

Evaluation of Receiver-Feedback Techniques for Fragmentation over LPWANs

Sergio Aguilar, *Member, IEEE*, Rafael Vidal, and Carles Gomez

Abstract—The Internet Engineering Task Force (IETF) has standardized a new framework for IPv6 support over Low Power Wide Area Networks (LPWANs), called Static Context Header Compression and Fragmentation (SCHC). SCHC includes acknowledgment (ACK)-based mechanisms for reliable fragmented packet transmission. For the latter, SCHC defines a Receiver-Feedback Technique (RFT), called Compressed Bitmap (CB), by which a receiver reports to the sender whether the fragments carrying a packet have been received or not. Such information is carried as ACK payload. Considering the extraordinary frame size and message rate constraints of LPWANs, ACK payload size becomes crucial. In this paper, we compare the performance of CB with that of several alternative RFTs, namely List of Lost Fragments (LLF), List of Deltas (LoD), and Uncompressed Bitmap (UB), where the latter is used as a benchmark. We evaluate the considered RFTs in terms of ACK size, number of Layer 2 (L2) frames needed to carry an ACK, and ACK Time on Air. Our analysis shows that the use of RFTs different from CB offers significant performance improvement in many scenarios. Furthermore, we provide guidance on which RFT should be used for different packet sizes, error rates and error patterns.

Index Terms—LPWAN, fragmentation, IPv6, IETF, IoT, LoRaWAN, Sigfox, SDNV, SCHC.

I. INTRODUCTION

IN recent years, Low Power Wide Area Network (LPWAN) technologies have emerged as a new category of low-power wireless communication solutions for Internet of Things (IoT) applications. Examples of prominent LPWAN technologies include LoRaWAN, Sigfox or Narrowband IoT (NB-IoT) [1]–[3]. LPWANs follow a simple star network topology, where IoT devices are connected directly to a radio gateway, while the supported link range is, typically, in the order of kilometers [4]–[8]. This approach offers low network infrastructure cost, often at the expense of extraordinarily reduced bit rate, message rate and frame payload size.

In order to exploit the full potential of LPWAN devices and technologies, LPWANs need to be provided with Internet connectivity. However, the main current LPWAN technologies have been designed without native Internet Protocol (IP) support. With the aim to enable use of IPv6 (and related upper layer protocols) for LPWAN devices, the Internet Engineering

Task Force (IETF) LPWAN Working Group (WG) was created [9].

The main product of the IETF LPWAN WG is the specification of an adaptation layer framework, called Static Context Header Compression and Fragmentation (SCHC) [10], [11]. The need for this solution is justified by the fact that prior efforts to support IPv6 over low-power wireless technologies, such as 6LoWPAN or 6Lo, yield a too high overhead in the light of the severe constraints of LPWAN technologies [4]. In order to overcome this issue, SCHC defines ultra-lightweight header compression, as well as LPWAN-tailored fragmentation mechanisms. This paper focuses on aspects of the latter.

In order to adapt to the potentially diverse requirements of different LPWAN technologies and deployments, SCHC offers different fragmentation modes, namely: No-ACK, ACK-Always, and ACK-on-Error. The last two modes support acknowledgments (ACKs) and selective fragment retries. In both ACK-Always and ACK-on-Error, the receiver generates (upon fragment loss in the latter) a selective ACK after a group of fragments have been sent. That is, the ACK informs the sender about which fragments have been received or not from the considered group of fragments. The way in which such information is encoded is given by the Receiver-Feedback Technique (RFT) defined in SCHC, which is called Compressed Bitmap (CB). As per CB, the ACK payload carries a bitmap where the k -th bit of the bitmap indicates whether the k -th fragment has been received or not. In some cases, the bitmap may be compressed, which represents a performance optimization, compared with early versions of SCHC that made use of a simple Uncompressed Bitmap (UB). However, the performance of CB has not been evaluated, and, to the best of our knowledge, alternative RFTs have not been considered for SCHC.

In this paper, we investigate the performance of CB, along with that of two alternative RFTs called List of Lost Fragments (LLF) and List of Deltas (LoD). LLF is a binary-encoded list of Fragment Numbers (FNs) that correspond to lost fragments. LoD is based on the differences (deltas) between the FN of consecutive lost fragments. For efficiency, deltas are encoded with variable length formats using Self-Delimiting Numerical Values (SDNV) [12]. Regarding the latter, we investigate 4 different approaches which use base encoding format sizes of 2, 3, 4 and 5 bits. We also include UB, as a benchmark, in our study.

We evaluate the performance of the considered RFTs by means of extensive simulation. To this end, we developed Sim-RFT, an ad-hoc simulator that allows to analyze RFT performance for different fragmented packet sizes, and under

Manuscript received December 28, 2020; accepted September 14, 2021. This work was supported in part by the Spanish Government through project TEC2016-79988-P and project PID2019-106808RA-I00, AEI/FEDER, EU

S.Aguilar, C. Gomez and R. Vidal are with the Department of Network Engineering, Universitat Politècnica de Catalunya, 08860 Castelldefels, Barcelona, Spain (e-mail: sergio.aguilar.romero@upc.edu, carlesgo@entel.upc.edu, rafael.vidal@entel.upc.edu)

Copyright ©2021 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

different error rates and patterns. The main performance parameters evaluated are the ACK payload size, the number of Layer 2 (L2) frames required to carry an ACK, and the ACK Time on Air (ToA). Our results show that CB only outperforms the alternative RFTs considered for short fragmented packet sizes, or under high error rates.

The remainder of the paper is organized as follows. In Section II we review existing work related to SCHC and fragmentation over LPWANs. We overview SCHC reliable fragmentation over LPWANs in Section III, and describe the RFTs considered in this paper in Section IV. In Section V, we present Sim-RFT, along with the configuration settings and error patterns used in the study. In Section VI, we evaluate the performance of the RFTs under a range of conditions, and discuss the obtained results. Finally, we provide the main conclusions from this work in Section VII.

II. RELATED WORK

In this section, we review research works that focused on SCHC and fragmentation over LPWANs. Some of them investigated both SCHC compression and fragmentation mechanisms [11], [13]–[19], while others focused on fragmentation functionality alone [20], [21].

The authors in [11] provided an overview of SCHC. As a future work item, they proposed a reliable fragment delivery mechanism whereby a single ACK would report on the delivery success or failure of all the fragments that carry a large packet. When the number of fragments per packet is too high, it may be challenging to fit the receiver report in only one L2 frame. Therefore, the authors pointed out the need to consider alternative RFTs, instead of CB, specified in SCHC, which under some conditions may produce a too large ACK payload. However, the authors neither described nor evaluated any alternative RFTs.

Suciu et al. evaluated the efficiency of fragmentation in dense LPWAN networks [20]. However, authors did not consider receiver-feedback mechanisms in their study. Another work defined and evaluated the effect of an aggressive fragmentation strategy for LPWANs, i.e., performing fragmentation even if the packet to be carried fits the L2 frame [21]. The study used UB to report fragment reception status in negative ACKs (NACKs). However, authors neither studied the impact of error patterns on the NACK size, nor considered different RFTs.

Other works analyzed the performance of IPv6 header compression [13]–[15] and/or fragmentation over LPWAN by using SCHC [16]–[19]. Abdelfadeel et al. [13], [14] and Ayoub et al. [15] focused only on SCHC header compression. Moons et al. [16] compared the memory footprint of SCHC header compression and fragmentation with that of a 6LoWPAN-based solution. For SCHC fragmentation, they used UB as RFT. The authors in [17] compared the different SCHC fragmentation methods, assuming an ideal communication channel, in terms of channel occupancy, goodput and delay. A mathematical model to calculate the ACK message overhead of ACK-on-Error mode, and how to optimally tune its most critical parameters, is presented in [18]. However, UB was

assumed in both [17] and [18]. The authors in [19] evaluated SCHC header compression and fragmentation when using an end-to-end CoAP (Constrained Application Protocol) broker to connect LoRaWAN devices by using a publish/subscribe scheme. However, the No-ACK fragmentation mode was used, therefore no RFT was studied in this work.

Based on our literature analysis, and to the best of our knowledge, we conclude that previous work neither evaluates the performance of RFTs different from UB (not even CB, which is the one used in SCHC), nor the impact of different packet sizes, error rates, and error patterns on RFT performance.

III. RELIABLE FRAGMENTATION OVER LPWAN

In order to offer flexibility, considering the diversity of LPWAN technologies and scenarios, SCHC defines three fragmentation modes, namely: No-ACK, ACK-Always, and ACK-on-Error. The latter two modes offer reliable fragmentation. This section provides an overview of SCHC fragmentation, focusing on its reliable fragmentation modes.

In SCHC, when a sender intends to transmit a packet of a size greater than the L2 frame Maximum Transmission Unit (MTU), the packet is broken into smaller pieces. Each one of the pieces becomes the payload of a fragment, which also comprises a fragment header (including a FN), and padding (if required to adapt to the characteristics of the underlying LPWAN technology). The fragment format is illustrated in Fig. 1. Each fragment is carried by one L2 frame.

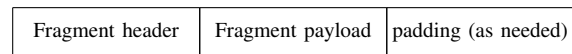


Fig. 1. Fragment format.

In the reliable fragmentation modes, the receiver provides feedback to the sender by means of one or more ACKs. An ACK is composed of an ACK header and an ACK payload. The ACK payload provides information about successful or failed reception of a set of fragments, encoded in a way that is specific to the RFT in use. In SCHC, CB is the standardized RFT, which is an improved version of UB (see Section IV). As an optimization, the ACK payload is not present in the ACK when all corresponding fragments have been correctly received, as shown in Fig. 2. The sender will retransmit any fragments reported lost by an ACK.

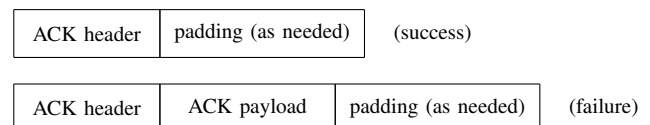


Fig. 2. ACK formats

In ACK-Always, an ACK is sent by the receiver after a group of fragments has been transmitted. In ACK-on-Error, an ACK is only sent when at least one fragment has been lost, with the exception of an ACK that is unconditionally sent at the end of the fragmented packet transmission.

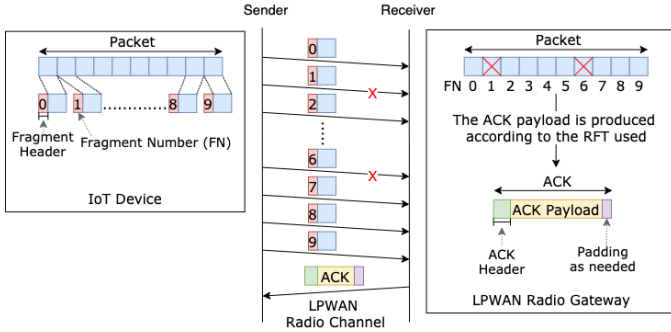


Fig. 3. Example of the transmission of a 10-fragment packet with errors.

In this paper, we assume an efficient configuration of reliable fragmentation where a single ACK provides feedback on the delivery success or failure of the whole set of fragments that transport a packet. If the ACK size cannot fit one L2 frame, each additional L2 frame required includes an ACK header as well.

Fig. 3 presents an example of the transmission of a packet that requires 10 fragments to be carried. Two fragments (with FNs 1 and 6) are lost, whereas the other fragments are correctly received. At the receiver, once the last fragment (which is signaled by a dedicated FN value) is received, an ACK carrying a payload produced by the RFT in use is assembled and transmitted. In the following section, the example shown in Fig. 3 will be used to illustrate the behavior of each RFT studied in this paper.

IV. RECEIVER-FEEDBACK TECHNIQUES FOR LPWAN FRAGMENTATION

In this section, we describe the four RFTs considered in our performance evaluation, namely: UB, CB, LLF, and LoD.

UB was the initially considered RFT for SCHC. In this paper, UB is used as a benchmark. CB, an improved version of UB, is the RFT standardized in SCHC. We introduce LLF and LoD as RFTs with potential to offer good performance in some scenarios. Each considered RFT follows a different approach, and is thus expected to perform differently, depending on conditions such as error rates, error distribution and packet size.

A. Uncompressed Bitmap (UB)

UB was introduced in early stages of the design of SCHC. This RFT is based on representing the sequence of received fragments by means of a sequence of bits, called a bitmap. Each bit in the bitmap corresponds to a fragment of the packet, where the k -th bit is set to 1 or 0 when the k -th fragment has been received or not, respectively. The leftmost bit of the bitmap corresponds to the first fragment of the packet.

Following the example presented in Fig. 3, when UB is used, the receiver generates the bitmap, sets to one the bitmap bits that correspond to successfully received fragments, and sets to zero the remaining bits (in the example, the ones corresponding to FNs 1 and 6). The resulting bitmap has a size of 10 bits, since the packet size is 10 fragments. Finally,

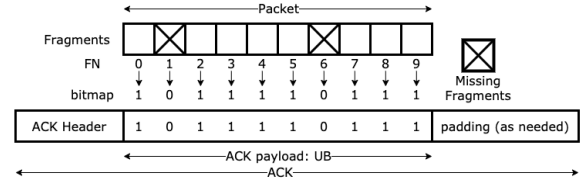


Fig. 4. Example of ACK payload for UB, and ACK format, for the transmission of a 10-fragment packet where two fragments (with FNs 1 and 6) are lost.

the ACK is assembled by prepending the ACK header to the bitmap, as shown in Fig. 4. If needed, padding bits are appended at the end of the bitmap.

B. Compressed Bitmap (CB)

CB is an RFT designed to reduce the size of the bitmap produced by UB, when possible. To this end, a receiver operates as follows. Firstly, a bitmap is built as described in subsection IV-A. Then, in order to compress the bitmap, the receiver analyzes each bitmap bit from right to left. All contiguous bitmap bits set to 1 are removed. The receiver will stop this procedure when a 0 is found or when it reaches the leftmost bitmap bit. The result of this operation is a compressed version of the bitmap. After the ACK header is prepended to the compressed bitmap, the size of the latter may need to be adjusted, depending on the minimum data unit size supported by the underlying LPWAN technology. For example, if that technology is byte-oriented, the minimum number of bitmap bits with value 1 are restored on the right, so that the ACK header plus the ACK payload have a size multiple of an integer number of bytes. In some cases, padding may be needed as well. The sender can reconstruct the original bitmap from the (potentially) compressed bitmap received in the ACK message, as the sender knows the number of fragments sent to carry a given packet.

Following the example presented in Fig. 3, Fig. 5 illustrates how the size of the bitmap obtained in Fig. 4 is reduced from 10 bits (with values 1011110111) to 8 bits (i.e., 10111101), assuming a 1-byte ACK header and a byte-oriented underlying LPWAN technology.

Note that the compression degree that can be achieved with CB depends on which is the last lost fragment carrying data from a fragmented packet. For example, an error in the transmission of the last fragment will not allow compressing the corresponding bitmap when using CB.

C. List of Lost Fragments (LLF)

We define LLF as an alternative RFT that produces the sequence of binary-encoded FNs of the lost fragments (if any) that carry a packet. In contrast with UB, which has a fixed length for a given packet size, LLF produces a variable length ACK payload, which is roughly proportional to the Fragment Error Rate (FER).

Fig. 6 shows the ACK payload that corresponds to the example presented in Fig. 3, where the fragments with FNs 1 and 6 are lost, when LLF is used. The ACK payload comprises these

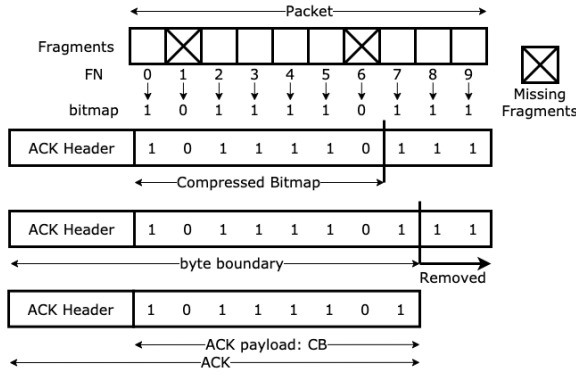


Fig. 5. Example of a bitmap before compression and the corresponding compressed bitmap. The rightmost sequence of consecutive bitmap bits set to 1 is removed to obtain the compressed bitmap. The example assumes a byte-oriented underlying L2 LPWAN technology. In this example, padding is not needed.

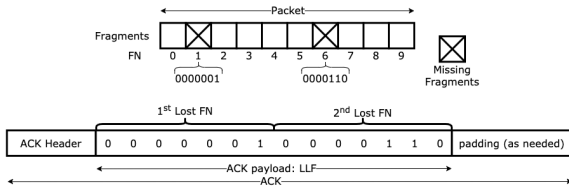


Fig. 6. With LLF, the FNs of lost fragments are converted to binary and appended to the SCHC ACK Header. Finally, padding is added as needed.

two FN values, converted to binary code with a size of 7 bits per FN, thus leading to a 14-bit ACK payload. Note that the 7-bit encoding per FN allows to identify fragments numbered in a range from 0 to 127. Considering a 10-byte fragment payload (which is typical in many LPWAN scenarios), the FN range will allow to unambiguously identify each fragment of a 1280-byte packet, thus allowing compliance with the IPv6 MTU requirement [22].

In general, an LLF implementation can be built with a simple concatenation of the Fragment Numbers of missing fragments in a string. For example, in an MCU that supports C programming language, it can be done by using the “strcpy” function, which is part of the string.h standard library of C. This will represent a code size increase in the order of just a few bytes. Since SCHC includes the calculation of a Reassembly Check Sequence (RCS) based on a Cyclic Redundancy Check (CRC) of 32 bits, which presents significantly more computational complexity than the considered RFTs, an LLF implementation would not add a significant amount of code footprint.

D. List of Deltas (LoD)

We define LoD as an RFT where the receiver reports the differences (hereinafter, deltas) between the FNs of any two consecutive lost fragments. Instead of encoding absolute fragment numbers as in LLF, LoD exploits the smaller expected size of binary encoded deltas. In LoD encoding, the first lost fragment absolute FN is encoded as a reference. Subsequent encoded values are the deltas between consecutive lost fragments’ FNs.

Following the example presented in Fig. 3, when LoD is used, the FN of the first lost fragment (i.e., 1) is encoded as reference value. The next lost fragment corresponds to FN = 6, so a delta of 5 is then encoded. Note that the ACK payload produced by LoD is not only sensitive to the FER but also to the fragment loss distribution.

In order to optimize the LoD encoding, the number of bits to encode each delta needs to be variable, allowing to represent smaller deltas with a lower number of bits. To this end, we use SDNV encoding [12]. This technique allows a simple way of representing non-negative integers efficiently and with variable length. We next describe SDNV in detail.

SDNV represents a number by means of one or more elementary data units, which we refer to as bases. A base is a fixed-length set of bits used to fully or partially encode a number. The most significant bit of a base is a control bit reserved to determine whether that base is the last one for representing a number (in that case, the control bit is set to 0). The remaining bits in a base are data bits, i.e., they are used to encode actual values. Therefore, in each base used there is a 1-bit overhead. If a number cannot be encoded by using just one base, then additional bases are added as needed. In order to encode a number by using SDNV, the following steps are followed:

- 1) The number to be encoded is converted to binary.
- 2) The binary-encoded number bits are encapsulated, from left to right, in as many bases as needed, using the available data bits in each base. If needed, the base with the most significant binary number bit is padded with zeros on the left in order to fill in all data bits of that base.
- 3) The control bit of each base is appropriately set.

The SDNV standard uses a base of one byte, thus there are 7 data bits available in each base. For the sake of efficiency, in the evaluation carried out in this paper, we consider several smaller base sizes. We use the notations SDNV-x and LoD-x to denote the usage of SDNV with a base size of x bits. Fig. 7 shows two examples of how two numbers in decimal (10 and 123) are encoded in SDNV-3 and SDNV-5, as per the steps provided above.

	Example 1		Example 2	
Step 1	10 (decimal) → 0b1010		123 (decimal) → 0b1111011	
Step 2	SDNV-3	SDNV-5	SDNV-3	SDNV-5
Step 3	1 10 0 10	0 1010	1 01 1 11 1 10 0 11	1 0111 1011
SDNV-encoded Number	110010	01010	10111110011	1011101011

Fig. 7. Example of different SDNV-encoded numbers. Data bits are represented in bold font.

Following the example presented in Fig. 3, where fragments 1 and 6 were lost, Fig. 8 illustrates the LoD encoding for different base sizes. In this example, LoD-4 and LoD-2 produce the shortest-sized ACK payload.

Note that SDNV encoding is based on a simple algorithm that requires around 12/13 lines of code for decoding/encoding operations, respectively, when using python and standard libraries. This will represent a code footprint increase in the

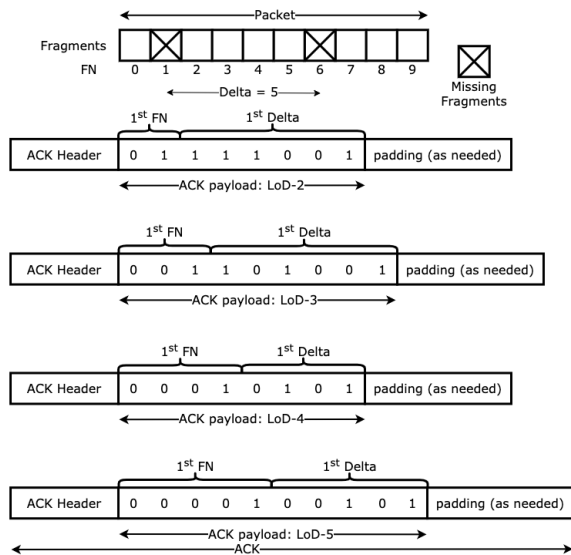


Fig. 8. Examples of ACKs, using LoD, and for the fragment losses shown in Fig. 3 (i.e., fragments with FN=1 and FN=6), for different SDNV bases. The decimal numbers to be encoded are 1 (first FN) and 5 (first delta). All considered SDNV base sizes require one base to encode the FN of the first lost fragment. To encode the delta, SDNV-4 and SDNV-5 only require 1 base, whereas SDNV-2 and SDNV-3 need 3 and 2 bases, respectively.

order of tens of bytes, that depends on the embedded microcontroller, compiler and programming language used. Considering that SCHC involves several operations (e.g., including a 32-bit CRC for integrity checks, as mentioned earlier), LoD would not require a significant amount of additional program storage for an embedded microcontroller.

V. SIMULATION ENVIRONMENT

In this section, we present Sim-RFT, the simulation environment that we use in this paper to compare the performance of the different RFTs introduced in the previous section. We describe how Sim-RFT works and detail the main fragmentation-related parameters assumed in our evaluations, along with the characteristics of the error patterns considered.

A. Main features

Sim-RFT is an ad-hoc tool that we have developed to simulate fragmented packet transmission and reception over a lossy channel. After each simulated fragmented packet transmission, Sim-RFT creates the corresponding ACK, for each RFT.

Sim-RFT provides three main performance parameters from each obtained ACK: i) ACK size, ii) number of L2 frames required to carry the ACK, and iii) ToA for each ACK. These performance parameters take into account the main L2 frame characteristics of the underlying LPWAN technology considered: L2 frame header size, and L2 MTU.

Our study focuses on the fragmented packet first transmission attempt, as it will create the largest ACK payload size for all evaluated RFTs, except for UB, which always yields the same ACK payload size for all transmission attempts.

As a side-contribution of this work, we offer Sim-RFT publicly [23].

TABLE I
LORAWAN MTU VALUES

Channel Plan	Country/ Region	L2 MTU (bytes)	
		Minimum DR	Maximum DR
EU863-870	Europe	51	242
	Middle East		
	Africa		
US902-928	America	11	242
CN779-787	China	51	242
EU433	Europe	51	242
	Middle East Africa		
AU915-928	Australia	11	242
CN470-510	China	51	242
AS923	Asia	19	250
KR920-923	South Korea	51	242
IN865-867	India	51	242
RU864-870	Russia	51	242

B. Settings

In order to maximize the applicability of our evaluation results, in this paper we configure Sim-RFT to use 3 different L2 MTU values: 11 bytes, 51 bytes and 242 bytes, for both the uplink and the downlink. Table I shows the LoRaWAN L2 MTU values for all LoRaWAN channel plans. The 11-byte L2 MTU corresponds to the maximum frame payload size of LoRaWAN US915 Data Rate 0 (DR0) and AU915 DR0. It is also similar to the 12-byte uplink, 8-byte downlink L2 MTU of Sigfox [3], [24], and the 19-byte L2 MTU of LoRaWAN AS923 DR0. The 51-byte L2 MTU corresponds to DR0 for the following LoRaWAN regional bands: EU868, CN779, EU433, CN470, KR920, IN865, and RU864. The 242-byte L2 MTU corresponds to the maximum one in LoRaWAN for the highest DR of all regions (except AS923), and it is similar to the 250-byte L2 MTU of AS923 with its maximum DR (see Table I). Therefore, conclusions from the evaluation will be useful when considering fragmentation over LoRaWAN in all available regions or countries where it is defined, and also over Sigfox. Note that, in a LoRaWAN frame, only an integer number of bytes can be carried. Accordingly, Sim-RFT will add padding bits if required. Appendix A shows the LoRaWAN full set of L2 configuration parameters used in the simulations. Fragment header and ACK header sizes of 1 byte are assumed.

Sim-RFT does not support native L2 retransmission mechanisms (e.g., LoRaWAN confirmed data messages [1]), as it is not required by SCHC (e.g., when used over LoRaWAN [25]), and also because they can have a negative impact on uplink throughput [26] and cause considerable network performance decrease [27].

C. Error Patterns and Rates

In order to evaluate the performance of the considered RFTs in the presence of errors, two different error distributions are supported in Sim-RFT: a uniform error distribution [28]–[57], and a burst error distribution [34]–[50]. These error

distributions cover a comprehensive set of characteristics of LoRaWAN networks such as frame loss over distance [31], [33], [34], [37], [41], [51]–[53], different use cases [31], [35], [54]–[57], mobile or stationary devices [33], [46], [49], [50], network capacity [54], and collisions [33], [53]. Frame loss burstiness may be due to channel effects [34], [36], mobility [34], [35], limited coverage [38], [40], [41], [44], or opportunistic coverage [42]–[44]. Event-driven communication may also lead to burst errors, as many devices may try to communicate at the same time during a relatively long time interval [45]–[48]. Table II summarizes the types of error distributions identified in LoRaWAN literature.

The uniform error distribution is modeled by using a Bernoulli process, with a fragment error probability equal to the FER . The burst error distribution is modeled by using a discrete Markov chain composed of two states [34]: a good state, where there are no fragment errors, and a burst state, where there is a burst of λ_L consecutive fragment errors. The probability of transition from a good state to the burst state is referred to as Burst Occurrence Probability (BOP). λ_L is modeled as a random variable that follows a Poisson distribution (see Fig. 9). Once in the burst state, several fragments are lost, and then the chain transitions back to the good state. Each fragmented packet transmission starts in the good state. We consider that transmitting different fragmented packets corresponds to independent events. Therefore, if a burst length λ_L is larger than the remaining number of fragments to be transmitted, the resulting burst length (λ_{RL}) will be smaller, hence the burst will be truncated ($\lambda_{RL} \leq \lambda_L$). As the fragmented packet size increases, the probability that a burst will be truncated decreases and larger burst lengths are more likely (see Figs. 9a and 9b).

In order to capture the characteristics of a wide range of LoRaWAN scenarios, as reported in prior work, in this paper we consider FER values of 1%, 10% and 20%. $FER = 1\%$ corresponds to good channel conditions, with sporadic fragment errors [28], [31], [50]. A FER of up to 10% is expected in industrial deployments [29], [30], and it can also be found under certain LoRaWAN configurations for static devices [33] and mobile ones [39]. FER up to 20% was found in adverse environments [32], with link distance being the primary cause of losses [41], [50].

On the other hand, we consider BOP values of 1% and 2%, with an average burst size (λ) of 10 fragments (see Fig. 9), which captures burst error characteristics found in the literature. Burst error lengths between 2 and 30 frames have been reported, with the range between 2 and 7 frames corresponding to the most likely burst error length [34]–[38]. By modeling λ_L with a Poisson distribution, with an average burst size of 10 fragments, and considering burst truncation, the resulting burst length (λ_{RL}) distribution concentrates 40% of burst sizes between 2 and 7 fragments, with an actual average burst length of 7 fragments, while still providing burst sizes up to 30 fragments (see Fig. 9).

VI. EVALUATION

In this section, we use Sim-RFT to investigate by simulation the performance of the RFTs presented in Section IV for

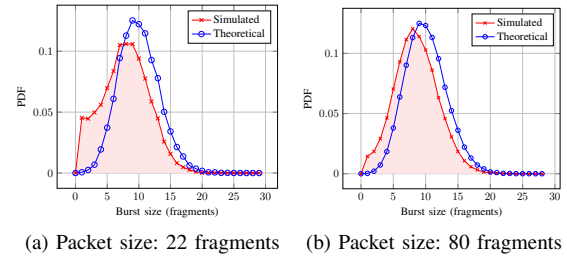


Fig. 9. Probability Density Function (PDF) of the theoretical vs simulated burst length ($\lambda = 10$).

TABLE II
LORAWAN ERROR DISTRIBUTIONS IN LITERATURE

References	Error distribution	
	Uniform	Burst
[28]–[33]	Yes	No
[34]–[57]	Yes	Yes

different FER values, different error distributions (uniform and burst), and for a range of packet sizes that require fragmentation, including the MTU required for IPv6, i.e., 1280 bytes. Note that there exist applications that involve longer packet sizes. These include waveform captures, data logs, and large data packets using rich data types [58]. As introduced in Section V-B, LoRaWAN is assumed as the underlying LPWAN technology. Regarding LoD, we consider base sizes of 2, 3, 4, and 5 bits.

The performance metrics in the evaluation are: i) ACK payload size, ii) number of L2 frames needed to carry an ACK, and iii) ACK ToA, hereinafter ToA, for each considered RFT. The ACK payload size is critical to LPWAN performance. On the one hand, the downlink channel of an LPWAN radio gateway is a bottleneck for the whole network. Note that most LPWAN traffic is sent in the uplink, part of that traffic requires downlink transmissions (e.g., ACKs), and the number of LPWAN devices per radio gateway may be large. Furthermore, there exist spectrum access regulations that restrict the duty cycle in some world regions and frequency bands (e.g., LoRaWAN operates in Europe in the 868 MHz, which is limited to a maximum duty cycle of 1%). Reducing the ACK size, increases the number of IoT devices that can be supported per LPWAN radio gateway. On the other hand, the ACK size has a direct impact on the energy consumption of IoT devices. The number of L2 frames needed per ACK measures the ACK fragmentation overhead. If the ACK size exceeds the L2 frame maximum payload size, additional frame transmissions (including their corresponding L2 headers) are required, reducing efficiency. There may also be a negative impact on cost, as some operators charge by the number of downlink messages sent. Finally, ToA captures the channel occupancy over time due to ACK transmission, which is relevant to the scalability of the LPWAN.

Each individual result provided has been obtained as the average over one million simulations.

A. ACK payload size analysis

In this section, we evaluate the average ACK payload size for the different considered RFTs. In order to capture the full impact of ACK payload size when considering the characteristics of the underlying LPWAN technology, padding bits (if any) are included in the ACK payload size results shown. The section is organized into two subsections, which focus on the performance of the RFTs for the uniform, and for the burst error distributions, respectively.

1) *Uniform losses*: In this subsection, we evaluate the impact of a uniform fragment loss distribution on the ACK payload size for the evaluated RFTs, for FER values of 1%, 10% and 20%, uniform losses, and a range of packet sizes.

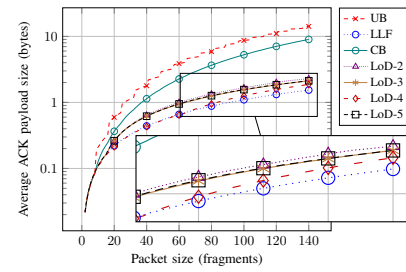
Fig. 10a presents the average ACK payload size for all considered RFTs, and for a FER of 1% and uniform losses. The sizes of the different ACK payloads produced by the different RFTs are very similar for small packet sizes (i.e., packet sizes between 1 and 10 fragments). As packet size increases, the ACK payload size grows very rapidly for UB and CB. UB produces the greatest ACK payload size, which is linear with the number of fragments needed to carry a packet, and has a step-like behavior due to padding. CB offers better performance than UB, although its improvement is limited by losses, which reduce CB's compression gain.

LLF also has a linear behavior with packet size, since the number of fragments lost per packet is, in average, a fraction (approximately equal to the FER) of the total number of fragments required to transport the packet. Since a FER of 1% is low, LLF produces the smallest ACK payload size among the different RFTs evaluated, for all packet sizes considered.

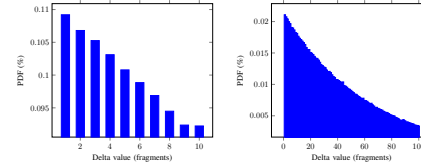
The considered LoD variants exhibit a similar behavior for small packet sizes. However, as packet size increases, LoD-2 tends to produce a greater ACK payload size due to the greater overhead of SDNV-2 when encoding large deltas. LoD-3 and LoD-5 yield a similar ACK payload size, with LoD-4 being the optimal LoD encoding for the conditions considered. LoD-4 performs very similarly to LLF for a small packet size. However, as packet size increases, and since FER is low, deltas tend to increase, and therefore, LoD-4 produces a slightly greater ACK payload size than LLF.

In order to better understand the performance of the different LoD solutions, we analyzed the statistics of the deltas. Figs. 10b and 10c depict the delta PDF for packet sizes of 10 and 100 fragments, for $FER = 1\%$ and uniform losses, respectively. For both packet sizes, the delta PDF decreases steadily, from a delta value of 1, which is the most frequent delta value, up to the packet size (in number of fragments). As the probability of a delta value decreases, its encoded size increases. LoD-4 provides better performance due to its suitable trade-off between low encoding overhead and the relatively large deltas stemming from relatively infrequent errors.

Fig. 11a shows the average ACK payload size for $FER = 10\%$ and uniform losses. Similarly to the study for $FER = 1\%$ (Fig. 10a), for small packet size (of up to 10 fragments), all RFTs generate a similar ACK payload size. As packet size increases, UB/CB and LoD-3/LoD-4 yield the largest and



(a) Average ACK payload size vs packet size



(b) Packet size: 10 frag-ments (c) Packet size: 100 frag-ments

Fig. 10. Delta Probability Density Function (PDF), for $FER = 1\%$ and uniform losses.

the smallest ACK payload sizes, respectively. UB produces an ACK payload size independent of the error rate, whereas CB offers lower improvement than for $FER = 1\%$ due to the more frequent presence of losses at the end of packet transmission for $FER = 10\%$. LLF yields now a greater ACK payload size than the tested LoD schemes, as the ACK payload size is now 10 times greater than the one for a FER of 1%. Regarding the LoD schemes, LoD-2 still requires more bits to encode the deltas than the other considered LoDs, despite the fact that smaller deltas are more frequent for $FER = 10\%$ than for $FER = 1\%$ (see Figs. 10b and 10c, and Figs. 11b and 11c). On the other hand, LoD-3 is a more efficient encoding for $FER = 10\%$ than it was for $FER = 1\%$, leading to an ACK payload size very similar to the LoD-4 one. LoD-5 uses 5 bits to encode small deltas, and therefore produces a slightly larger ACK payload size than LoD-3 and LoD-4.

Fig. 12a shows the average ACK payload size for $FER = 20\%$ and uniform losses. In this case, LLF yields the largest ACK payload size among the considered RFTs, since the LLF ACK payload size is roughly proportional to the FER , which is very high at 20%. The rest of RFTs produce now very similar ACK payload sizes.

Regarding the LoD RFTs, LoD-3 yields the shortest ACK payload size, since deltas are now smaller than for $FER = 10\%$ (see Figs. 12b and 12c) and they can be encoded more efficiently with a 3-bit base than with a 4-bit base. LoD-2 suffers from a too high overhead to encode deltas that are still too large for the short 2-bit base, whereas the same deltas are too small for the larger 5-bit base in LoD-5. For $FER = 20\%$, UB and CB generate an ACK payload size that is in average very similar to the LoD-4 one. The UB ACK payload size remains independent of the error rate, whereas the high FER of 20% leads CB to perform very similarly to UB. Therefore, UB and CB offer relatively good performance for high FER and uniform losses.

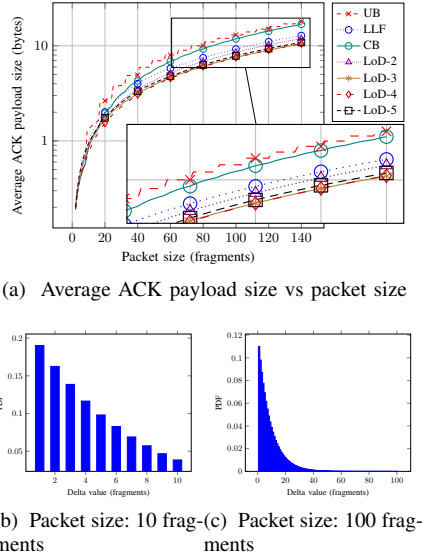


Fig. 11. Delta Probability Density Function (PDF), for $FER = 10\%$ and uniform losses.

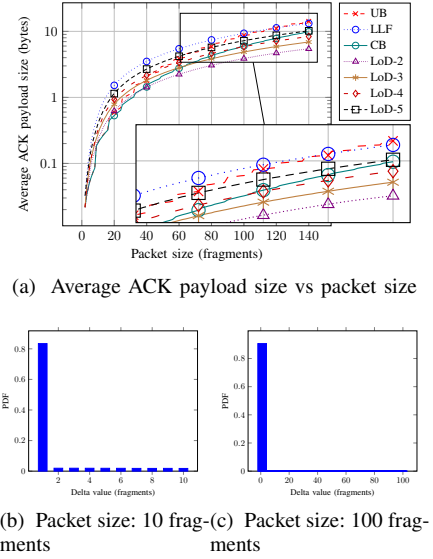


Fig. 13. Delta Probability Density Function (PDF), for $BOP = 1\%$.

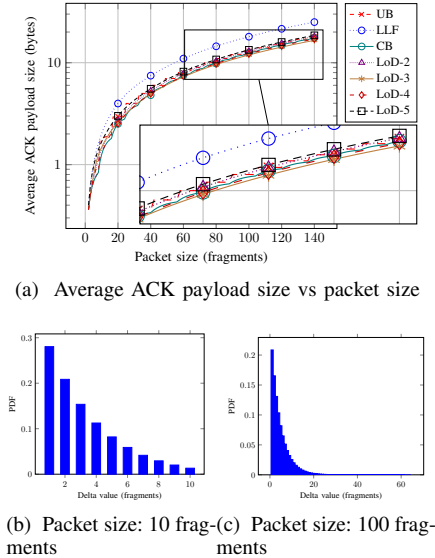


Fig. 12. Delta Probability Density Function (PDF), for $FER = 20\%$ and uniform losses.

2) *Burst losses*: This subsection analyzes the performance of the considered RFTs for the burst error distribution, for BOP of 1% and 2%, and $\lambda = 10$ fragments (i.e., FER of 10% and 20%, respectively).

Fig. 13a depicts the average ACK payload size for $BOP = 1\%$ and $\lambda = 10$ fragments. For small packet sizes (i.e., between 1 and 30 fragments), CB and LoD-2 produce the smallest ACK payload sizes, offering similar performance. For burst errors, CB improves the performance of UB to a greater extent than for uniform errors and the same FER (Fig. 11a). In a burst error distribution, it is more likely that the last loss will occur earlier in the packet transmission, thus allowing the compression advantages of CB to a greater extent. On the other hand, in a burst error distribution, there is a high probability of a delta value being equal to 1 (e.g., 0.83 for 10-fragment packets and 0.90 for 100-fragment packets, see Figs. 13b

and 13c, respectively), with other deltas corresponding to the distance between bursts. Since LoD-2 encodes the delta value of 1 with the lowest encoding overhead, LoD-2 offers good performance, outperforming the other RFTs for packet sizes greater than 40 fragments.

LoD-3 produces a smaller average ACK payload size than LoD-4. This is because LoD-3 requires less bits to encode the deltas (which are often equal to 1). For the same reason, LoD-5 performs worse than LoD-4. As expected, LLF exhibits a linear behavior with packet size, and generates, for packet sizes up to ~ 120 fragments, the largest ACK payload size among all the evaluated RFTs. We have also evaluated the ACK payload size for $BOP = 2\%$ and $\lambda = 10$ fragments. The relative behavior of the different RFTs is qualitatively similar to the one obtained for $BOP = 1\%$, albeit for greater ACK payload size. Therefore, the results for $BOP = 2\%$ are not shown for the sake of brevity.

B. Average number of L2 frames and ToA gain

In this section, we evaluate two important performance metrics derived from the ACK payload size results: the average number of L2 frames needed to carry an ACK (denoted AN_{L2F}) and the ToA, for all packet sizes, L2 MTU values (i.e., 11 bytes, 51 bytes, and 242 bytes), and fragment loss scenarios considered in Section VI-A. For a given RFT, we represent the ToA in relative terms as the ToA Gain (ToA_{Gain}), which is obtained by dividing the UB ToA by the ToA of the considered RFT. ToA_{Gain} evaluates the benefits, if any, that can be obtained from using an RFT different from UB. ToA is calculated as described in Appendix A.

1) *Uniform losses*: Fig. 14 shows the AN_{L2F} and the ToA_{Gain} results for $FER = 1\%$ and uniform losses, for the L2 MTU values considered. For a packet size smaller than 80 fragments, and for an 11-byte L2 MTU, the AN_{L2F} is always one, as the ACK payload in this range fits the maximum payload size of one L2 frame (see Fig. 10a). For L2 MTU values of 51 and 242 bytes, the AN_{L2F} is always equal to 1.

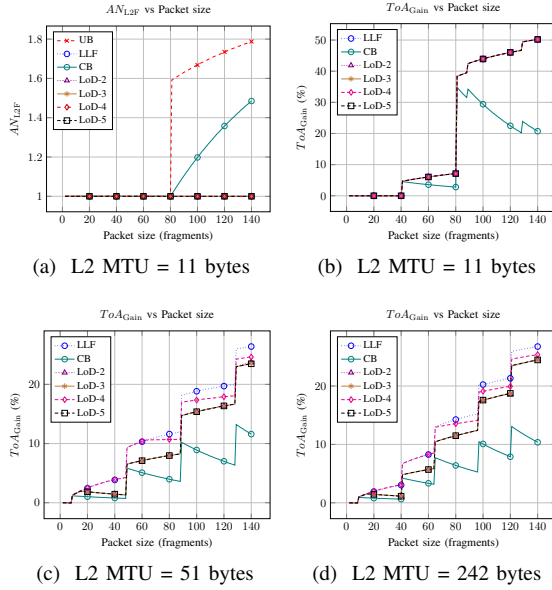


Fig. 14. Average number of L2 frames (a) and ToA_{Gain} (b,c,d) vs packet size for $FER = 1\%$ and uniform losses. In (b) results for LLF, and all LoD variants considered are overlapped.

On the other hand, the ToA_{Gain} is negligible for all considered RFTs for packet sizes up to 40 fragments. This is due to the L2 header overhead, which is much greater than the ACK payload size differences for the considered RFTs, for short packet sizes. As the L2 MTU increases, and for the same range of packet sizes, the L2 overhead has lower impact, and LoD-2 and LoD-3 present a ToA_{Gain} up to 4.8%. For greater packet sizes, all RFTs other than UB achieve a significant ToA improvement, for all L2 MTUs analyzed, since the differences in ACK payload size become more significant. The stepwise behavior of the ToA_{Gain} is due to the relation between the LoRaWAN frame size and its payload size (see further details in Appendix A.2).

Fig. 15 shows the AN_{L2F} and the ToA_{Gain} for $FER = 10\%$ and uniform losses for the considered L2 MTU values. For an 11-byte L2 MTU, the AN_{L2F} is also equal to one for packet sizes up to 40 fragments for LLF, and up to 80 fragments for all other RFTs. Since LLF depends strongly on the number of losses occurred during a packet transmission, its ACK payload size varies significantly, sometimes requiring two L2 frames to carry an ACK, even if its average ACK payload size is smaller than those of UB and CB (see Fig. 11a). For L2 MTU values of 51 and 242 bytes, the AN_{L2F} is always equal to 1. As a result, LLF exhibits even negative ToA_{Gain} for some values within the considered packet size range. For packet sizes greater than 80 fragments, UB and CB require two L2 frames to carry an ACK more often than the rest of RFTs, and the LoD variants offer the best performance. The frequent additional L2 frame penalizes UB and CB, introducing a significant ToA_{Gain} of up to $\sim 45\%$ for the rest of RFTs. This ToA_{Gain} increase decreases with packet size, with LoD variants, i.e., LoD-3, LoD-4 and LoD-5, achieving a similar ToA_{Gain} , with values up to 27%. As the L2 MTU value increases, and for packet sizes between 2 and 8 fragments, UB

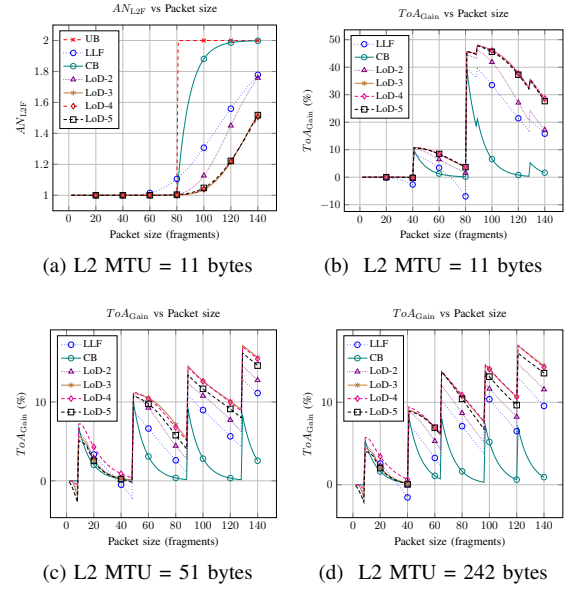


Fig. 15. Average number of L2 frames (a) and ToA_{Gain} (b,c,d) vs packet size for $FER = 10\%$ and uniform losses.

and CB offer the best performance, since other RFTs yield a negative ToA_{Gain} . As packet size increases, LoD-3 and LoD-4 become optimal, with ToA_{Gain} of up to 16%. On the other hand, the ToA_{Gain} of CB is up to only 9%, since this RFT can provide a relatively low bitmap compression degree due to fragment losses occurring at the end of packet transmission with relatively high probability.

Fig. 16 illustrates the AN_{L2F} and the ToA_{Gain} for $FER = 20\%$ and uniform losses, for the considered L2 MTUs. For a small L2 MTU (i.e., 11 bytes), results reflect how, for high FER , UB and CB generally offer good performance, compared with the rest of RFTs considered. LLF yields the largest AN_{L2F} and a negative ToA_{Gain} (down to -83% for a packet size of 80 fragments). Such AN_{L2F} increase happens because, in some cases, the ACK payload requires two L2 frames to be carried. As L2 MTU increases, LoD-3 and LoD-4 become the optimal RFTs, with a ToA_{Gain} of up to 9%. A seesaw ToA_{Gain} pattern arises because of the ToA stepwise behavior of UB, which makes UB yield better values than LoD-3, LoD-4, and the same ToA values as CB, for a short range of packets. For L2 MTU values of 51 and 242 bytes, the AN_{L2F} is always equal to 1.

2) *Burst losses*: Fig. 17 depicts the AN_{L2F} and ToA_{Gain} for $BOP = 1\%$ and $\lambda = 10$ fragments for 11-byte, 51-byte and 242-byte L2 MTUs, respectively. For a large range of packet sizes, LLF exhibits the largest AN_{L2F} for the 11-byte L2 MTU, and the smallest ToA_{Gain} for all L2 MTUs analyzed, due to its large ACK payload. Since fragment losses concentrate in bursts, LoD-2 benefits from its small overhead when encoding the highly frequent delta value of 1, and offers the best overall performance in terms of AN_{L2F} for the 11-byte L2 MTU, and ToA_{Gain} for all L2 MTUs analyzed. LoD-2 is followed closely by LoD-3 and LoD-4, as these LoD techniques have a larger overhead when encoding small deltas (see Fig. 13c). As the L2 MTU and packet size increases,

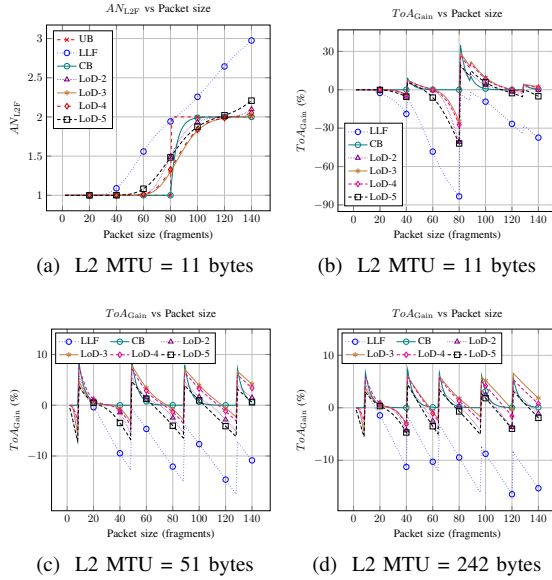


Fig. 16. Average number of L2 frames (a) and ToA_{Gain} (b,c,d) vs packet size for $FER = 20\%$ and uniform losses.

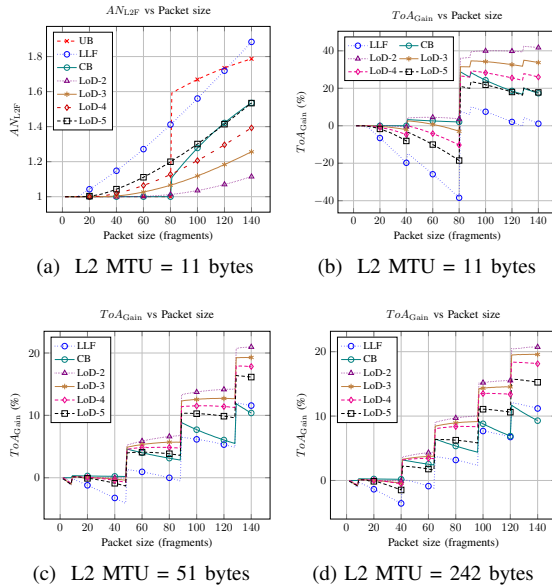


Fig. 17. Average number of L2 frames (a) and ToA_{Gain} (b,c,d) vs packet size for $BOP = 1\%$ and $\lambda = 10$.

UB is outperformed by all other RFTs considered (except LLF). The AN_{L2F} and ToA_{Gain} results for $BOP = 2\%$ are qualitatively similar to those for $BOP = 1\%$, with LoD-2 being the optimal RFT for a large range of packet sizes and for all L2 MTUs considered.

VII. CONCLUSION

In this paper we have presented a thorough evaluation of RFTs for reliable fragmentation over LPWAN. We considered CB (which is the RFT used in SCHC), UB (as a benchmark), and two alternative RFTs: LLF and LoD (the latter, with 4 different encoding variants). We developed the Sim-RFT

simulator to perform the evaluation. LoRaWAN was assumed as the underlying LPWAN technology.

Our results show that the optimal RFT depends on the channel conditions (in terms of error rate and error distribution), on L2 MTU, and on the size of the packet to be fragmented and carried. CB is not optimal in all the scenarios evaluated. CB tends to offer the best performance for small packets and high error rates. In such conditions, its encoding based on a bitmap (which is further optimized compared with UB) is efficient. As packet size increases, regardless of the L2 MTU, LoD variants tend to become optimal. Regarding the latter, for uniform errors, 3-bit and 4-bit bases offer the best trade-off. For burst errors, LoD-2 offers the best performance, as it minimizes the encoding overhead for the very frequent delta of 1. Finally, for high quality links with very low error rate, and uniformly distributed errors, LLF provides the most efficient encoding.

Using the optimal RFT for a given scenario allows to achieve performance benefits such as a higher downlink network capacity (which is especially critical when duty-cycle regulations are in force), greater network scalability, and lower IoT device energy consumption.

APPENDIX A LORAWAN OPERATION AND SETTINGS

In this section, we describe the main characteristics and settings of LoRaWAN, emphasizing the ones that are most relevant to the evaluation carried out in this paper. The section is divided into two parts, which focus on the LoRaWAN physical layer, and the LoRaWAN L2 layer, respectively.

1) *LoRaWAN physical layer*: LoRaWAN is based on LoRa as the physical layer. LoRa is a spread spectrum modulation scheme based on chirp spread spectrum (CSS) technology [59].

In LoRa networks, the time duration of a frame transmission at a given Spreading Factor (SF) and channel bandwidth (BW) is called the ToA of a frame (ToA_{frame}). The available BW is of 125 kHz, 250 kHz or 500 kHz. The SF takes values from 7 to 12. The ToA_{frame} can be defined as the time required for the transmission of the preamble plus the payload of the LoRa frame [60], and can be obtained as follows:

$$ToA_{frame} = T_{preamble} + T_{payload}. \quad (1)$$

The preamble is a sequence of a programmable number of symbols for receiver synchronization. Its transmission time can be calculated as follows:

$$T_{preamble} = (n_{preamble} + 4.25) \cdot T_{sym}, \quad (2)$$

where $n_{preamble}$ is the aforementioned number of symbols. The symbol period (T_{sym}) depends on the channel BW and SF selected, and can be calculated as follows:

$$T_{sym} = \frac{2^{SF}}{BW}. \quad (3)$$

The transmission time of the payload can be calculated as:

$$T_{payload} = (payloadSymbNb) \cdot T_{sym}, \quad (4)$$

TABLE III
LoRa AND LoRaWAN CONFIGURATION PARAMETERS

LoRa Configuration Parameters			
Region/Country	US	EU	CN
Channel Plan	US902-928	EU863-870	CN779-787
Data Rate (DR)	0	0	5
Spreading Factor (SF)	10	12	7
Bandwidth (kHz)	125	125	125
Indicative physical bit rate (bit/sec)	980	250	5470
$n_{preamble}$		8 symbols	
Header enable (H)		Disable (0)	
Low Data Rate Optimization (DE)		Disable (0)	
Coding Rate (CR)		4/5	
CRC present		YES (1)	
LoRaWAN Configuration Parameters			
L2 control headers		13 bytes	
L2 MTU (N)	11 bytes	51 bytes	242 bytes

where $payloadSymbNb$ is the number of symbols of the LoRa frame payload and header [60]. The $payloadSymbNb$ can be calculated as follows:

$$payloadSymbNb = 8 +$$

$$\max\left(\text{ceil}\left(\frac{8PL - 4SF + 28 + 16 - 20H}{4(SF - 2DE)}\right)(CR + 4), 0\right), \quad (5)$$

where PL is the payload size in bytes, and other parameters (SF, H, DE and CR) can be found in Table III.

Table III presents the LoRa and LoRaWAN configuration parameters used in the evaluation presented in Section VI.

2) *LoRaWAN L2 layer*: LoRaWAN specification [1] defines a LoRaWAN L2 frame format. The LoRaWAN L2 frame is carried by the LoRa frame. The L2 LoRaWAN frame has a 13-byte control header. Table III presents the LoRaWAN configuration parameters used in the evaluation presented in Section VI.

To better understand the relation between the LoRaWAN L2 payload size and the ToA_{frame} , Fig. 18 shows the ToA_{frame} for the configurations of LoRa and LoRaWAN presented in Table III, for different LoRaWAN L2 payload sizes. Note that ToA_{frame} shows a stepwise relationship with the LoRaWAN payload. This happens because of how LoRa physical layer determines the $payloadSymbNb$ (see (5)).

3) *LoRaWAN Device Class*: LoRaWAN defines three classes of devices (Class A, Class B, and Class C), which are relevant to the energy consumption of a device and communication delay. However, LoRaWAN device class is not relevant for this paper, since it is orthogonal to communication error characteristics.

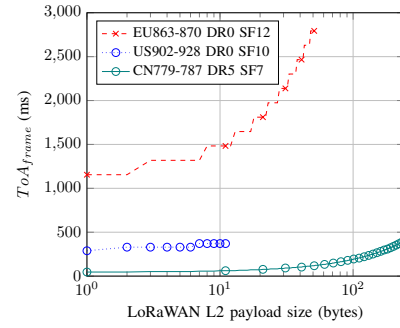


Fig. 18. ToA_{frame} for different LoRaWAN L2 payload sizes obtained by using (1) and the parameters shown in Table III.

ACKNOWLEDGMENT

This research is funded in part by the ERDF and the Spanish Government through project TEC2016-79988-P and project PID2019-106808RA-I00, AEI/FEDER, EU.

REFERENCES

- [1] LoRa Alliance Technical Committee, "LoRaWAN 1.1 Specification," https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_v1.1.pdf, Oct 2017.
- [2] A. Lavric, A. I. Petriariu, and V. Popa, "Long range SigFox communication protocol scalability analysis under large-scale, high-density conditions," *IEEE Access*, vol. 7, pp. 35 816–35 825, 2019.
- [3] C. Gomez, J. C. Veras, R. Vidal, L. Casals, and J. Paradells, "A Sigfox energy consumption model," *Sensors*, vol. 19, no. 3, 2019.
- [4] S. Farrell, "Low-Power Wide Area Network (LPWAN) Overview," RFC 8376, May 2018. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc8376.txt>
- [5] A. Ikpehai, B. Adebisi, K. M. Rabie, K. Anoh, R. E. Ande, M. Ham-moudeh, H. Gacanan, and U. M. Mbanaso, "Low-power wide area network technologies for internet-of-things: A comparative review," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2225–2240, April 2019.
- [6] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "Overview of cellular LPWAN technologies for IoT deployment: Sigfox, LoRaWAN, and NB-IoT," in *Proc. of IEEE PerCom Workshops*, Athens, Greece, 2018.
- [7] H. Mroue, A. Nasser, S. Hamrioui, B. Parrein, E. Motta-Cruz, and G. Rouyer, "MAC layer-based evaluation of IoT technologies: LoRa, SigFox and NB-IoT," in *Proc. of IEEE MENACOMM*, Jounieh, Lebanon, 2018.
- [8] L. Casals, B. Mir, R. Vidal, and C. Gomez, "Modeling the energy performance of LoRaWAN," *Sensors*, vol. 17, no. 10, 2017.
- [9] LPWAN Working Group. IPv6 over Low Power Wide-Area Networks (lpwan). [Online]. Available: <https://datatracker.ietf.org/wg/lpwan/about/>
- [10] A. Minaburo, L. Toutain, C. Gomez, D. Barthel, and J.-C. Zúñiga, "SCHC: Generic Framework for Static Context Header Compression and Fragmentation," RFC 8724, April 2020. [Online]. Available: <https://rfc-editor.org/rfc/rfc8724.txt>
- [11] C. Gomez, A. Minaburo, L. Toutain, D. Barthel, and J. C. Zuniga, "IPv6 over LPWANs: connecting low power wide area networks to the internet (of things)," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 206–213, 2020.
- [12] W. Eddy and E. Davies, "Using Self-Delimiting Numeric Values in Protocols," Internet Requests for Comments, RFC Editor, RFC 6256, May 2011. [Online]. Available: <https://www.rfc-editor.org/info/rfc6256>
- [13] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "LSCHC: Layered Static Context Header Compression for LPWANs," in *Proc. of CHANTS, Session: Security & IoT*, New York, USA, 2017.
- [14] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "Dynamic context for static context header compression in LPWANs," in *Proc. of 14th DCOSS*, New York, USA, 2018.
- [15] W. Ayoub, F. Nouvel, S. Hmede, A. E. Samhat, M. Mroue, and J.-C. Prévotet, "Implementation of SCHC in NS-3 simulator and comparison with 6LoWPAN," in *Proc. of 26th ICT*, Hanoi, Vietnam, 2019.

- [16] B. Moons, A. Karaagac, J. Haxhibeqiri, E. De Poorter, and J. Hoebeke, "Using SCHC for an optimized protocol stack in multimodal LPWAN solutions," in *Proc. of IEEE WF-IoT*, Limerick, Ireland, 2019.
- [17] S. Aguilar, A. Marquet, L. Toutain, C. Gomez, R. Vidal, N. Montavont, and G. Z. Papadopoulos, "LoRaWAN SCHC fragmentation demystified," in *Ad-Hoc, Mobile, and Wireless Networks*, M. R. Palattella, S. Scanzio, and S. Coleri Ergen, Eds. Springer International Publishing (Cham), 2019, pp. 213–227.
- [18] S. Aguilar, P. Maillé, L. Toutain, C. Gomez, R. Vidal, N. Montavont, and G. Z. Papadopoulos, "Performance analysis and optimal tuning of IETF LPWAN SCHC ACK-on-Error Mode," *IEEE Sensors Journal*, vol. PP, 07 2020.
- [19] J. Sanchez-Gomez, J. Gallego-Madrid, R. Sanchez-Iborra, J. Santa, and A. F. Skarmeta, "Impact of SCHC compression and fragmentation in LPWAN: A case study with LoRaWAN," *Sensors*, vol. 20, no. 1, 2020.
- [20] I. Suciú, X. Vilajosana, and F. Adelantado, "An analysis of packet fragmentation impact in LPWAN," in *Proc. of IEEE WCNC*, Barcelona, Spain, 2018.
- [21] I. Suciú, X. Vilajosana, and F. Adelantado, "Aggressive fragmentation strategy for enhanced network performance in dense LPWANs," in *Proc. of IEEE PIMRC*, Bologna, Italy, 2018.
- [22] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC 8200, July 2017. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc8200.txt>
- [23] Sim-RFT. [Online]. Available: <https://www.github.com/saguilarlevel/sim-RFT>
- [24] Sigfox, "Sigfox connected objects: Radio specifications," <https://build.sigfox.com/sigfox-device-radio-specifications>, 2 2019, rev. 1.3.
- [25] O. Gimenez and I. Petrov, "Static Context Header Compression and Fragmentation (SCHC) over LoRaWAN," Internet Requests for Comments, RFC Editor, RFC 9011, April 2021.
- [26] A. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, "Does bidirectional traffic do more harm than good in LoRaWAN based LPWA networks?" *CoRR*, vol. abs/1704.04174, 2017. [Online]. Available: <http://arxiv.org/abs/1704.04174>
- [27] M. Capuzzo, D. Magrin, and A. Zanella, "Confirmed traffic in LoRaWAN: Pitfalls and countermeasures," in *Proc. of 17th Med-Hoc-Net*, 2018, pp. 1–7.
- [28] J. Petäjäjärvi, K. Mikhaylov, M. Hämäläinen, and J. Iinatti, "Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring," in *Proc. of 10th ISMICT*, Massachusetts, USA, 2016, pp. 1–5.
- [29] Best practices. [Online]. Available: <https://www.thethingsindustries.com/docs/devices/best-practices/>
- [30] J. Haxhibeqiri, A. Karaagac, F. Van den Abeele, W. Joseph, I. Moerman, and J. Hoebeke, "LoRa indoor coverage and performance in an industrial environment: Case study," in *Proc. of ETFA*, Limassol, Cyprus, 2017, pp. 1–8.
- [31] T. Petrić, M. Goessens, L. Nuaymi, L. Toutain, and A. Pelov, "Measurements, performance and analysis of LoRa FABIAN, a real-world implementation of LPWAN," in *Proc. of 27th IEEE PIMRC2016*, Valencia, Spain, 2016, pp. 1–7.
- [32] G. Callebaut and L. Van der Perre, "Characterization of LoRa point-to-point path loss: Measurement campaigns and modeling considering censored data," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1910–1918, 2020.
- [33] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of LoRa: Long range & low power networks for the internet of things," *Sensors*, vol. 16, no. 9, 2016.
- [34] P. J. Marcelis, V. Rao, and R. V. Prasad, "DaRe: Data Recovery through application layer coding for LoRaWAN," in *Proc. of 2nd IoTDI*, New York, USA, 2017.
- [35] U. Coutaud, M. Heusse, and B. Tourancheau, "Adaptive data rate for multiple gateways LoRaWAN networks," in *Proc. of 16th WiMob*, Virtual Conference, 2020, pp. 1–6.
- [36] E. Ruano Lin, "LoRa protocol. evaluations, limitations and practical test," Master's thesis, Universitat Politècnica de Catalunya, <https://upcommons.upc.edu/handle/2117/98853>, 5 2016.
- [37] U. Coutaud, M. Heusse, and B. Tourancheau, "High reliability in LoRaWAN," in *Proc. of 31st IEEE PIMR2020*, Virtual Conference, 2020, pp. 1–7.
- [38] R. Sanchez-Iborra, I. G. Liaño, C. Simoes, E. Couñago, and A. F. Skarmeta, "Tracking and monitoring system based on LoRa technology for lightweight boats," *Electronics*, vol. 8, no. 1, 2019. [Online]. Available: <https://www.mdpi.com/2079-9292/8/1/15>
- [39] S.-Y. Wang, J.-E. Chang, H. Fan, and Y.-H. Sun, "Performance comparisons of NB-IoT, LTE Cat-M1, Sigfox, and LoRa moving at high speeds in the air," in *Proc. of 25th IEEE ISCC*, Rennes, Francia, 2020, pp. 1–6.
- [40] E. Pramananto, M. Ulfa, and A. Kurniawan, "Coast panic-emergency situation monitoring system on west and east sailing lane of surabaya using LORAWAN technology," in *Proc. of CENIM*, Surabaya, Indonesia, 2018, pp. 100–105.
- [41] J. Petäjäjärvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, "On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology," in *Proc. of 14th ITST*, Copenhagen, Denmark, 2015, pp. 55–59.
- [42] J. Herrera-Tapia, E. Hernández-Orallo, A. Tomás, C. T. Calafate, J.-C. Cano, M. Zennaro, and P. Manzoni, "Evaluating the use of sub-gigahertz wireless technologies to improve message delivery in opportunistic networks," in *Proc. of 14th IEEE ICNSC*, Calabria, Italy, 2017, pp. 305–310.
- [43] M. Sandra, S. Gunnarsson, and A. J. Johansson, "Internet of buoys: An internet of things implementation at sea," in *Proc. of 54th ACSSC*, California, USA, 2020, pp. 1096–1100.
- [44] C. A. Trasviña-Moreno, R. Blasco, A. Marco, R. Casas, and A. Trasviña-Castro, "Unmanned aerial vehicle based wireless sensor network for marine-coastal environment monitoring," *Sensors*, vol. 17, no. 3, 2017.
- [45] L. Sciuillo, F. Fossemo, A. Trotta, and M. Di Felice, "LOCATE: A LoRa-based mOBile emergenCy mANagement sYsTEm," in *Proc. of IEEE GLOBECOM*, Abu Dhabi, United Arab Emirates, 2018, pp. 1–7.
- [46] O. A. Saraereh, A. Alsaraira, I. Khan, and P. Uthansakul, "Performance evaluation of UAV-Enabled LoRa networks for disaster management applications," *Sensors*, vol. 20, no. 8, 2020.
- [47] J. J. Kang and S. Adibi, "Bushfire disaster monitoring system using low power wide area networks (LPWAN)," *Technologies*, vol. 5, no. 4, 2017.
- [48] V. Gupta, S. K. Devar, N. H. Kumar, and K. P. Bagadi, "Modelling of IoT traffic and its impact on LoRaWAN," in *Proc. of IEEE GLOBECOM*, Singapore, 2017, pp. 1–6.
- [49] R. Casas, A. Hermosa, A. Marco, T. Blanco, and F. J. Zarazaga-Soria, "Real-time extensive livestock monitoring using LPWAN smart wearable and infrastructure," *Applied Sciences*, vol. 11, no. 3, 2021.
- [50] D. Patel and M. Won, "Experimental study on low power wide area networks (LPWAN) for mobile internet of things," in *Proc. of 85th IEEE VTC Spring*, Sydney, Australia, 2017, pp. 1–5.
- [51] U. Coutaud and B. Tourancheau, "Channel coding for better QoS in LoRa networks," in *Proc. of 14th WiMob*, Limassol, Cyprus, 2018, pp. 1–9.
- [52] T. Ameloot, P. Van Torre, and H. Rogier, "Periodic LoRa signal fluctuations in urban and suburban environments," in *Proc. of 13th EuCAP*, Krakow, Poland, 2019, pp. 1–5.
- [53] J. Haxhibeqiri, F. Van den Abeele, I. Moerman, and J. Hoebeke, "LoRa scalability: A simulation model based on interference measurements," *Sensors*, vol. 17, no. 6, 2017.
- [54] E. D. Ayele, C. Hakkenberg, J. P. Meijers, K. Zhang, N. Meratnia, and P. J. M. Havinga, "Performance analysis of LoRa radio for an indoor iot applications," in *Proc. of IoTGC*, Funchal, Madeira Island, 2017, pp. 1–8.
- [55] R. Sanchez-Iborra, J. Sanchez-Gomez, J. Ballesta-Viñas, M.-D. Cano, and A. F. Skarmeta, "Performance evaluation of LoRa considering scenario conditions," *Sensors*, vol. 18, no. 3, 2018.
- [56] Semtech. Predicting LoRaWAN Capacity. [Online]. Available: <https://loro-developers.semtech.com/library/tech-papers-and-guides/predicting-lorawan-capacity/>
- [57] T. Attia, M. Heusse, B. Tourancheau, and A. Duda, "Experimental characterization of LoRaWAN link quality," in *Proc. of IEEE GLOBECOM*, Hawaii, USA, 2019, pp. 1–6.
- [58] P. Thubert, "IPv6 over low-power wireless personal area network (6LoWPAN) selective fragment recovery," Internet Requests for Comments, RFC Editor, RFC 8931, November 2020.
- [59] What is LoRa? [Online]. Available: <https://www.semtech.com/loro/what-is-lora>
- [60] Semtech, "SX1272/3/6/7/8: LoRa Modem Designer's Guide AN1200.13," Jul. 2013. [Online]. Available: https://www.semtech.com/uploads/documents/LoraDesignGuide_STD.pdf



Sergio Aguilar received the B.S. degree in electrical engineering and a degree in communication systems from the Universidad de Costa Rica (UCR) in 2006 and 2008, respectively. He received the M.Sc. degree in telecommunication engineering and management from the Universitat Politècnica de Catalunya (UPC) in 2016, where he is currently pursuing the Ph.D. degree in network engineering. His interests include the Internet of Things, LPWAN technologies and the IETF SCHC framework.



Carles Gomez received his Ph.D. degree from Universitat Politècnica de Catalunya (UPC) in 2007. He is an associate professor at the same university. He is a co-author of numerous technical contributions including papers published in journals and conferences, IETF RFCs, and books. His current research interests focus mainly on the Internet of Things. He serves as an editorial board member of several journals. He is also an IETF 6Lo working group chair.



Rafael Vidal received his M.Sc. and Ph.D. degrees from the Universitat Politècnica de Catalunya (UPC) in 1997 and 2009, respectively. He is an associate professor at the same university. He has worked in several publicly funded research projects, and is co-author of several papers published in journals and conferences. His current research interests include performance of wireless networks (in particular, low-power wide-area networks) and the Internet of Things.