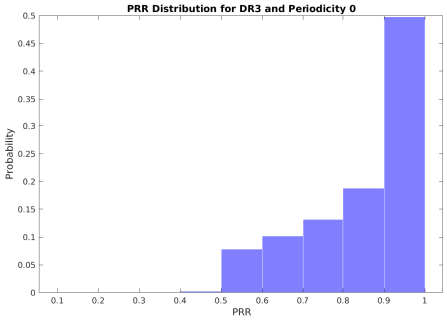
for almost all data-rates. The probability of having a PRR greater than 90% has reduced to 0.3 for DR5 and 0.7 for DR0. As we increase the number of member end-devices, each reduction in PRR probability will translate to a significant number of end-devices, hence hampering the total multicast transmission. For example, if you have 1000 member end-devices, a PRR reduction of 0.1 translates to 100 end-devices. Hence, improvement techniques are highly desired for real world multicast deployments that could have such magnitude of fading.

5.1.1 Improving multicast capacity

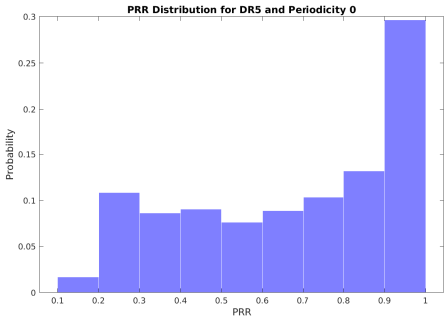
A multicast capacity could be improved in two ways: by relaxing the PRR requirement and by improving the PRR distribution. The first can be accomplished by using erasure coding techniques such as the one provided by the specification [18]. The second can be accomplished by using any technique that will improve the reception of a multicast packet to more member end-devices.



(a) DR0 and periodicity 0



(b) DR3 and periodicity 0



(c) DR5 and periodicity 0

Figure 5.2: PRR distribution of LoRaWAN Class B multicast for Nakagami $m = 0.5$ fading model.

The LoRaWAN Fragmented Data Block specification suggests an erasure coding scheme that is based on parity-check code. An addition of 10% redundancy will be able to accomplish a full recovery with 10% of missing fragments, hence relaxing the PRR requirement for retransmission. The downside of this technique is it increases the size of data-transmission, which translates to downlink air-time and time-off period. Nevertheless, it is still better than retransmitting the whole packet, and other optimal erasure coding schemes could be researched. This, however, is outside the scope of this work.

The second method of improvement, which is improving the reception of the multicast packets in member end-devices, is the main technique of improvement that we used in our study and that we will explore in the next section.

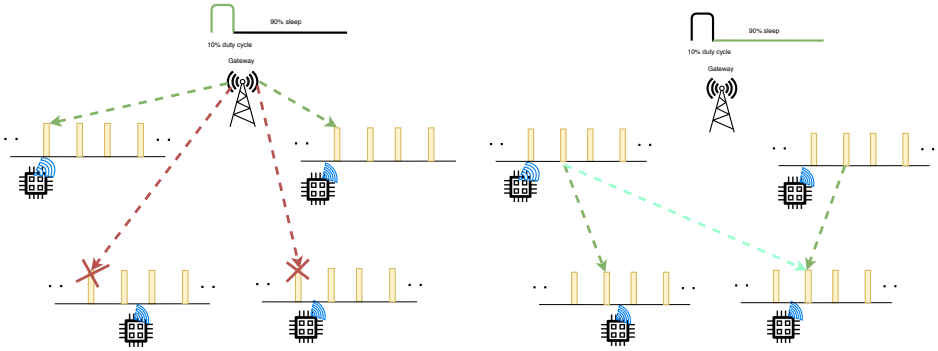
5.1.2 Ping-slot relaying: exploiting unused ping-slots for improving multicast capacity

The main idea behind this technique is trying to assist the duty cycle limited gateways in improving reception of an already transmitted multicast packet. This is done by using the end-devices that already received the packet to help others recover lost packets, during the time-off period of the gateways. Since the end-devices are spread over a space and they transmit the packet following the reception, this technique will exploit both space and time diversity. However, we want to accomplish this in such a way that it is *compatible with the LoRaWAN specification* and *without a need to modify the network-server*, while being *able to benefit member end-devices that do not yet use the technique.*

One approach to meet this goal is to re-purpose the function of the ping-slots that are already aligned throughout the multicast members. After receiving a ping packet in the ping-slots, the packet receiving end-devices can use the next ping-slot to retransmit the packet. However, since more than one nodes transmit the packet during that pings, it leads to massive collision between transmitters. This is where the capture effect of LoRa comes into play. Capture effect is an effect usually seen in frequency modulated signals, where the strongest signal is demodulated successfully from colliding packets. To exploit the capture effect more, the end-devices randomly select the transmission power based on the number of member end-devices there are in the group. The bigger the number of member end-devices, the higher the end-device density and hence less probability is assigned to higher transmission powers. On the other hand, the smaller the number of member end-devices, the lower the end-device density and hence high probability is assigned to the higher transmission powers[2]. Figure 5.3 shows this procedure where two end-devices that successfully received a packet are relaying to compensate for the lost packet in the other two end-devices. Figure 5.4 shows the flowchart for relaying algorithm. As we can see from the flowchart, a hop count is included in the packet, to limit the number of ping-slots that could be used for ping-slot relaying after a multicast packet is received. If left unlimited, it might eventually interfere with the next ping-slot transmissions from the gateway.

As this algorithm uses unused ping-slots after each packet reception, its performance drops and could even be counteractive if the ping-slot relaying end-devices relay during the time the gateway is sending ping downlink. This limits the operation of the algorithm to higher ping-slot periodicities where we know that the gateway will be in time-off period following the transmission; which is in line with our original goal of assisting duty cycle

[2] The lowest transmission power that a device could select for relaying is 0 dB whereas the highest power is 14 dB (the maximum transmission power that a LoRa end-device can accomplish without using a power boost [13]).



(a) Multicast reception failed in two devices and succeeded in two.

(b) The successfully receiving end-devices used the time-off period of the gateway to retransmit the packet they received.

Figure 5.3: Relaying with unused ping-slots.

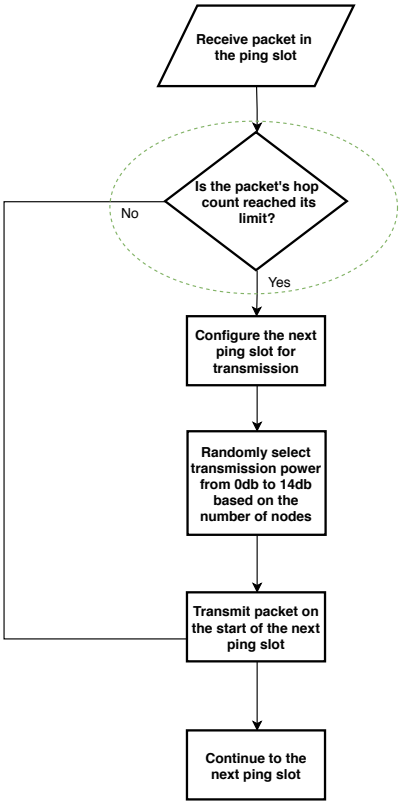


Figure 5.4: Relaying flowchart for multicast member end-device after receiving a multicast packet.

limited gateways, rather than replacing them. Table 5.1 shows the maximum number of ping-slots that fall in the time-off period of the gateway, after transmitting a packet with the maximum packet length that the corresponding data-rates supports.

Data-rate	**Ping-slot periodicity**				
	0	**1**	**2**	**3**	**4**
DR0	23.00	11.00	5.00	2.00	1.00
DR1	12.00	6.00	3.00	1.00	0.00
DR2	6.00	3.00	1.00	0.00	0.00
DR3	6.00	3.00	1.00	0.00	0.00
DR4	6.00	3.00	1.00	0.00	0.00
DR5	3.00	1.00	0.00	0.00	0.00

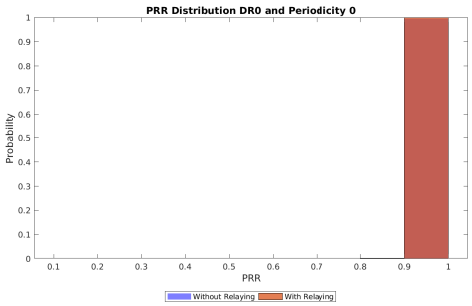
Table 5.1

Therefore, the algorithm has to be activated – while setting up the multicast session – in consultation with the application-server depending on the ping-slot periodicity and the gateway loads. The higher the load the less ping-slots are utilized by gateway and hence ping-slot relaying should be activated even for the higher ping-slot periodicities. The lower the load the more the ping-slots are utilized by the gateway, and hence the algorithm will be limited to higher ping-slot periodicities.

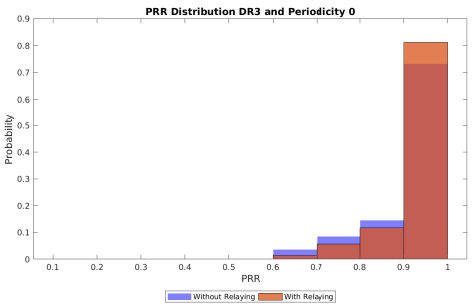
5.1.3 Evaluation of ping-slot relaying

To see the performance of the ping-slot relaying, we re-run the simulation we did in Section 5.1 with the ping-slot relaying enabled. As the ping-slot periodicity 0, for all data-rates, gives a time-off duration that include at least 1 ping-slot, we set the network-server to transmit packets at maximum rate possible. We run for both Nakagami $m = 1$ and Nakagami $m = 0.5$; the resulting distribution is shown in Figure 5.1.3 and Figure 5.1.3 respectively. Because ping-slot relaying utilizes space diversity, we can see that the performance of the ping-slot relaying increases when the fading was increased from Nakagami $m = 1$ to Nakagami $m = 0.5$. Furthermore, we can see from both figures that the algorithm has shown improvement in the distribution by moving end-devices from lower PRR to higher PRR.

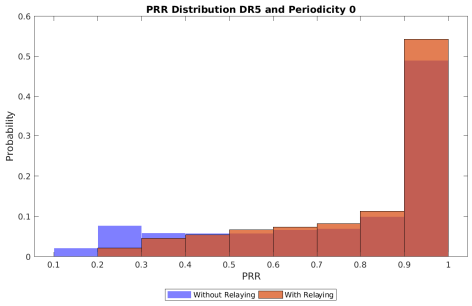
To show how this enhancement in multicast capacity improves scalability, we calculated the expectancy of retransmissions using the distributions given in Figure 5.1.3. We assign PRR to a given number of member end-devices (N) according to the distribution. Now, Let Ti be the number of retransmissions for end-device i, where $i \leq 1 \leq i \leq N$. Assuming the probability of reception for two transmissions are independent with one another, Ti is geometrically distributed. And, assuming that the reception of one



(a) DR0 and periodicity 0.



(b) DR3 and periodicity 0.



(c) DR5 and periodicity 0.

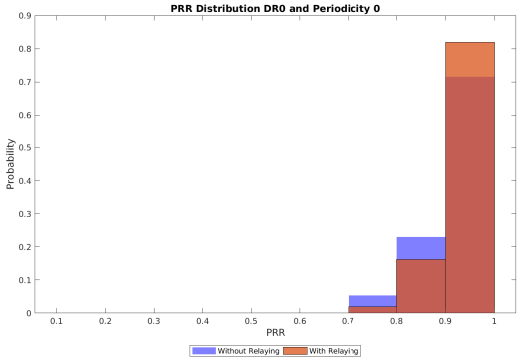
Figure 5.5: PRR distribution of LoRaWAN Class B multicast for Nakagami $m = 1$ fading model, before and after relaying.

end-device is mutually independent with the others, the maximum number of retransmission necessary for all the end-devices to receive a transmitted packet can be expressed as a multiplication of their CDF (Cumulative Distribution Function) as shown in Figure 5.1.

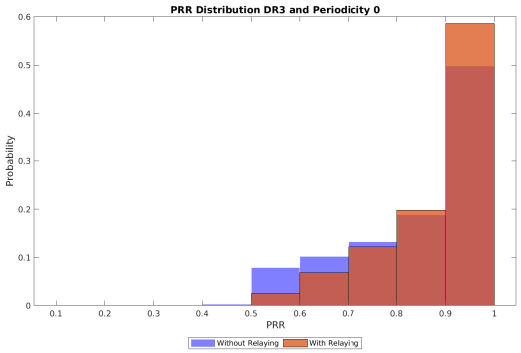
$$P(\max_{1 \leq i \leq N}(T_i) \leq t) = \prod_{i=1}^{N} P(T_i \leq t) \tag{5.1}$$

Let T be the total number of retransmissions required for all the nodes to successfully receive a packet. Then, the expected number of retransmissions ($E(T)$) for all the end-devices to receive a packet can be computed using Equation 5.2, because the number of retransmissions is always a non-negative integer.

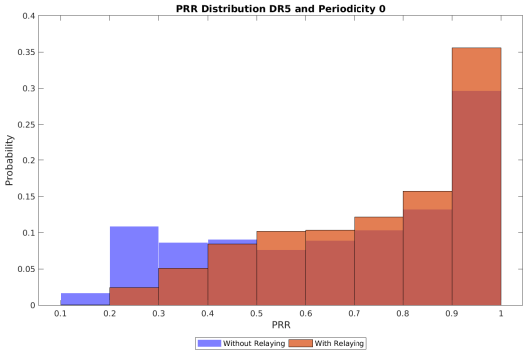
$$E(T) = \sum_{t=0}^{\infty} P(\max_{1 \leq i \leq N}(T_i) > t)) \tag{5.2}$$



(a) DR0 and periodicity 0.



(b) DR3 and periodicity 0.



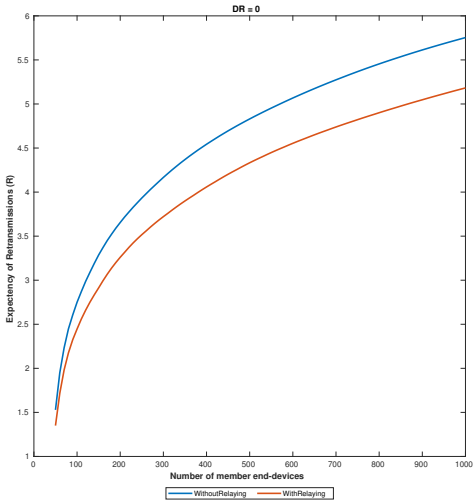
(c) DR5 and periodicity 0.

Figure 5.6: PRR distribution of LoRaWAN Class B multicast for Nakagami $m = 0.5$ fading model, before and after relaying.

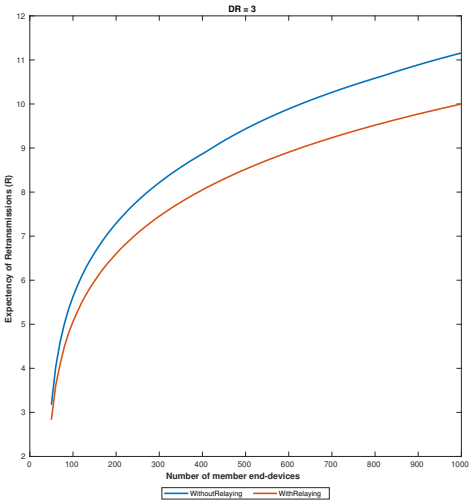
Now, combining Equation 5.1 with Equation 5.2, we drive Equation 5.3.

$$E(T) = \sum_{t=0}^{\infty}(1 - \prod_{i=1}^{N} P(T_i \leq t)) \tag{5.3}$$

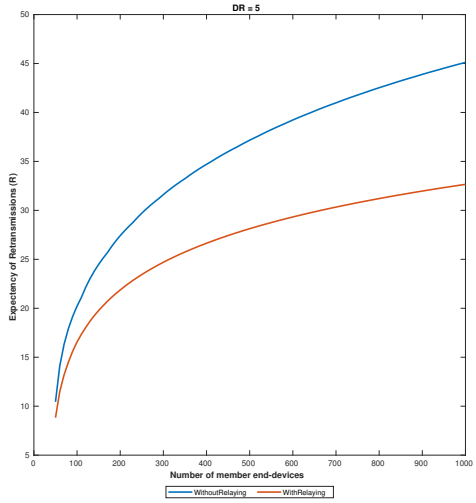
We numerically computed Equation 5.3 and plotted the expected number of retransmissions required to deliver a packet to all multicast members as the number of member end-devices (N) increases, in Figure 5.7. *From the plot, we can see that the number of retransmissions reduced by the ping-slot relaying technique increases with the number of member end-devices. Furthermore, such improvement is even higher as the multicast capacity gets worse. For example, when the number of member end-devices is around 800 nodes, DR0 and DR3 are improved by 10% and DR5 (which has the worst capacity) is improved by 30%.*



(a) DR0 and periodicity 0.



(b) DR3 and periodicity 0.



(c) DR5 and periodicity 0.

Figure 5.7: The expectancy of retransmissions required to successfully deliver a packet to all member end-devices in Nakagami $m = 0.5$ channel fading, before and after relaying.

5.2 Class B coexistence with Class A: end-device

The LoRaWAN specification [10] mandates, after an end-device switched to Class B, that it should still be able to send Class A uplinks and if necessary receive downlinks in the Rx1/Rx2. This can lead to conflicts between these two classes in the end-devices, as there is no mention in the specification how these two classes should co-exist and which one should be prioritized. Francois Delobel et al. [19] has mentioned this problem when listing Class B unicast limitations. And Class B multicast is not any different and is subject to same issue. This conflict could happen in three forms: when the uplink overlaps either with ping-slot window or ping-slot reception, when the Rx1 receive window overlaps with the ping-slot window, and when the Rx2 receive window overlap with the ping-slot window.

The magnitude of this conflict depends on the ping-slot periodicity of Class B, the data-rate of both the Class B ping-slots and Class A transmission, and the air-time of the Class A transmission. Figure 5.8 shows the worst-case conflict between these two classes for all ping-slot periodicities. From the figure we can see that lower ping-slot periodicities are more vulnerable to conflict: periodicity of 3 and above leads to a maximum of one conflict per transmission, whereas periodicities bellow 3 could lead to a conflict more than once following a single transmission.

Dealing with these conflicts, however, is not straight forward. Each decision has it pros and cons and careful consideration has to be made before implementing one. The following section delves into possible strategies to resolve these conflicts along with their corresponding pros and cons.

5.2.1 Prioritizing the Class A uplinks

One way to handle this conflict is by prioritizing Class A uplinks over Class B so that the Class A operation stays the same. This method works best if the transmission happens without collision and the network-server successfuly recieved it. In this case, the network-server can refrain from using the overlaping ping-slots for the multicast transmission, given that the unicast device address and the multicast group address are associated in the network-server.

Nevertheless, we can not be sure that a Class A transmission will always successfully reach the gateway. Furthermore, since the ping-slots are the slots where the network-server sends Class B multicast downlinks, overlapping transmissions with the ping-slots increase the probability of missing both the uplink and the multicast transmissions. Either way, if the network-server does no know that the pings are not opened, it may lead for a multicast transmission to fail in the end-device.

If this happens frequently to multiple members of a multicast group this will impact the overall multicast performance. And since Class B multicast

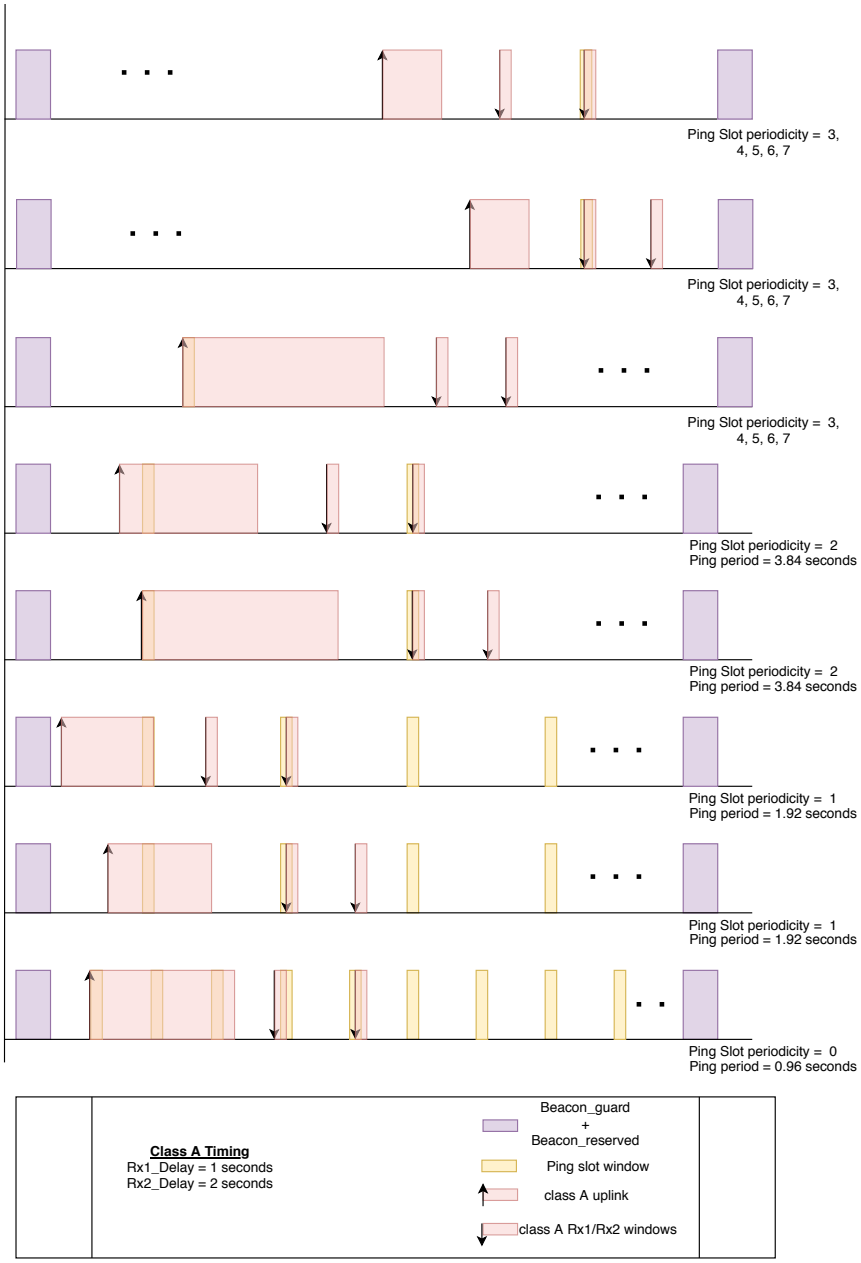


Figure 5.8: Worest case conflict between Class A and Class B.

transmission involve multiple end-devices in which both detecting a packet failure and retransmission are expensive, such solution does not scale well with number of member end-devices.

5.2.2 Prioritize Class B over Class A uplinks

In this technique, the Class A uplinks are always postponed to a later time until there is no conflict. This is greatly advantagous for the Class B multicast operation as it will be kept unaffected. Nevertheless, this will increase the delay of uplink transmission as the ping-slot periodicity decreases and as the air-time of transmission increases. In the worst case, this might lead to starvation in which there is no space to fit the uplink transmission to. This method, therefore, will no always work.

5.2.3 Best effort uplink scheduling

This technique is a modification of the solution in the previous section (Section 5.2.2) however it improves the uplink delay and resolves the starvation problem. Here, rather than postponing the uplink transmission until a suitable space is found, the MAC layer will schedule a conflicting upink, right after the end-of the conflicting ping-slot; or if a packet is received during that ping, it schedules it at the end of the reception. This makes sure that an uplinks have minimum impact on class B ping slots, while at the same time experiencing the minimum delay possible.

5.2.4 Best effort uplink scheduling with downlink estimation

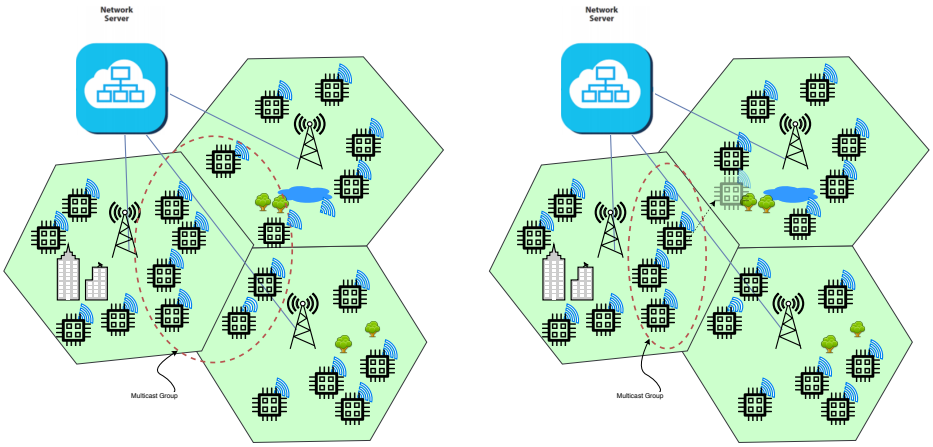
The technique given in Section 5.2.3 minimizes the level of conflict between Class B multicast ping-slots and the uplink transmissions. Nevertheless, for the lowest ping-slot periodicities such as ping-slot periodicity 0, this might not totally avoid it.

The next enhancement we introduce to the solution is including multicast downlink estimation to avoid conflicts that might lead to packet-loss. This can simply be implemented by including multicast downlink rate as one of the paramters when setting up a multicast group. The end-devices can then use this information in time of conflict.

5.3 Multicell scaling

Multicast members of end-devices can span over a number of cells rather than being concentrated in a single cell. This may be because the multicast group includes end-devices in multiple cells as shown in Figure 5.9a , or because of a mobile end-device has moved to a neighboring cell as shown in Figure 5.9b. In both cases, a multicast transmission involves multiple cell and hence multiple gateways.

LoRaWAN gateways are, however, duty cycle limited and all of them may not be available at the same time during the time of multicast. Since they also serve other Class B multicast groups, unicast end-devices and Class



(a) Multicell multicast groups.

(b) A moving member of a multicast group moving to a neighboring cell.

Figure 5.9: Multicell multicast member.

A donwlinks some of the gateways might either be busy transmitting or unable to transmit due to duty cycle limitation. And, as the number of cells involved in a multicast transmission increases, the probability of finding all gateways free for transmission became more difficult. Equation 5.3 shows the probability of a multicast transmission ($P(multicellTransmission)$) in all n cells with n gateways, where each gateway has a probability $P_{gateway}$ of being free at the particular multicast transmission.

$$P(multicellTransmsission) = P_{gateway}(1) \times P_{gateway}(2) \times ... \times P_{gateway}(n) \tag{5.4}$$

In order to deal with this problem, one possible method is to reserve a slot in all gateways, for a particular multicast session. The slot has to be reserved in such a way that the gateways involved in a multicast transmission are neither transmitting nor in their time-off period during these slots. We know, however, from Section 4.1 that reserving a slot in LoRaWAN gateways is not easy and has an impact on the overall performance of a cell. Therefore, a slot-reservation mechanism should be designed in such a way that it will not hamper the operation of cells, while at the same time minimizing waiting delay for the multicell multicast transmissions. We leave this problem as a question for future research.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis we studied the scalability aspect of LoRaWAN Class B multicast by dissecting it into two problems: group scalability and member scalability. We accomplished this by studying related specifications and by extending an already existing LoRaWAN Class A ns-3 module to accommodate LoRaWAN Class B multicast. *This makes this thesis the first work in LoRaWAN Class B multicast as well the simulator the first simulator to simulate LoRaWAN Class B multicast.*

Under group scalability we have *re-evaluated the gateway beacon blocking behavior and reached a different conclusion than the previous study done on LoRaWAN Class B unicast end-device scalability.* We have seen that the beacon blocking does not have a consistent behaviour and that it could either get better or worse after the number of mulitcast groups (or end-devices in case of unicast) reach saturation[1], depending on the airtime of the packet and the transmission pattern. We also have shown that lower ping-slot periodicities reach to this saturation with fewer number of groups compared to higher ping-slot periodicities. *We then explored three novel techniques that 100% prevent beacon blocking. Each technique improves on the previous technique to allow more Class B downlink transmissions without blocking the beacons.* We ended our study on group scalability, by looking into the overhearing and collision as well as throughput performance. We have shown that *proportion of overheard packets to received packets has already began to be significant with just 32 multicast groups.* Hence, *although increasing the number of gateways per cell might improve the throughput, the performance will still be hindered by the overheard packets which then will create collision.* The effect of data-rate and periodicity was also included in the study. We

[1]When the number of multicast groups or unicast end-devices exceed the ping-period.

have seen that lower data-rates has increased the ratio of overheard packets as well as worsened the throughput. The lower ping-ping slot periodicities, on the other side, significantly improved the throughput while increasing the overhearing ratio.

The second part of our study – multicast member scalability – mainly focuses on multicast capacity. We defined mulicast capacity with PRR probability distribution for member end-devices and studied it for different data-rates. The results show that although higher data-rates have better throughput compared to the lower data-rates, they have less capacity when we come to accommodating more member end-devices. Therefore, we concluded that, although *higher data-rates are good choices for smaller number of end-devcies, lower data-rates are better option as the number of member end-devices increase.* We also demonstrated using m-Nakagami model that in the presence of channel fading ($m = 0.5$) all data-rates are subject to significant multicast capacity degradation. To mitigate this, *we developed a novel extension to the standard LoRaWAN Class B multicast operation that improves the multicast capacity. It uses the ping-slots of the end-devices to assist the gateways in their time-off duration. For around 800 member end-devices, it improved the number of retransmissions for DR5 – which has the worst multicast capacity – by 30%, whereas for lower data-rates – which relatively have better multicast capacity – it has shown around 10% improvement.*

The member scalability study then focused on the Class A coexistence with Class B and multicell multicast scalability. We have shown that the conflict between Class A uplinks and Class B ping-slots is more significant for lower ping-slot periodicities. Four possible conflict resolutions were proposed which focus on minimizing the impact both on the Class A transmission as well as Class B multicast performance. The study ends with a brief explanation about the multicell multicast member scaling problem. A ping-slot reservation technique was proposed as one solution that could minimize mutlicast waiting time – the time to wait until all required gateways are available for mutitcasting.

6.2 Future Work

This work can be extended further for more performance study and improvement of the various techniques suggested.

- As this thesis mainly focused on using single gateway, reinvestigating both the group and member scalability performance in presence of multiple gateway could give interesting insights.

- Battery consumption – which is also another important performance metrics for IoT – was not studied in this thesis. Hence, studying the

battery consumption for the overhearing as well as for multicast session transmissions could shed more light on Class B multicast performance.

- Combining erasure coding schemes (such as DaRe [27] or the one proposed by the specification) with ping-slot relaying is also not explored in this thesis. Further research on this could be a good extension for the ping-slot relaying technique proposed.

- The ping-slot relaying technique suggested and evaluated in this thesis purely exploits capture effect and hence it is not loss free. A study in the collision of identical LoRaWAN packets and constructive interference methods such as glossy [14] [28], could dramatically improve the performance.

- The techniques explored to protect beacon blocking can be extended in such a way that the beaconless operation mode is also exploited for optimizing performance by purposefully skipping beacons when required.

- Finally, as already alluded in Section 5.9, gateway reservation techniques for Class B multicast transmission over multiple cell can be a potential research with practical significance for LoRaWAN operators.

Appendix A

Theoretical calculation for the number of ping-slots that could result in beacon blocking

Table A.1 lists the number of ping-slots that could result in beacon blocking if transmission takes place in those slots when the maximum packet size is used for each data-rate. The list is calculated by computing the number of ping-periods ($n_{pingPeriod}$) that the combined air-time of a packet (T_{packet}) and its corresponding time-off duration ($9T_{packet}$) cover beyond the beacon-guard ($beaconGuard$) as given in Equation A.5. Zero or negative values show that the transmission will not result in beacon blocking where as any number beyond zero show that beacon blocking might follow a transmission. The integral part of the calculated number of ping-slots represents the number of pings that would result in beacon blocking irrespective of the pingOffset, and the fractional part represents the percentage of a pingOffsets that could result in beacon blocking for the ping-slots that may or may not lead to beacon blocking depending on the pingOffset. As we can see from the table, none of the data-rates have zero beacon blocking ping-slots if the maximum packet size is used for transmission. The maximum air-time of a packet (T_{packet} in Equation A.5) in turn is calculated by using the maximum payload allowed for each data-rate as given in [7]. The corresponding physical layer payload (PL) is then calculated using the MAC structure given in [10]. Equation A.1, Equation A.2 and Equation A.4 can then be used in order to calculate the maximum air-time for the maximum packet for each data rate [13]. Equation A.4 takes the SF (spreading factor), PL (the physical layer payload length), CRC (either 0 or 1, depending on whether a CRC is used or not), IH (which is 1 if the implicit header mode is used) and DE (which is one if low data rate optimization is enabled) to calculate payload

in terms of symbols ($n_{payload}$). Since for downlink, the explicit header mode is used together with a CRC [10], we used this configuration to calculate $n_{payload}$. Equation A.2 calculates the actual payload air-time by multiplying the number of symbols ($n_{payload}$) by the duration of each symbol given by T_s which can be calulated using Equation A.3. Finally, Equation A.1 calculates the total air-time of the transmission by adding the air-time of the payload together with the air-time of the preamble.

$$T_{packet} = T_{preamble} + T_{payload} \tag{A.1}$$

$$T_{payload} = n_{payload} \times T_s \tag{A.2}$$

$$T_s = \frac{2^{SF}}{BW}, \text{ where BW is bandwidth} \tag{A.3}$$

$$n_{payload} = 8 + max(\lceil \frac{8PL - 4SF + 28 + 16CRC - 20IH}{4(SF - 2DE)} \rceil (CR + 4), 0) \tag{A.4}$$

$$n_{pingPeriod} = \frac{T_{packet} + 9T_{packet} - beaconGuard}{pingPeriod} \tag{A.5}$$

Table A.1: Number of pings that could lead to beacon blocking per beacon period when the maximum packet size is used.

Data-rate	**Ping-slot periodicity**							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	22.560	11.280	5.640	2.820	1.410	0.705	0.353	0.176
1	10.571	5.286	2.643	1.321	0.661	0.330	0.165	0.083
2	4.150	2.075	1.037	0.519	0.259	0.130	0.065	0.032
3	3.926	1.963	0.981	0.491	0.245	0.123	0.061	0.031
4	3.707	1.854	0.927	0.463	0.232	0.116	0.058	0.029
5	0.718	0.359	0.179	0.090	0.045	0.022	0.011	0.006

Appendix B

Maximum packet size for each ping-slot to prevent beacon blocking

In Section 4.1.2 we have described a technique to limit the packet-size per ping-slot in-order to avoid systematic beacon blocking. The results are plotted in Figure B.1. As we can from the figure, the ping-slots that are closer to the beacon guard are still utilized for smaller packets. The figure also demonstrates that this technique is not suitable for lower ping-slot periodicity which have smaller number of pings per beacon window and hence are always limited to lower packet size irrespective of the ping-offset. This gets, worse with lower data-rates. For example, DR0 and DR1 can not be used to transmit any packet if the ping-slot periodicity is 7.

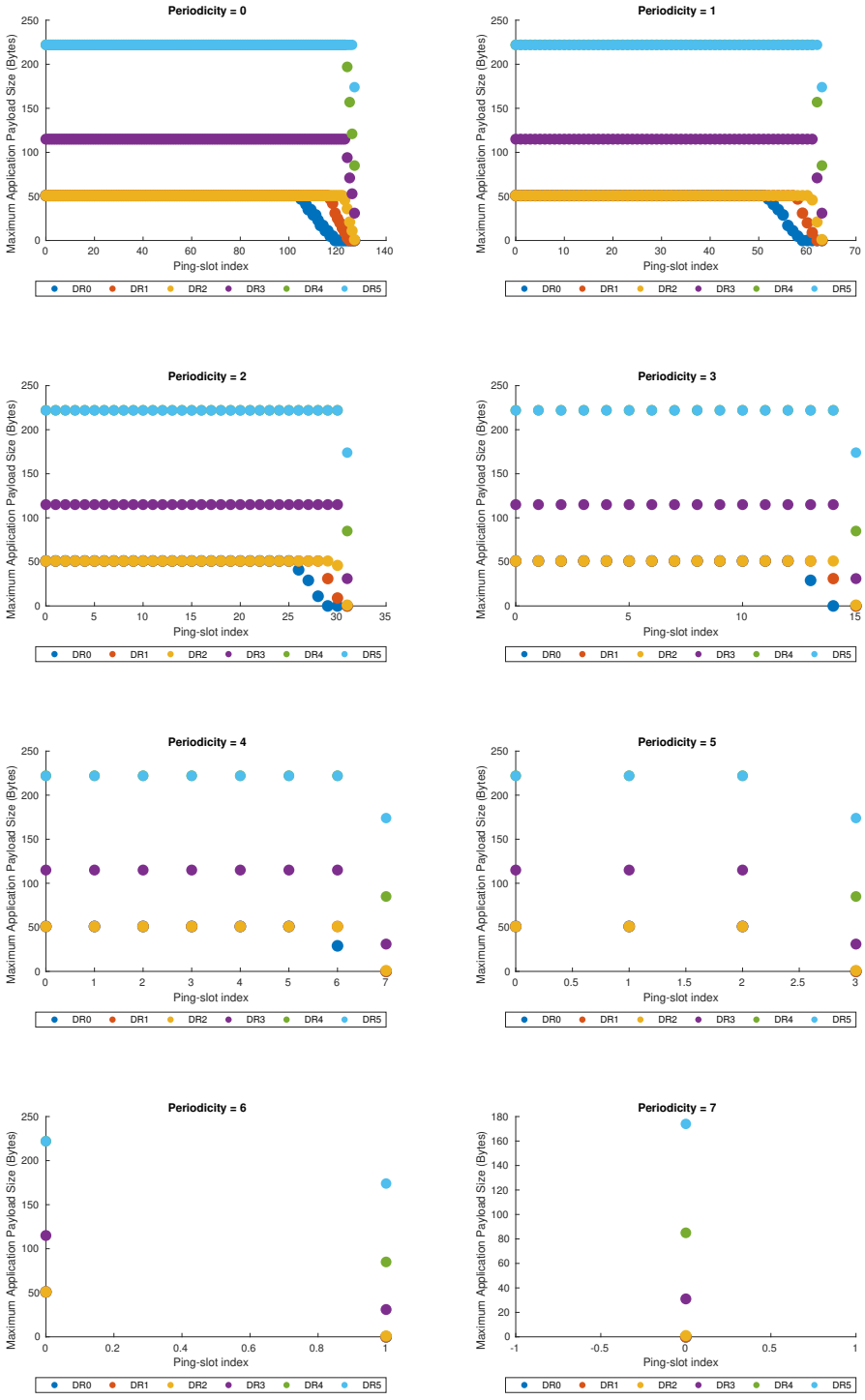


Figure B.1: Maximum application payload size per ping-slot

Bibliography

[1] R. Arias, K. L. Lueth, and A. Rastogi. The effect of the Internet of Things on sustainability — World Economic Forum. Accessed on: 2019-08-01. [Online]. Available: https://www.weforum.org/agenda/2018/01/effect-technology-sustainability-sdgs-internet-things-iot/

[2] LoRa-Alliance, "About LoRa Alliance® — LoRa Alliance." [Online]. Available: https://lora-alliance.org/about-lora-alliance

[3] Vit Prajzler, "LoRa Gateways and Concentrators — LORIOT.io," 2015. [Online]. Available: https://www.loriot.io/lorawan.html

[4] Semtech, *LoRaModulation Basics Semtech*, May 2015.

[5] Techplayon, "LoRa Link-budget and Sensitivity Calculations - Example Explained !" 2017. [Online]. Available: http://www.techplayon.com/lora-link-budget-sensitivity-calculations-example-explained/

[6] T. M. Workgroup, *A technical overview of LoRa and LoRaWAN*, November 2015.

[7] LoRaAlliance, *LoRaWAN 1.1 Regional Parameters*, 2017.

[8] ETSI, *Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1 000 MHz frequency range with power levels ranging up to 500 mW; Part 1: Technical characteristics and test methods*, 2014.

[9] Semtech, *ETSI Compliance of the SX1272 / 3 LoRa Modem*, July 2013.

[10] LoRaAlliance and LoRa Alliance, *LoRaWAN 1.1 Specification*, 2017.

[11] Lennart Nordin, "LoRaWAN Device Classes: A, B and C." [Online]. Available: https://zakelijkforum.kpn.com/lora-forum-16/lorawan-device-classes-a-b-and-c-10972

[12] D. J. J. B. Miguel Luis, Gregory Cristian, "Lora-net/loramac-node," https://github.com/Lora-net/LoRaMac-node, 2019.

[13] Semtech, "SX1272/73 Datasheet Rev. 3.1," no. Rev. 3.1, p. 129, 2017. [Online]. Available: http://www.semtech.com/images/datasheet/sx1272.pdf

[14] M. Bor, J. Vidler, and U. Roedig, "Lora for the internet of things," in *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '16. USA: Junction Publishing, 2016, pp. 361–366. [Online]. Available: http://dl.acm.org/citation.cfm?id=2893711.2893802

[15] A. Rahmadhani and F. Kuipers, "When lorawan frames collide," in *Proceedings of the 12th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, ser. WiNTECH '18. New York, NY, USA: ACM, 2018, pp. 89–97. [Online]. Available: http://doi.acm.org/10.1145/3267204.3267212

[16] Tech-FAQ, "What is Multicast Wireless?" [Online]. Available: http://www.tech-faq.com/what-is-multicast-wireless.html

[17] U. States, *LoRaWAN Remote Multicast Setup Specification*, 2018.

[18] N. (Semtech), A. (Actility), J. (Kerlink), J.-P. (STMicroelectronics), and J. (The Things Network Foundation), *LoRaWAN Fragmented Data Block Transport Specification v1.0.0*, 2017.

[19] F. Delobel, N. El Rachkidy, and A. Guitton, "Analysis of the Delay of Confirmed Downlink Frames in Class B of LoRaWAN," *IEEE Vehicular Technology Conference*, vol. 2017-June, 2017.

[20] J. Finnegan, S. Brown, and R. Farrell, "Evaluating the Scalability of LoRaWAN Gateways for Class B Communication in ns-3," *IEEE Conference on Standards for Communications and Networking (CSCN)*, 2018.

[21] X. Yang, E. Karampatzakis, C. Doerr, and F. Kuipers, "Security vulnerabilities in lorawan," in *Proceedings of ACM/IEEE International Conference on Internet of Things Design and Implementation, IoTDI 2018*. IEEE, 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8366983/

[22] ns3- Network Simulator, "ns-3 Tutorial Release ns-3.29," , 2019. [Online]. Available: https://www.nsnam.org/docs/release/3.29/tutorial/ns-3-tutorial.pdf

[23] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale lorawan networks in ns-3," *IEEE*

INTERNET OF THINGS JOURNAL, vol. 4, no. 6, pp. 2186–2198, 2017. [Online]. Available: http://dx.doi.org/10.1109/JIOT.2017.2768498

[24] B. Reynders, Q. Wang, and S. Pollin, "A lorawan module for ns-3: Implementation and evaluation," in *Proceedings of the 10th Workshop on Ns-3*, ser. WNS3 '18. New York, NY, USA: ACM, 2018, pp. 61–68. [Online]. Available: http://doi.acm.org/10.1145/3199902.3199913

[25] D. Magrin, M. Centenaro, and L. Vangelista, "Performance evaluation of LoRa networks in a smart city scenario," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, may 2017, pp. 1–7. [Online]. Available: http://ieeexplore.ieee.org/document/7996384/

[26] A. Goldsmith, *Statistical Multipath Channel Models*. Cambridge University Press, 2005, p. 6498.

[27] P. J. Marcelis, V. Rao, and R. V. Prasad, "Dare: Data recovery through application layer coding for lorawan," in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, ser. IoTDI '17. New York, NY, USA: ACM, 2017, pp. 97–108. [Online]. Available: http://doi.acm.org/10.1145/3054977.3054978

[28] C. H. Liao, G. Zhu, D. Kuwabara, M. Suzuki, and H. Morikawa, "Multi-Hop LoRa Networks Enabled by Concurrent Transmission," *IEEE Access*, vol. 5, 2017.