DaRe: Data Recovery through Application Layer Coding for LoRaWAN

P.J. Marcelis
Embedded Software
Delft University of Technology
The Netherlands
pjmarcelis@gmail.com

V. Rao
Embedded Software
Delft University of Technology
The Netherlands
v.rao@tudelft.nl

R.V. Prasad
Embedded Software
Delft University of Technology
The Netherlands
r.r.venkateshaprasad@tudelft.nl

ABSTRACT

Internet of Things (IoT) solutions are increasingly being deployed for smart applications. To provide good communication for the increasing number of smart applications, there is a need for low cost and long range Low Power Wide Area Network (LPWAN) technologies. LoRaWAN is an energy efficient and inexpensive LPWAN solution that is rapidly being adopted all around the world. However, LoRaWAN does not guarantee reliable communication in its basic configuration. Transmitted frames can be lost due to the channel effects and mobility of the end-devices. In this study, we perform extensive measurements on a new LoRaWAN network to characterise spatial and temporal properties of the LoRaWAN channel. The empirical outage probability for the farthest measured distance from the closest gateway of 7.5 km in our deployment is as low as 0.004, but the frame loss measured at this distance was up to 70%. Furthermore, we show that burstiness in frame loss can be expected for both mobile and stationary scenarios. Frame loss results in data loss, since in the basic configuration frames are only transmitted once. To reduce data loss in LoRaWAN, we design a novel coding scheme for data recovery called DaRe, which extends frames with redundant information that is calculated from the data from previous frames. DaRe combines techniques from convolutional codes and fountain codes. We develop an implementation for DaRe and show that 99% of the data can be recovered with a code rate of 1/2 for up to 40% frame loss. Compared to repetition coding DaRe provides 21% more data recovery, and can save up to 42% energy consumption on transmission for 10 byte data units. DaRe also provides better resilience to bursty frame loss. This study provides useful results to both LoRaWAN network operators as well as developers of LoRaWAN applications. Network operators can use the characterisation results to identify possible weaknesses in the network, and application developers are offered a tool to prevent possible data loss.

CCS CONCEPTS

•Networks →Network performance analysis; •Theory of computation →Error-correcting codes; •Software and its engineering →*Designing software;*

IoTDI 2017, Pittsburgh, PA USA

DOI: http://dx.doi.org/10.1145/3054977.3054978

KEYWORDS

LoRa, LoRaWAN, LPWAN, network measurements, burstiness, forward error correction, DaRe, data recovery, erasure coding, channel coding, application layer coding, fountain code, convolutional code

ACM Reference format:
P.J. Marcelis, V. Rao, and R.V. Prasad. 2017. DaRe: Data Recovery through Application Layer Coding for LoRaWAN. In *Proceedings of The 2nd ACM/IEEE International Conference on Internet-of-Things Design and Implementation, Pittsburgh, PA USA, April 2017 (IoTDI 2017),* 12 pages.
DOI: http://dx.doi.org/10.1145/3054977.3054978

1 INTRODUCTION

With miniaturisation of computing and communication devices, the number of sensors used in our daily life is increasing exponentially in the recent years. With the rise and projected scale of applications in IoT, many technological solutions are proposed for various applications such as smart industry, smart street lighting, smart cities, and vehicle to infrastructure communications for vehicle monitoring and traffic updates [3]. Many early adopters of IoT used short range communication for their sensors requiring multiple hops before reaching the Internet cloud. However, recent advances have enabled a simpler technological solution called Low Power Wide Area Networks (LPWAN) that offer low energy communication over distances of multiple kilometers, making sensors send data to the Internet backbone over only one hop.

LoRaWAN (Long Range Wide Area Network) is one of the several competing LPWAN technologies with amongst others SigFox, NB-IoT and Weightless [5]. LoRaWAN has been the most successful of these technologies in providing an easily accessible LPWAN [6]. This protocol is being developed by the LoRa Alliance [24], which is an open alliance. LoRaWAN is specifically being developed for applications with small end-devices that have to send small amounts of data over large time intervals. In comparison with other LPWAN technologies, LoRaWAN claims an inexpensive, secure and power efficient communication method for these devices. Therefore, it is important to study LoRaWAN and characterise it in a practical deployment.

LoRaWAN is based on LoRa, which describes the physical layer of the communication. LoRa uses a chirp spread spectrum modulation, i.e., the data is modulated on a carrier signal that constantly changes frequency in a fixed bandwidth. This modulation technique is said to give LoRa a good resilience to interference, multipath and Doppler effects [20]. A LoRaWAN frame can be received by multiple gateways that forward the frame to a central LoRaWAN network server. The network server checks device policy, decrypts

the frame payload, removes duplicates, and forwards the data to the corresponding application server.

Since LoRaWAN communication takes place over long distances, frames can be lost due to propagation losses and other physical phenomena such as shadow fading, reflection, and scattering due to urban clutter. The LoRa modulation scheme supports various spreading factors over the carrier wave, with each spreading factor offering a certain data rate and robustness against the physical phenomena. Spreading factor 12 (SF12) is the most robust of the six available spreading factors and trades off data rate for increased range and increased reception ratio at the gateways. In order to study and characterise frame loss, we perform real-world measurements with a LoRaWAN network. We have collected data extensively (around 23,000 frames) over several days in stationary and mobile scenarios. Since the network is still in development, the LoRaWAN services are not yet open for general public. This allows us to do first of its kind measurements of LoRaWAN in an *almost* collision free LoRaWAN network, something that will be hard to do in the future. With the collected data, we observe that there is a significant amount of frame loss that occurs as the end-device moves farther away from a gateway. We observe a loss of up to 53% when the end-device is around 6 km away from the nearest gateway. Furthermore, we observe that frame loss in the channel can be quite bursty.

Frame loss leads to loss in data. Since an IoT application is often data-driven data loss must be minimal, so it is desired to recover the data lost in LoRaWAN communication. There are several constraints that need to be met for LoRaWAN data recovery:

(1) LoRaWAN targets to serve up to 15,000 devices per gateway. Automatic Repeat Request (ARQ) with acknowledgement (ACK) mechanisms is an option in the LoRaWAN protocol. However this option is not default since it introduces a lot of communication overhead and requires downlink communication. With increasing numbers of devices in a LoRaWAN network, the transmission of ACKs by the half-duplex gateways will lead to frame loss in the uplink communication. Hence ACKs affect the scalability of the network as well as increase the energy consumption on the end-devices.
(2) The solution must not require any change in the gateways since such a solution is expensive for existing deployments.
(3) The LoRaWAN communication channel can introduce bursty frame loss, as we will show in Section 4.

Considering these constraints, we infer that any proposed technique should operate at the application layer such that the solution can work with already deployed LoRaWAN networks. Also scenarios with bursty frame loss should be handled.

In this work, we design and implement an application layer coding scheme called *DaRe* to reduce data loss. The need for such a scheme is further motivated by the ALOHA-type medium access technique employed in LoRaWAN, which will inevitably cause a lot of collisions leading to frame loss. DaRe is a convolutional erasure coding scheme with novel techniques employed to maximise data recovery and minimise overhead. DaRe does not intend to recover the lost frames but it enables the recovery of the data from lost frames using forward error correction on the application level.

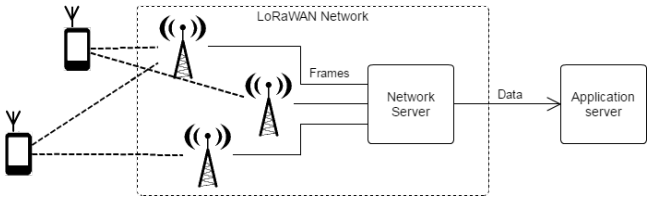


Figure 1: Schematic overview of a LoRaWAN network.

Since the coding method will be implemented on embedded devices with low computational capabilities, we design DaRe to be a low complexity mechanism.

Specifically, our contributions through this work are as follows:

(1) To the best of our knowledge, we are the first to perform large-scale measurements of LoRaWAN in an *almost* collision-free deployment of multiple gateways. We collect data sets for stationary and mobile scenarios over several days.
(2) With the data sets, we are the first to characterise both spatial (frame loss over distance) and temporal (burstiness of frame loss) properties of the channel. We find that the channel can be bursty even when the end-device is stationary.
(3) To recover data from lost frames, we propose a novel application layer coding technique called DaRe, based on both convolutional codes and fountain codes. We use an algebraic framework to describe and simulate the coding technique.
(4) We develop an implementation of DaRe for LoRaWAN. We evaluate this implementation by (i) emulating results for theoretical channels, (ii) emulating DaRe on the data sets collected in the network measurements, and (iii) performing frame and data loss measurements with a device running DaRe in runtime. We find that we can recover up to 99% of the data up to 40% frame loss with a code rate of 1/2. With a code rate of 1/5, we can recover 99% up to 68% frame loss. Furthermore, we show that the solution has little overhead compared to other low-complexity solutions.

The rest of the paper is organised as follows. In Section 2 a brief introduction on LoRaWAN is given. The method of data collection is explained in Section 3. The frame loss characterisation of LoRaWAN is done in Section 4. In Section 5, DaRe and the algebraical framework is introduced. The performance of DaRe is assessed through simulations in Section 6. In Section 7, an implementation of DaRe is proposed and evaluated. In Section 8, existing research on LoRaWAN and erasure coding is discussed. We conclude in Section 9.

2 AN OVERVIEW OF LORAWAN

In this section, we briefly introduce LoRaWAN. A LoRaWAN network has a star of stars topology: the end-devices transmit data directly to one or more gateways, which in turn connect directly to a central network server. A schematic of this network topology is

shown in Figure 1. The network server removes duplicate messages and forwards them to an application server.

LoRaWAN operates in the license-free ISM band. The 915 MHz and 868 MHz bands are used in the USA and in Europe respectively. The band in Europe has certain regulations [25]. A maximum transmit power of 14 dBm is allowed on the end-devices, and the maximum allowed transmit duty cycle is 1%. This means devices can only transmit 1% of the time. This applies to the gateways as well.

Devices communicating over LoRaWAN have three operating classes - class A, B and C - with an increasing order of downlink communication (communication from a gateway to a device). Class A is the default class of operation. A device in this class adopts an ALOHA-like scheme to send a LoRaWAN frame. After a device in Class A has transmitted a frame, it will listen for two time slots in which the device can receive data. Only in these time slots is a Class A device in receiving mode. Class B extends Class A by having additional receiving slots at preset times. A Class C device will always be in receiving mode unless it is transmitting. Class A is the most preferred operating class since it has the least energy consumption.

A LoRaWAN frame is transmitted with a certain spreading factor. The spreading factor can range from SF7 to SF12 and expresses the number of chirps that are used per symbol. Each symbol is encoded in 2^{SF} chirps. A higher spreading factor has a longer transmission time and lower data rates but will give a longer range. LoRaWAN supports the automatic setting of the spreading factor with Adaptive Data Rate (ADR). The network server monitors the quality of the uplink communication and informs the device on the best spreading factor to use.

The LoRa physical layer employs a soft hamming block error correcting code with a code rate 4/5 to reduce bit errors. However, this does not eliminate frame loss completely. The reception of a LoRaWAN frame relies primarily on detecting the preamble of a frame. If the preamble is not detected by the receiver, the complete frame is not received. This makes the LoRaWAN communication channel a packet erasure channel. Frames are either completely received, or not received at all.

Error detection in a frame is done by means of a 32-bit message integrity code (MIC). This MIC also guarantees message authenticity, since it is a cipher-based message authentication code (CMAC) calculated with a device specific key.

3 SETUP AND DATA COLLECTION SCENARIOS

In this section, we describe our data collection setup, scenarios, method and the data sets collected for the analysis of the LoRaWAN communication channel.

3.1 Measurement Setup

3.1.1 Network. The LoRaWAN network used for this research was in development. Because of this there were only a handful of devices transmitting data on the network. Therefore, we consider frame loss as a result of collisions to be negligible. Furthermore, the configuration of the network is still being improved and the gateways are positioned only for (close to) line-of-sight coverage.

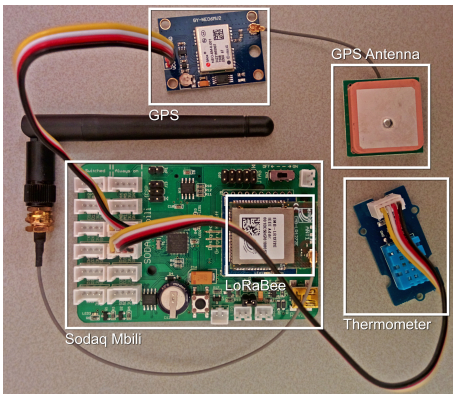


Figure 2: The data collection device. A Sodaq Mbili Rev. 4 [23] with EMB-LR1272E LoRaBee module, DHT11 thermometer and GY-NEO6MV2 GPS module.

For this reason we only collect data in outdoor scenarios, and frame loss is expected to be lower when the network is in production.

While the measurement devices adhered to the duty cycle limits, they were exempted from any usage limits on the number of messages. We use Thingpark Wireless Logger[1] to log the LoRaWAN frames received by the network server. Wireless Logger stores payload and metadata of each received LoRaWAN frame. As explained before, multiple gateways can receive a single frame. Unfortunately only up to three receiving gateways per frame are shown in the logger. That is, only the three gateways that received the LoRaWAN frame with the strongest received signal strength indicator (RSSI) can be identified in the logs. This complicates a per gateway analysis of frame reception. To eliminate false negatives (assuming a gateway did not receive a frame, while in reality it did), only the strongest three receiving gateways per data set are used to do per gateway reception analysis. But the small probability for false negatives makes the results in this paper tend towards the worst case of the actual frame loss characteristics.

All the gateways in the LoRaWAN network are situated at an average height of 27 m. The antennas have a gain of 11 dBi. The gateways are positioned on an average of 8 km apart. Furthermore, the maximum distance that an end-device can be positioned from its closest gateway is 7.5 km.

3.1.2 End-Device. The end-device used for data acquisition is a Sodaq Mbili Rev. 4 [23], shown in Figure 2. The antenna has a gain of 2 dBi. The device operates in Class A.

To characterise the LoRaWAN communication channel in time and space, measurements were done at multiple locations. To perform the analysis after data collection, we need the location from which each frame was transmitted. A first version of the device did not include a GPS module, so the location from which LoRaWAN frames were sent were matched using the timestamp to a GPS track made with the mobile application Strava[2]. The final version of the device did have the GPS module, so the coordinates were sent in the frame payload itself.

[1] http://www.thingpark.com/
[2] https://www.strava.com/

The LoRaBee module is a separate component and is connected to the Sodaq Mbili with a serial interface. The serial interface is used to transmit data, retrieve the received data, and to set MAC parameters. The sub-band and bandwidth are set according to Semtech standards [19]. The bandwidth is 125 kHz for all transmissions. Furthermore, we set the transmission power to the maximum allowed 14 dBm.

The spreading factor is controlled by the ADR mechanism, since a fixed spreading factor is not supported by the device. This means frames could be sent with any spreading factor. However, the largest part of the frames (~95%) were transmitted using SF12. Only these frames have been used for data analysis.

The spreading factor and the coordinates for frames that were not received needed to be guessed. The coordinates were interpolated on a straight line between the neighbouring received frames. The spreading factor for the missing frames was taken to be the lowest spreading factor of the neighbouring received frames.

3.2 Data Collection Scenarios

We identify two scenarios for our data collection: data from stationary locations and data when the end-device is moving (mobile data). All the data was collected with the end-devices placed next to a window or outdoors.

3.2.1 Stationary data. The stationary data sets were generated with the end-devices transmitting every 15 s, 10 minutes or 15 minutes. During measurements the devices were in a constant position and orientation. The stationary data contains roughly 18,000 frames.

3.2.2 Mobile data. The mobile data sets were collected using the device shown in Figure 2. During data collection the device was battery powered. Measures were taken to make sure that the battery was sufficiently charged and operational during the entire duration of data collection. For generating mobile data we chose two methods to make it mobile: by bicycle (bike) and by car.

A bike has been ridden with an average speed of 22 km/h for approximately 300 km in total. The farthest distance to the closest gateway is 7.5 km, and the average distance to the closest gateway is 3.2 km. The end-device was also taken in a car for approximately 350 km in total. The average speed was around 80 km/h (mainly on highways). In this scenario, approximately 80% of the LoRaWAN frames sent were from rural area, and the rest were from an urban area. The selected terrain was flat, without significant hills or mountains. The mobile data contains approximately 5,000 frames.

4 FRAME LOSS CHARACTERISATION

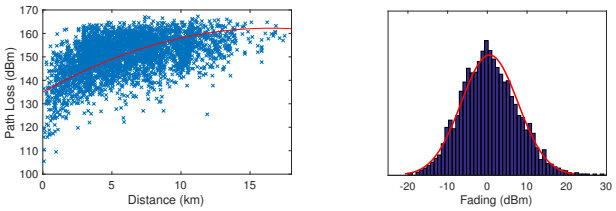
Before we present our coding scheme we first analyse the data sets in order to characterise the channel and the frame loss. We present our findings from the analysis in this section.

4.1 Channel Model and Outage

The channel model is a function of the distance between the sender (end-device) and the receiver (gateway). We use the bike data set to determine the model, since it provides measurement points at different distances. The Doppler effect is negligible because of the relative low speed. Since the bike data is collected from all different routes, we can negate the effect of permanent shadowing and other effects due to surroundings. We look at the measured received signal strength indicator (RSSI) and the signal-to-noise ratio (SNR) for different transmission distances to calculate path loss [16] given by,

$$PL = RSSI + SNR + P_{TX} + G_{TX} + G_{RX}. \quad (1)$$

Here, P_{TX} is the transmit power, G_{TX} and G_{RX} are the gains of the transmitter and the receiver antenna respectively. We then fit a line with a second order polynomial fit to estimate the expected path loss. This is shown in Figure 3(a). Based on the fit we estimate the path-loss exponent, which is found to be 2.71. The shadow fading, obtained by subtracting the path loss from the expected path loss, is shown in Figure 3(b). The fading follows the log-normal distribution with a mean of 0.56 and standard deviation of 7.11. With these parameters and the receiver sensitivity of the gateway (around -137 dBm), we can estimate the cell outage probability [8]. The outage probability is the probability that the received power at a given distance falls below the receiver sensitivity due to path loss and shadowing. Due to paucity of space, we refer the reader to [8] for details. The outage probability for the farthest distance of 7.5 km is found to be 0.004, indicating that the coverage of the current deployment is good.



(a) (b)

Figure 3: Channel Model. (a) Attenuation due to path loss is modelled using bike data set, giving a path-loss exponent of 2.71. (b) The fading follows log-normal distribution as shown by the distribution fit.

4.2 Frame Loss over Distance

In order to characterise frame loss, or *erasure* of frames, due to the channel effects, we analyse the data from all our data sets. We first look at frame loss as a function of distance between the end-device and the gateway. Since we can determine the location of each sent frame, we can calculate the distance from the end-device to the gateways and check whether the frame has been received. To account for location estimation inaccuracies we consider bins of 1.5 km in which we calculate the average frame reception ratio (FRR). Figure 4 shows the FRR with respect to the distance of the end-device from the gateway. It is evident from the figure that more frames are lost as the distance to the gateway increases. While the outage probability is quite low at 7.5 km, the frame loss is significant at that distance: almost 70% of the frame are lost around that distance.

Finding #1: Frame loss is quite significant in LoRaWAN despite an almost collision-free channel.

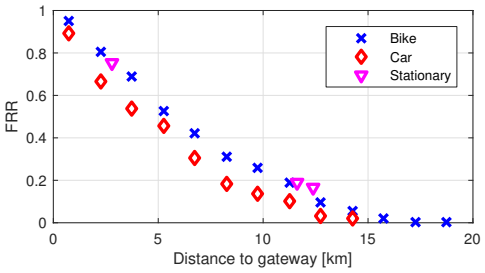


Figure 4: Frame Reception Ratio (FRR) as function of the distance to a gateway for bicycle, car and stationary data sets.

We can observe another interesting aspect in this figure, because the data sets are collected at different moving speeds. Due to Doppler shifts, it is expected that higher speeds will result in higher frame loss although LoRa is claimed to be quite robust to Doppler shifts [20]. However, the difference in frame loss between data sets for a given distance is not highly significant, especially between the stationary and the bike data sets. The car data set shows more loss, with a maximum difference of around 17%. From this, we can conclude that the bike data set gives a lower bound on the expected frame reception ratio for a stationary end-device at a certain distance.

4.3 Burstiness

We analyse the frame loss pattern over time in LoRaWAN. While some data sets exhibited uniform random erasures of frames, other data sets showed more consecutive erasures than to the expectation for an IID erasure probability. This prompted us to look at the burstiness of the erasures. Burstiness is a temporal property of the channel. The channel shifts between poor and good states with a correlation between frame delivery events. This results in erasures being closer to each other for a bursty data stream compared to a non-bursty data stream. While burstiness in wireless links has been shown extensively, a metric to express burstiness is not standardised. We take a simple and effective measure of using the *Hurst exponent* to quantify the burstiness [9].

In time-series analysis, the Hurst exponent is used as a measure to indicate the long-term "memory" or the correlation between events. In our case, the Hurst exponent, H, is used to measure if the same set of events occur successively over a long-term. This parameter, H, is also called the self-similarity parameter. A data stream is said to be self-similar if it looks roughly the same on any scale. Data streams with burstiness will show lower self-similarity than data streams without burstiness.

In order to estimate H, we consider the variance-time plot [9]. That is, a plot of the variance of the sample mean $\bar{X}$ of the data stream for sub-series of length l. This can be expressed as,

$$\mathrm{Var}(\bar{X}_l) \rightarrow \alpha n^{2H-2}, \ \forall \alpha > 0. \tag{2}$$

For a data stream of length m, the mean of sub-series j of length l is given by,

$$\bar{X}_l(j) = \frac{1}{l} \sum_{i=jl+1}^{(j+1)l} X_i, \ j \in \mathbb{N}_{[0,m/l]}. \tag{3}$$

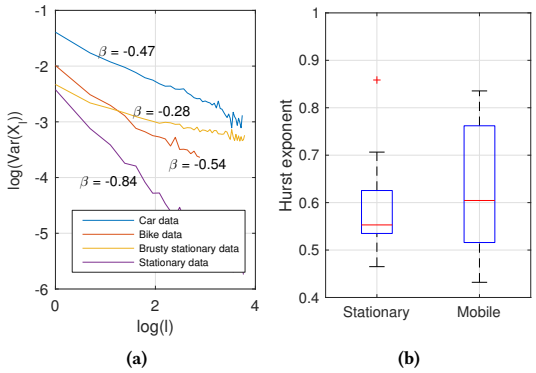


Figure 5: Burstiness quantification. (a) Variance-time plot for four different data sets. (b) Box plot showing burstiness in the two different data set types. There is more burstiness in mobile (car and bike) data. median($\hat{H}_{\text{stationary}}$) = 0.56, median($\hat{H}_{\text{bike}}$) = 0.6 and median($\hat{H}_{\text{car}}$) = 0.77.

By plotting $\log(\mathrm{Var}(\bar{X}_l))$ against $\log(l)$, as shown in Figure 5(a) for four data sets, the self-similar parameter H can be estimated (denoted by $\hat{H}$) by fitting a line to this plot and using the slope, β in

$$\hat{H} = 1 + \frac{\beta}{2}. \tag{4}$$

This value is between 0 and -1 since the slope, β is negative. Data streams without burstiness show a $\hat{H}$ value of around 0.5. Data streams with burstiness show larger values for $\hat{H}$. In Figure 5(b) a box plot of the calculated values for $\hat{H}$ for the different type of data sets (stationary and mobile) is shown.

We observe that the mobile data sets show on average more burstiness than the stationary data. For both data set types there are data sets that show burstiness. Burstiness in a mobile data set is due to mobility, while burstiness in stationary data sets is the result of channel effects.

Finding #2: Channel effects and mobility cause burstiness in frame loss in LoRaWAN.

For the design of the data recovery method it is relevant to know if burstiness occurs. When burstiness is present, the data recovery method should spread the redundant information more over time.

4.4 Modelling the Erasure Channel

Around half of our data sets show uniform distributed frame loss. So the erasure probability of a frame can be considered to be independent and identically distributed (IID). A valid channel model for these data sets is a Bernoulli channel model with erasure probability p_e. We define this as the default LoRaWAN communication channel.

For the data sets that show burstiness we use the Gilbert Elliot model, the most commonly used model for bursty erasure channels [7]. The Gilbert Elliot model is based on a Markov chain with two states: a good state where there is no frame loss and a bad state in which a frame is lost with probability p_{loss}. The probability to transit from the good state to the bad state is expressed with p_{GB}, and the probability of transition from the bad state to the good state

is expressed with p_{BG}. The channel model has an average erasure probability p_e that can be calculated with

$$p_e = \frac{p_{loss}}{1 + \frac{p_{BG}}{p_{GB}}}. \quad (5)$$

The parameters of the model to simulate the behaviour of this data set can be determined empirically. For example, the empirically calculated Gilbert Elliot model parameters that describe the burstiness of the car data are $(p_{GB}, p_{BG}, p_{loss}) = (0.25, 0.21, 0.85)$.

5 DARE: DATA RECOVERY IN LORAWAN

Most applications of LoRaWAN use Class A mode in which the end-device transmits its sensor data periodically but infrequently using an ALOHA-like protocol. We define the period of the sensor reading equal to the period of transmission of its sensor data, or "data unit". This data is at the heart of the IoT applications. Since frame loss leads to data loss, frame loss must be minimised as much as possible.

In the previous section, we have demonstrated that the frames transmitted over LoRaWAN can experience significant frame loss. Frame retransmission using acknowledgement schemes is possible over LoRaWAN, but it cannot be employed for all frames due to several reasons: (a) Gateways are expected to serve a large number of devices, but can only serve a limited number of acknowledgements, since gateways must obey the ISM-band duty cycle limit. Therefore, acknowledgements over LoRaWAN are not scalable. (b) Even if a work-around is used by employing other channels, retransmissions are still not scalable since the number of collisions will increase with the increasing number of devices. (c) If an acknowledgement and ARQ mechanisms are employed at the end-devices, it is more expensive in terms of energy for the end-devices.

This necessitates a solution to minimise data loss due to frame loss without employing acknowledgements. There are several requirements for such a solution: (a) Increase in transmissions must be minimal so as to be scalable and have minimal overhead in terms of energy. (b) No changes to the LoRaWAN specifications or network should be needed such that the solution works on existing applications as well. (c) The solution must be of low complexity at the sending side, since the end-device is an embedded device with limited computational and energy capabilities.

To this end, we propose a coding technique, called DaRe (Data Recovery), that borrows from both fountain and convolutional coding techniques, and works at the application layer. Thus DaRe can be employed for any application without changes to the LoRaWAN protocol. In this section, we shall first give an overview of how DaRe works before proceeding into its details.

5.1 DaRe - An Overview

For data recovery an existing data set should be extended with redundant information, such that the original data set can be recovered if only a subset of the transmitted data is received. LoRaWAN is a packet erasure channel (frames are either received or completely lost), so DaRe must spread the redundant information from the data in one frame across other frames. That way, a lost frame can be recovered using the redundant information from other frames.

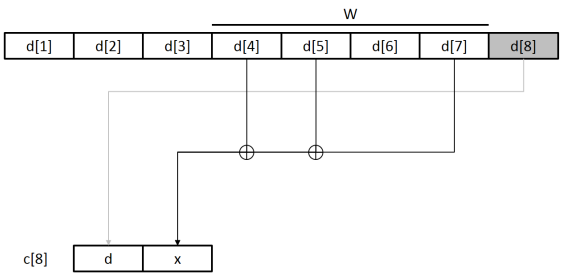


Figure 6: A schematic explanation of DaRe. In this example, R=1/2, W=4, Δ=0.75. The code words for the frame at time instance $t = 8$ are calculated by concatenating the data units from $t = 8$ and a parity check of previous data units from $t = 4, 5, 7$.

In order not to disrupt the communication stream of an existing LoRaWAN application, the moment a sensor reading is put in a data unit it should be transmitted in a LoRaWAN frame. Redundant information to be sent along in the frame can therefore only be computed from previous data units. So DaRe calculates the redundant information in a convolutional manner.

The redundant information included in a frame is a parity check of randomly selected previous data units. This is similar to a fountain code. A parity check is a vector of parity bits for each bit position in the data units. Traditional fountain codes perform the coding over a data block. But since we want the redundant information to be calculated in a convolutional manner, we use a sliding window approach with a finite window.

Since we intend to keep the complexity low for embedded devices, we work only with Galois Field 2 (GF(2)). This implies that the multiplications and additions are bit-wise 'AND' and 'XOR' operations respectively.

When using DaRe, an end-device sends frames containing the current data unit and redundant information. The amount of redundant information is determined by the code rate R. The number of previous data units is determined by the window size W and the degree Δ. A schematic explanation of determining the frame payload using DaRe is shown in Figure 6. The coding parameters are explained more in detail in the following section.

The performance of DaRe can be expressed using the data recovery ratio (DRR), defined as following:

$$\text{DRR} = \frac{\text{Number of data units recovered}}{\text{Number of data units transmitted}}. \quad (6)$$

5.2 Coding Parameters

There are three parameters in the context of DaRe: code rate (R), window size (W) and degree (Δ). We define them and discuss their influence on the coding scheme in this section.

5.2.1 Code rate. The code rate (R) is the ratio between the size of original data and the size of the data actually transmitted. It expresses the amount of redundant information added in transmission, and is calculated with the following equation:

$$R = \frac{\text{number of data bytes}}{\text{number of transmitted data bytes}}. \quad (7)$$

If we have data units of size y bytes, the size of the frame content after adding the redundant information will be y/R. A lower valued code rate is expected to provide more data recovery.

If the size of the data units, y, is one byte, the code rate can only be of the form $1/(y+x)$, where x is the number of redundant data units appended to the frame. This is because the smallest unit that can be easily processed on an embedded device is one byte. If y is more than one byte, data units can be split into k fragments if y is divisible by k. Then the code rate can be of the form $k/(k+x)$. In this case the calculation of the parity checks is done on data fragment level instead of data unit level. This allows for a higher number of code rate values; including code rates higher than 1/2 making the transmitted frames smaller with some redundancy.

Since the code rate determines the size of a transmitted frame, the code rate is limited by the maximum possible size of the frame. In LoRaWAN, the maximum allowed frame content is 55 bytes in SF12 and 222 bytes in SF7. The other spreading factors have a size between these two extreme values.

5.2.2 Window Size. The window size (W) expresses the number of previous data units to consider for calculating the redundant information. The window size is limited by the memory available on the end-device. A larger window size will spread redundancy for a single data unit over more frames, increasing the recovery probability of this data unit. A larger window size will also increase the maximum length of consecutive erasures that can be recovered. However, increasing the window size will also increase the recovery delay of the data unit.

5.2.3 Degree. The degree (Δ) expresses the relative number of previous data units from the selected window to include in the parity check. For example, in a situation where W=10 and Δ=0.5, there will be 5 data units included in a parity check.

For DaRe we choose the degree to be constant over different parity checks. Specific fountain code implementations propose a special degree distribution, but all these implementations are patented [14, 18]. To stay out of the way of these patents, and to keep the more research focused, the degree is kept constant.

The exact combination of data units in a parity check is picked randomly for each parity check. This is done to eliminate the influence of certain patterns in the selection. This leads to an average result for all different sequences of exact combinations.

We can compute an optimal value for the degree Δ for a given window size W. The optimal value means, for a given window size, there exists a value of Δ that offers the best recovery results. By using this optimal value of Δ in DaRe, we only need to input two parameters for coding: the window size W and the code rate R.

The degree has a major influence on the performance of the coding scheme. In Figure 7(a) the data recovery ratio (DRR) is plotted for a fixed code rate and window size while varying the degree, over channels with different IID erasure probabilities. The results are from simulations as described in Section 6. It can be seen that a too low or a too high degree gives a lower DRR. With a degree of 0, there is no redundant information so the data recovery ration is equal to DRR = $1 - p_e$. For degree values close to 1, the recovery

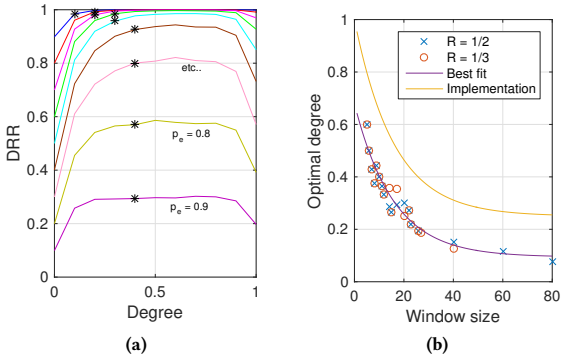


Figure 7: Determining optimal degree value Δ_{optimal}. (a) DRR for R=1/2 and W=10 for different erasure probabilities when varying Δ. The black crosses mark the lowest values for Δ that give the maximal DRR for different erasure probabilities. The largest of these values is Δ_{optimal}=0.4. (b) Δ_{optimal} for different values for R and W. There is an exponential relation between Δ_{optimal} and W. A relation with rounded-off fixed point coefficients is chosen as implementation.

ratio decreases as well. For each value of p_e there is a range for the degree value that gives the maximum possible DRR. The lowest value for Δ that gives the maximum possible DRR for a specific p_e is $\Delta_{\text{optimal}}(p_e)$ and is plotted as black crosses in Figure 7(a). The value that gives the maximum possible DRR for all p_e is Δ_{optimal}. For R=1/2 and W=10 the value for Δ_{optimal}=0.4.

In Figure 7(b) the value for Δ_{optimal} is plotted for different W and R. We can observe from the figure that R does not influence Δ_{optimal}, so Δ_{optimal} is only subject to W. We can also observe from the figure that the optimal degree value Δ_{optimal} and window size W have an exponential relation plotted as "Best fit".

In order to reduce the number of floating point computations for the embedded device, we round-off the coefficients of the relation between W and Δ_{optimal} to the closest fixed point binary representation. Although the coefficients are rounded-off, the value for Δ still lies in the range that gives maximal recovery rates for all erasure probabilities. The fixed point relationship is given by:

$$\Delta_{\text{optimal}}(W) = \left(\frac{1}{2} + \frac{1}{2^2}\right) \exp\left(-\frac{1}{2^4}W\right) + \frac{1}{2^2}, \quad (8)$$

and is plotted in Figure 7(b) as "Implementation".

5.3 Mathematical Framework

The framework presented here is a general framework to perform application level erasure coding and is applicable for Galois Fields higher than two as well. A glossary of symbols used in this section is provided in Table 1.

Let $d[t]$ represent data units generated at time instance t. All generated data units are of equal size and are divisible in k data fragments. By concatenating data units, we get the data set $D[t]$ that contains all generated data units previous to time instance t. $D[t]$ has maximum length W+1. Only the previous W previous data units need to be stored to calculate the redundant information. At time

Table 1: Glossary of symbols

Symbol	Description
$d[t]$	Data unit at a given time instance t of size k data fragments
$D[t]$	$D[t] = \{d[\tau] \mid \max(t - W - 1, 0) \leq \tau \leq t\}$
$c[t]$	Coded data at a given time instance.
$\mathbb{G}_t$	Generator matrix of time instance t describing relation between $D[t]$ and $C[t]$
D	$D = D[\infty]$
C	$C = \{c[\tau] \mid 0 \leq \tau \leq \infty\}$
$\mathbb{H}$	Matrix describing relation between D and C
C^*	The received coded data. $C^* = C \cup *$
$\mathbb{H}^*$	Relation matrix describing relations between C^* and D

instance t data unit $d[t]$ is transmitted concatenated with redundant information. This set is called $c[t]$. The redundant information is a parity check, so it can be expressed as a linear combination of previous data units with the following equation:

$$c[t] = D[t]\mathbb{G}_t, \tag{9}$$

where $\mathbb{G}$ is the generator matrix of size $k \times n$, and $n = k/R$.

Let the set C be defined as a concatenation of all values of $c[t]$, and the set D be defined as the set of all data units, we can state that,

$$C = D\mathbb{H}, \tag{10}$$

where,

$$\mathbb{H} = \begin{bmatrix} | & & & & \\ \mathbb{G}_1 & | & & & \\ | & \mathbb{G}_2 & | & & \\ & | & \mathbb{G}_3 & | & \\ & & | & \mathbb{G}_4 & \\ & & & | & \ddots \end{bmatrix} \tag{11}$$

Through the erasure channel some data will be lost. The decoder receives only a subset of C. This gives a set $C^* \subseteq C$ of received coded data. For each $c[t]$ in C^* the corresponding generator matrix $\mathbb{G}_t$ should be retrievable. All these generator matrices can be concatenated into one matrix $\mathbb{H}^*$ that contains the relations between the data units at each time instance and the received code words. If this matrix $\mathbb{H}^*$ is invertible, all the data units can be recovered using the following expression,

$$D = C^*(\mathbb{H}^*)^{-1}. \tag{12}$$

If $\mathbb{H}^*$ is not invertible, not all data can be recovered. However, the rank of matrix $\mathbb{H}^*$ expresses the amount of unique data in the matrix and is a metric for the amount of data that can theoretically be recovered by the decoder.

5.4 DaRe in Mathematical Framework

The mathematical framework can be used to simulate DaRe by describing a specific form of the generator matrix $\mathbb{G}_t$. Since the coding scheme is a systematic code, the first columns of the generator matrix should be an identity matrix in order to include the fragments of the data unit $d[t]$. The other code words are parity checks of previous data units.

This generator matrix for DaRe is of the form,

$$\mathbb{G}_{t,\text{DaRe}} = \begin{bmatrix} \mathbb{I} & 0 \\ 0 & P \end{bmatrix}, \tag{13}$$

where $\mathbb{I}$ is the identity matrix of size $k \times k$. P is the parity check matrix in which each of its $n - k$ columns contain parity check lines computed from randomly selected $k\Delta$ data fragments from a window of kW data fragments.

5.5 Benchmark: Conventional Repetition Coding

A simple form of frame redundancy is repetition, i.e. to append previous data units to a frame. This method provides some redundancy and allows for recovery, but gives a lot of overhead. This method can also be expressed in the proposed mathematical framework. The generator matrix $\mathbb{G}_t$ would be time invariant and equal to an identity matrix $\mathbb{I}$ of size $n \times n$. We use this coding method as a benchmark for DaRe. It provides a reference for the performance.

6 NUMERICAL RESULTS

We perform numerical simulations with the mathematical framework in MATLAB to compare the performance of DaRe with the conventional coding. The metric used for comparison is data recovery ratio as defined in Equation 6. We determine the results for different R and W over both IID and bursty channels. Figure 8 shows the results from these simulations.

The benchmark results from the conventional coding are shown in Figure 8(a). It is intuitive that as the value for R decreases, DRR improves. We observe that for a DRR of 0.99 and code rate of 1/2, the maximum tolerable erasure probability is p_e=0.1. Figure 8(b) presents the results for DaRe for different R values with W=80. Compared with the benchmark, DaRe has higher recovery results. With R=1/2, a DRR of 0.99 can be achieved up to p_e=0.43. In Figure 8(c) the influence of W on the DRR is plotted. With an increasing window size we observe a higher DRR, but the improvement obtained after W=20 is quite insignificant. With W=10 a DRR of 0.99 is achievable up to p_e=0.27.

The results until now are with an IID erasure channel model. In Figure 8(d) results for a bursty channel are presented. We use the Gilbert Elliot model with parameters $(p_{GB}, p_{BG}) = (0.25, 0.21)$, while p_{loss} is varied. The results are plotted for p_e as defined in Equation 5. A value of $p_{loss} = 0.85$ describes burstiness similar to the observed car data set. For W=10, the maximum p_e for DRR=0.99 is 0.27. There is not much difference in performance for larger window sizes compared to Figure 8(c). Therefore, we can conclude that DaRe can handle both bursty and non-bursty erasures equally well.

7 PRACTICAL EVALUATION

In this section, we present implementation details for a DaRe encoder and decoder on an end-device and application server respectively. Furthermore, we present a method that helps adapt the parameters of DaRe on a real LoRaWAN network. Then we present results and analyse from the implementation.

7.1 Implementation

In order to get DaRe working in real-life there are two challenges: (a) The decoder must be able to construct the generator matrix similar to the one used for encoding in order to decode the redundant information, and (b) An algorithm is needed to recover previously

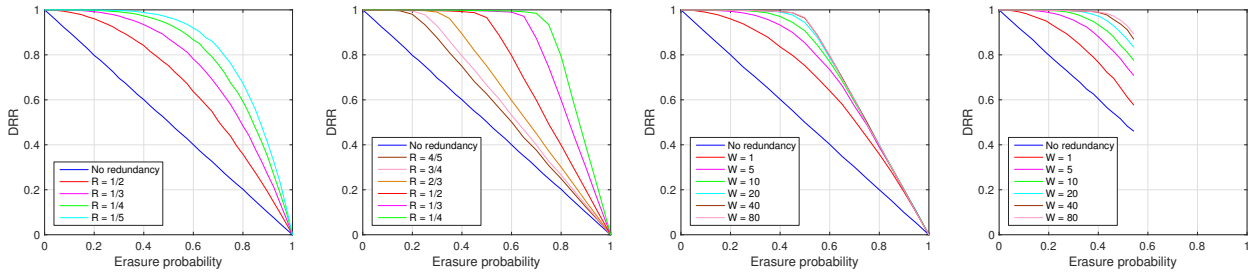


Figure 8: Simulation results for DaRe. (a) DRR for conventional repetition coding, the baseline. (b) DRR for DaRe with different values of R and W=80. Higher values of R give better results. (c) DRR for DaRe with different values of W and R=1/2. Smaller values of W give better results up to W=20. (d) DRR for DaRe with different values for W and R=1/2 for a Gilbert Elliot channel model $(p_{GB}, p_{BG}) = (0.25, 0.21)$ varying p_{loss} and plotting for p_e. p_e=0.46 is comparable to the car data.

lost data units from successfully received frames. We describe the aspects to solve these challenges.

7.1.1 Generator Matrix at Decoder. In DaRe coding, the parity check lines with size W and degree Δ are generated randomly. To be able to decode, the decoder must know the randomly generated parity check lines. This should not be sent on the LoRaWAN frame due to the limited payload size and the energy constraints of the device. We therefore compute the generator matrix on the end-device using a pseudo-random number generator (PRNG), such that the generator matrix can be reconstructed identically by the decoder. The PRNG is seeded with the frame sequence number.

We use a linear feedback shift register to implement the PRNG. The last $log_2(W)$ bits of the shift register are a pseudo-random number with a maximum value of $W-1$, which can be used as the index pointing to the data unit to be included in the parity check. By taking $W\Delta$ distinct numbers from the PRNG a pseudo-random parity check line of size W and degree Δ is generated.

7.1.2 Decoding Algorithm. We take our decoding strategy from the traditional parity check decoding methods. The decoder works in an iterative way as is explained in the flowchart in Figure 9. Each time a frame is received, the data unit in this frame is stored and the relation matrix $\mathbb{H}$ between the received parity checks and the data units is constructed. Known data units are removed from the matrix and Gaussian elimination is performed to remove linear dependence from the matrix. If this reduced matrix has invertible columns it means data units can be recovered from the matrix. The relation matrix is again reduced until no more columns can be inverted. If there is a leftover matrix, it is stored for the next iteration.

7.1.3 LoRaWAN Payload Format for DaRe. The coding parameters W and R will be sent in the payload of every frame along with the coded data, as shown in Figure 10. We implement lookup tables for all the possible values of R and W. Therefore, the end-device only needs to send the indices to the values used in the lookup tables for R and W. Since the number of entries in the tables are limited, we require only four bits each for sending the indices of W and R.

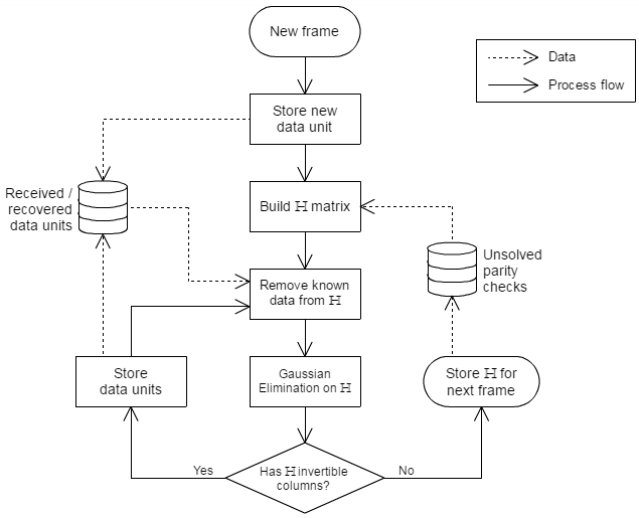


Figure 9: The decoding algorithm visualised in a flowchart. Data recovery is done by iteratively solving received parity checks when a frame is received.

R	W	$c[t]$
1 Byte		(k/R) data units

Figure 10: LoRaWAN payload format for DaRe.

7.1.4 Choosing and Adapting the Parameters. As explained in Section 5.2, the two main parameters of DaRe, R and W, should be chosen, and Δ is pre-computed with the chosen value for W. R and W are limited to the constraints set by the payload size and end-device memory respectively. By using the results in Figure 8, the parameter settings providing a desired DRR for the expected erasure probability and burstiness can be determined.

A static choice of parameters may offer varied data recovery performance depending on the long term changes in the erasures of

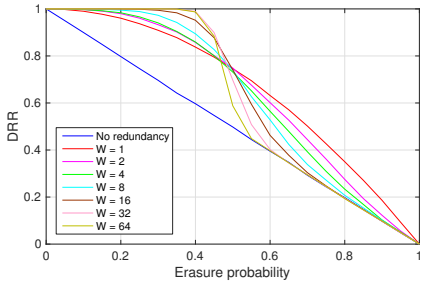


Figure 11: DRR for emulation of the implementation of DaRe for different window size W and R=1/2.

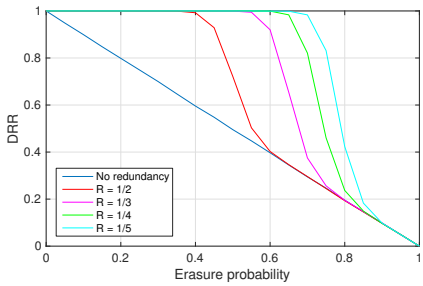


Figure 12: DRR for emulation of the implementation of DaRe for different coding rates and W=32.

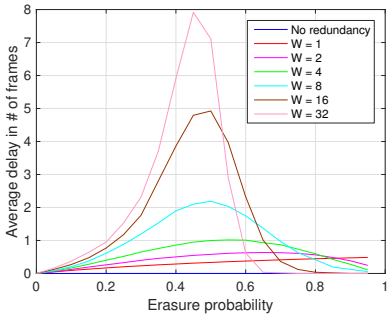


Figure 13: Recovery delay for emulation for different window size W and R=1/2.

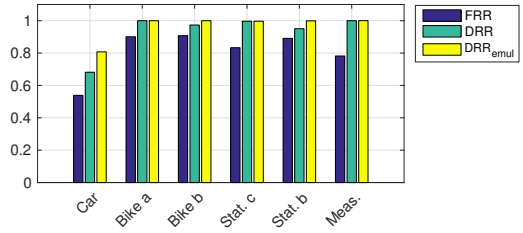


Figure 14: DRR in emulation results for five previously collected data sets and one measurement, using parameters R=1/2, W=8. 'Stat.' is stationary data. 'Bike b' and 'Stat. b' are bursty data sets. 'Meas.' is a data set acquired with the device running DaRe.

the channel. One way to adapt these parameters is to use a downlink LoRaWAN frame to notify the end-device about the average channel condition in the recent past. Typical LoRaWAN deployments also know the device locations. The downlink frame may contain the observed average erasures from the device and near-by devices in the recent past allowing the end-device to dynamically adapt the parameters in run-time to provide optimal performance for the new circumstances.

7.2 Evaluation Results

We present results from evaluating our implementation of DaRe on an IID channel that was emulated between the end-device and the application server. Figure 11 shows the results for DRR for various window sizes. The implementation gives lower recovery rates compared to the simulation results in Figure 8(c), especially for higher erasure probabilities. The difference is in the fact that while there can be unique information in the relation between the coded data and the data units, it does not mean all the data units can be recovered. We see that larger W leads to decrease of DRR faster at p_e=0.5. However, for $p_e < 0.4$, the results are comparable to the simulation results. For W=32, a DRR 0.99 is achieved up to p_e=0.4.

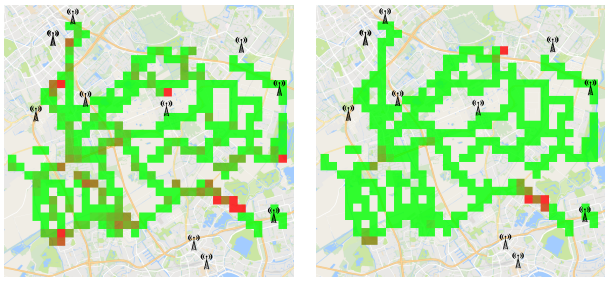
The maximum recovery results for different values of the code rate R are shown in Figure 12. It can be seen that the DRR increases for smaller values of R, however the maximal erasure probability to achieve a DRR of 0.99 is 4% lower compared to the simulation results in Figure 8(b).

The iterative decoding approach introduces a delay in the recovery of missed data units. Depending on the LoRaWAN application, delay can be a factor as well to determine the configuration parameters. When more frames are lost a larger delay can be expected, because more parity checks need to be received to get invertible columns in $\mathbb{H}^*$. While larger window sizes increase the DRR as shown in Figure 11, the average time taken before a previously lost data unit is recovered also increases as can be seen in Figure 13. For higher loss of frames more unrecovered data leads to a lower average recovery delay, resulting in a bell-shaped curve. The maximum average delay difference between W=16 and 32 is 2.8 frames.

7.3 Measurement Results

To evaluate DaRe for real-life data sets, we have (a) applied DaRe to the previously collected data, and (b) performed some measurements with the end-device running DaRe. In Figure 14 the DRR is given for six data sets, five previously collected and one newly collected with DaRe. In this figure, the DRR is the observed data recovery ratio and DRR_{emul} is the expected data recovery ratio if the data set has IID erasures. We conclude that the DRR is equal to the expected DRR for non-bursty data sets, and a little lower for bursty data sets.

Figure 15 shows a map of the DRR before and after DaRe coding is applied to the previously collected bike data. While there was 14% data loss without DaRe, we see that this number reduces to 3% with DaRe, illustrated by more green blocks in the figure on the right.



(a) (b)

Figure 15: Geographical measurement results. (a) FRR for the bike data without coding. p_e=0.14. (b) DRR using DaRe with R=1/2, W=8. p_e=0.03. On a color scale from green to red the frame loss is expressed. Green blocks indicate no loss and red blocks indicate loss of all frames.

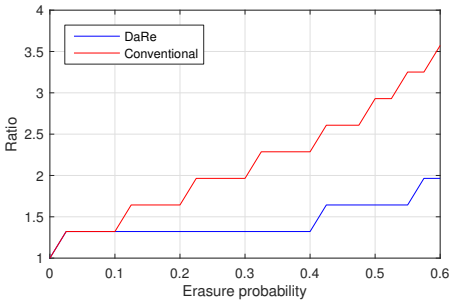


Figure 16: Transmission time increase for a 10 byte data unit for a desired DRR of 0.99, relative to the transmission time for uncoded data transmission. DaRe provides significant transmission time reduction.

7.3.1 Additional Transmission Time. Adding redundancy in communication, like with DaRe, requires transmitting more bytes. The largest contributor to energy consumption on an end-device is the frame transmission. Sending more bytes leads to significantly more energy consumption than additional computations. The impact of DaRe on the energy consumption of the end-device can be determined by calculating the additional transmission time needed. The air time of a LoRaWAN frame can be calculated using the formula given in [21].

Figure 16 shows the ratio of transmission time for DaRe and conventional coding when 99% data recovery is desired, compared to the transmission time when sending data without coding. DaRe reduces the additional transmission time compared to conventional coding up to 42% for a data unit size of 10 bytes. Larger data unit sizes will give even more transmission time reduction.

8 RELATED WORK

In this section we briefly discuss existing studies on LoRa, LoRaWAN and erasure coding to position our work.

8.1 LoRaWAN

Bor et al. have done extensive studies on IoT radios [5]. They concluded that LoRaWAN has larger communication ranges and other interesting features over other solutions, such as SigFox, NB-IoT, and Weightless. Centenaro et al. did a coverage test of LoRaWAN in the urban environment of the Italian city of Padova [6], and found a nominal coverage range of 1.2 km. Aref et al. reported on the range of LoRa using different physical layer configurations [1]. Petäjäjärvi et al. looked into the range of LoRaWAN [16] and observed a maximum communication range of 15 km on ground and close to 30 km on water. They also looked into LoRaWAN coverage in indoor environments [17]. In their test setup a minimum of 96% indoor frame reception rate was observed for distances up to 420 m. Mikhaylov and Petäjäjärvi et al. looked more closely into the limits of device and gateway throughput [13], giving the first insights on LoRaWAN network performance.

Most recently, Augustin et al. did a very detailed study into the properties of LoRa and LoRaWAN [2]. One of their findings is that using confirmed messages significantly reduce the throughput in a LoRaWAN network. They elaborate on properties of a LoRaWAN network, however, there is still no literature on measurements of LoRaWAN networks. Some measurements have been done on coverage of a single gateway, but not multiple gateways. In this paper we conducted measurements on a LoRaWAN network with multiple gateways. Furthermore all measurements in related studies are, hitherto, spatial measurements, mostly on coverage. In this paper we conducted the first temporal analysis of LoRaWAN communication.

8.2 Erasure Coding

Erasure channels characterise and model communication channels, describing how transmitted messages are either received, or erased. In the context of LoRaWAN there is a frame erasure channel, where complete frames are received or not. The erasure channel is different from the error channels, where messages are altered, but still the frames are received.

In communication theory majorly two methods exist to deal with erasures: Automatic Repeat reQuest (ARQ) and Forward Error Correction (FEC). The former uses retransmissions to retrieve lost data. With FEC messages are encoded in order to handle errors in transmission. In our situation, since downlink communication is expensive in LoRaWAN, FEC would be the best way of adding robustness in LoRaWAN communication.

Erasure codes are FEC for an erasure channel. Erasure coding can be found amongst others in distributed data storage, media streaming, multicast communication and deep space communication. The earliest erasure codes are Reed-Solomon codes [12]. However, these are resource expensive to use for large data sets. In 1960, Gallager developed the concept of Low-density parity-check codes (LDPC codes). LDPC codes are used in, for instance, the new DVB-S2 standard [15]. With LDPC codes data blocks are supplemented with parity bits before transmission. LDPC codes are block codes, where data is encoded in blocks.

Next to block codes are convolutional codes, which add redundancy via a sliding window instead of a data block. An example of a convolutional code is the class of Turbo codes [4]. Turbo codes

are iterative codes, where messages are sent along with parity bits computed in a recursive manner. The messages are decoded in a similar recursive way using soft decision.

Both LDPC and Turbo codes do error and erasure detection and correction. However there are communication schemes where erasure detection is done separately. In Fountain codes error detection is presumed given. Fountain codes were introduced by Luby [11, 12]. Code words are a linear combination of data fragments. At the decoder a random combination of as much linearly independent code words as number of data fragments can be used to recover all data. Examples of fountain codes are LT-codes [10] and Raptor codes [22]. With fountain codes the principle of rateless codes was also introduced. Rateless codes can produce an infinite number of code words, instead of a fixed number of code words in rated codes. Fountain codes are block codes, like LDPC codes.

Fountain codes provide a high level erasure resilience. Lost data fragments can be recovered from any code word. This principle could also be applied to a sliding window, making a convolutional fountain code. While there has been done some work on combining fountain codes and convolutional codes [26], there is none on applying fountain codes in a convolutional manner.

9 CONCLUSION

With many IoT applications on the rise, the number of IoT devices is proliferating. To cater to the communication needs of this large number of IoT applications, to collect data from these devices, new architectures and protocols have been proposed recently. One such protocol is LoRaWAN, a Low Power Wide Area Network (LPWAN) technology. While LoRaWAN can provide a large coverage, its basic operating mode cannot guarantee successful frame receptions due to the unreliable wireless channel and absence of retransmission schemes. In this paper we characterised frame loss in LoRaWAN. We performed large scale measurements in an almost collision free network deployment for different scenarios. With the data sets, we characterised frame loss in the network in terms of spatial and temporal properties. We found that there is significant frame loss. Furthermore, we demonstrated the frame loss can be bursty in nature even when a end-device is stationary.

Conventional wireless techniques such as using ACK for every transmitted frame are withheld in LoRaWAN to provide scalability: and to save transmission time on the gateways and also energy on the end-devices. Thus, it is up to the application layer to guarantee and increase the data reception. To this end, we propose a novel erasure coding method, DaRe, that reduces data loss in LoRaWAN significantly. DaRe is based on applying fountain codes on a sliding window. To simulate DaRe we described an algebraic framework. We evaluated DaRe both with emulations and by implementing it on an end-device. We achieved significant recovery of 99% with a code rate R=1/2 when channel erasure probability was 0.4. Comparing with a naive repetition coding method, DaRe reduces energy requirement up to a factor of 0.42. Further, we showed that DaRe can lower an average data loss of 14% in 300 km of biking to 3% data loss with R=1/2 and W=8.

Since the network is newly deployed, collisions are negligible. When more devices start connecting to the network collisions will start contributing significantly to the overall frame loss in the network. There will be an increased need for data recovery methods that do not add much load to the network. Therefore, we believe that DaRe provides a sustainable solution to improve the data throughput in LoRaWAN.

REFERENCES

[1] M. Aref and A. Sikora. 2014. Free space range measurements with Semtech Lora technology. In *2014 (IDAACS-SWS)*. 19–23.

[2] Alos Augustin, Jiazi Yi, Thomas Clausen, and William Townsley. 2016. A Study of LoRa: Long Range & Low Power Networks for the Internet of Things. *Sensors* 16, 9 (Sep 2016), 1466.

[3] Janina Bartje. 2016. The top 10 IoT application areas - based on real IoT projects. https://iot-analytics.com/top-10-iot-project-application-areas-q3-2016/. (aug 2016). Accessed: 2016-01-13.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima. 1993. Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Communications, 1993. ICC '93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, Vol. 2. 1064–1070 vol.2.

[5] Martin Bor, John Vidler, and Utz Roedig. 2016. LoRa for the Internet of Things. In *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*. Junction Publishing, 361–366.

[6] Marco Centenaro, Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. 2015. Long-Range Communications in Unlicensed Bands: the Rising Stars in the IoT and Smart City Scenarios. *CoRR* abs/1510.00620 (2015).

[7] E. N. Gilbert. 1960. Capacity of a burst-noise channel. *The Bell System Technical Journal* 39, 5 (Sept 1960), 1253–1265.

[8] Andrea Goldsmith. 2005. *Wireless communications*. Cambridge university press.

[9] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. 1994. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking* 2, 1 (Feb 1994), 1–15.

[10] M. Luby. 2002. LT codes. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*. 271–280.

[11] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman. 2001. Efficient erasure correcting codes. *IEEE Transactions on Information Theory* 47, 2 (Feb 2001), 569–584.

[12] D. J. C. MacKay. 2005. Fountain codes. *IEEE Proceedings - Communications* 152, 6 (Dec 2005), 1062–1068.

[13] K. Mikhaylov, Juha Petaejaejaervi, and T. Haenninen. 2016. Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology. In *European Wireless 2016; 22th European Wireless Conference*. 1–6.

[14] M. Mitzenmacher. 2004. Digital fountains: a survey and look forward. In *Information Theory Workshop, 2004. IEEE*. 271–276.

[15] A. Morello and V. Mignone. 2006. DVB-S2: The Second Generation Standard for Satellite Broad-Band Services. *Proc. IEEE* 94, 1 (Jan 2006), 210–227.

[16] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo. 2015. On the coverage of LPWANs: range evaluation and channel attenuation model for LoRa technology. In *ITS Telecommunications (ITST), 2015 14th International Conference on*. 55–59.

[17] J. Petjjrvi, K. Mikhaylov, M. Hmlinen, and J. Iinatti. 2016. Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring. In *2016 ISMICT*. 1–5.

[18] Jalaluddin Qureshi, Chuan Heng Foh, and Jianfei Cai. 2013. Primer and Recent Developments on Fountain Codes. *CoRR* abs/1305.0918 (2013). http://arxiv.org/abs/1305.0918

[19] Semtech. 2013. ETSI Compliance of the SX1272/3 LoRa Modem. http://www.semtech.com/images/datasheet/etsi-compliance-sx1272-lora-modem.pdf. (July 2013).

[20] Semtech. 2015. LoRaffl Modulation Basics. http://semtech.com/images/datasheet/an1200.22.pdf. (may 2015).

[21] Semtech. 2015. SX1272/73 - 860 MHz to 1020 MHz Low Power Long Range Transceiver. (March 2015).

[22] A. Shokrollahi. 2006. Raptor codes. *IEEE Transactions on Information Theory* 52, 6 (June 2006), 2551–2567.

[23] Sodaq. 2016. Sodaq Mbili. http://support.sodaq.com/sodaq-one/mbili/schema-rev-4/. (2016). Accessed: 2016-10-09.

[24] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent. 2015. LoRaWANffl Specification. https://www.lora-alliance.org/portals/0/specs/LoRaWAN%20Specification%201R0.pdf. (jan 2015).

[25] Texas Instruments. 2005. ISM-Band and Short Range Device Regulatory Compliance Overview. http://www.ti.com/lit/an/swra048/swra048.pdf. (may 2005).

[26] Mohammed Usman. 2012. Convolutional fountain distribution over fading wireless channels. *International Journal of Electronics* 99, 8 (2012), 1037–1050.