Application Layer Coding for IoT: Benefits, Limitations, and Implementation Aspects

Magnus Sandell, *Senior Member, IEEE* and Usman Raza

***Abstract*—One of the key technologies for future Internet of things (IoT)/machine-to-machine systems is low-power wide area networks, which are designed to support a massive number of low-end devices, often in the unlicensed shared spectrum using random access protocols. However, these usually operate without centralized control and since automatic repeat-request and acknowledgment mechanisms are not very effective due to the strict duty cycles limits and high interference in the shared bands, many packets are lost from collisions. In this paper, we analyze a recently proposed application layer coding scheme introduced in [1], which aims to recover lost packets by introducing redundancy in the form of a fountain code. We show how latency and decoding complexity is affected by the packet loss rate but also prove that there is a limit to what can be achieved by introducing more redundancy. The analysis is backed up by simulation results.**

***Index Terms*—Channel coding, sensor systems and applications, wide area networks.**

I. Introduction

LOW-POWER wide area (LPWA) networks are forecasted to connect a massive number of devices in future Internet of things (IoT)/machine-to-machine networks. Traditionally, cellular technologies (2G, 3G, 4G, etc.) and short-range wireless technologies (WLANs, ZigBee, Bluetooth, etc.) have been used for this purpose. However, their higher cost remains prohibitive for a wider adoption for applications that require inexpensive connectivity and low-end devices to monitor our cities, industry, infrastructure, and logistics. The emerging LPWA technologies such as LoRaWAN [2], SIGFOX [3], Ingenu RPMA [4], NB-IoT [5], and NB-FI [6] are designed to provide a better coverage than the existing cellular networks at significantly lower cost and power consumption. A long range of multiple kilometers saves the LPWA technologies from the hassle of deploying very dense networks and thus avoids the exorbitant cost and the maintenance effort associated with short-range wireless technologies. This also enables the end devices to connect to the network directly over a single hop, simplifying the design of the protocol stacks compared to multihop wireless technologies. Both business and technological benefits of these technologies are quickly realized by the industry. A number of mobile operators, independent users, and crowd-sourced start-up companies are already making strides in deploying LPWA networks across the globe.

Motivated by this fast adoption of the LPWA technologies, many recent works studied their performance and uncovered their practical limitations in providing reliable and scalable connectivity to a massive number of devices. It has become evident that these technologies are very prone to the intratechnology interference [7], cross-technology interference [8], high frame loss [1], and low capacity [9], clearly stressing the need for additional mechanisms to increase the reliability of the LPWA technologies. These reported problems are attributed to the combination of features that are unique to some LPWA technologies and were not present in the other long-range wireless cellular technologies and thus were not studied in detail earlier. These include the use of the license-exempt ISM bands and the random-access medium control protocols (such as ALOHA), as well as the transmission duty cycle limitations dictated by the regulations on the sub-GHz ISM bands across the globe. To offer an example, LoRaWAN [2] and SIGFOX [3], two popular technologies that use the sub-GHz ISM bands and ALOHA protocol, are subject to a 1% duty cycle limit for all wireless devices in most sub-bands in Europe. To respect this, the base stations can neither serve a large number of end devices [10] nor acknowledge all uplink transmissions from end devices. This means that the reliability-enhancing mechanisms such as automatic repeat request (ARQ) are not very effective because they require the base stations to acknowledge the uplink transmissions. Pop *et al.* [11] show that as the LoRaWAN network scales, the base stations are frequently not able to acknowledge the successfully received uplink messages due to the duty cycle limit, leading to retransmissions from the end devices and thus collisions in the network. Due to this reason, the ARQ-based schemes can harm rather than improve the overall reliability of the large networks. Marcelis *et al.* [1] show that application layer coding schemes improve reliability of LoRaWAN and do not require any downlink communication.

In this paper, we analyze the behavior of this application layer coding dubbed DaRe [1]. As LPWA technologies will handle up to millions of devices, it is of utmost importance to decode the received messages in minimal time and with low complexity. To this effect, low complexity decoding techniques are presented and shown to enable quick and efficient decoding of the stream of received packets by the cloud. As the proposed encoding and decoding techniques are not tied to any particular LPWA technology, they can be applied to a wide range of low-power networks.

Manuscript received August 8, 2017; accepted December 22, 2017. Date of publication January 30, 2018; date of current version February 22, 2019. *(Corresponding author: Magnus Sandell.)*

The authors are with the Telecommunications Research Laboratory, Toshiba Research Europe Ltd., Bristol BS1 4ND, U.K. (e-mail: magnus.sandell@toshiba-trel.com; usman.raza@toshiba-trel.com).

Digital Object Identifier 10.1109/JSYST.2018.2791659

Our Contributions

1) We analyze the application layer coding scheme in [1] and show how the code rate will impact system performance. We prove that for large packet loss probabilities, it is not sufficient to reduce the code rate since this will increase the interference to a critical level.
2) We consider latency as a metric and show how this depends on the packet loss probability.
3) We devise a novel decoding scheme that can reduce the complexity and latency at a small price in data recovery rate (DRR).

The rest of the paper is organized as follows. A brief background of LPWA is given in Section II as well as a short overview of fountain codes. The application layer coding is presented in Section III, simulation results in Section IV, and decoding is discussed in Section V. Finally, conclusions are drawn in Section VI.

II. Background

This section first describes in detail the unique peculiarities and challenges of the sub-GHz LPWA technologies that result in low reliability in the networks. We then provide a brief overview of fountain codes as a solution to improve application layer reliability of IoT applications.

A. Low Reliability of LPWA Technologies

The landscape of LPWA technologies is crowded with multiple competing technologies deployed in the license-free ISM bands as well as the licensed bands. This paper, however, focuses only on the former that are less reliable due to the reasons discussed next.

Link asymmetry: Most LPWA systems are characterized by highly asymmetric links with a dominant uplink compared to the downlink. These include SIGFOX [3], Ingenu RPMA [4], and IEEE 802.15.4k [12]. Some other LPWA technologies like Weightless-N [13] do not even provide any downlink support. The reasons for link asymmetry include the use of different modulation techniques in the up- and downlink, minimization of listening time at end devices to reduce their energy consumption, and the regional spectrum regulations. In this scenario when the downlink and uplink transmissions are significantly unbalanced, the reliability of the uplink transmissions cannot rely much on the downlink traffic such as acknowledgments. In fact, the techniques should be used to make the uplink transmissions more robust and resilient to packet losses in isolation with the downlink.

Spectrum regulations: Spectrum regulations on the use of sub-GHz ISM band across the globe are yet another contributing factor to the link asymmetry. These regulations often limit transmission power and transmission duty cycle to efficiently share the finite radio resources among the coexisting technologies. The transmission duty cycle limit implies that the end devices and the base stations can transmit only a few messages in a day. For example, SIGFOX under the strictest settings allows an end device to transmit a maximum of 140 uplink messages but receive only 4 downlink messages from a base station. Even LoRaWAN base stations can only transmit a limited number of downlink messages, preventing them from acknowledging more than a small fraction of uplink messages as the networks scale. Again due to this reason, the downlink acknowledgments cannot be relied upon to make the uplink more reliable. In case of a base station that is operating close to its transmission duty cycle limit, acknowledgments are not guaranteed against successful receptions. This may result in more retransmissions and more congestion in the network, leading to low reliability in the network.

High packet loss and interference: The practical trials with the large-scale LoRaWAN, one of the prominent LPWA technologies, have shown a high frame loss even in the absence of high interference [1]. An inevitable growth in the number of LoRaWAN devices will further increase levels of intratechnology interference. It has been shown that coverage probability will drop exponentially with network size in such networks [7] due to collisions in their use of “virtual channels.” High interference is also a by-product of using simplistic MAC protocols based on ALOHA by LoRaWAN and SIGFOX. While ALOHA simplifies the design of a medium access mechanism, its uncoordinated operation results in more collisions and excessive internetwork interference. As more IoT devices will start using different wireless technologies in the sub-GHz ISM bands, cross-technology interference is bounded to increase, further limiting the network reliability.

Despite the limited reliability offered by many LPWA technologies, some IoT applications may still require certain reliability guarantees. In this paper, we approach this problem with a low complexity encoding/decoding technique based on fountain codes. Fig. 1 shows the general architecture in which the end devices encode the application layer messages before sending them over any LPWA technology. The use of the encoding techniques on top of the LPWA technologies brings the benefit of not requiring any modification in the underlying LPWA technologies, which are often proprietary and are usually implemented in hardware. The encoded message travels through the radio access network and back-end system to the application servers, which decode the message. All packets are identified by a sequence number, enabling the application server to establish which packets have been lost. It can then recover these packets once a sufficient number of subsequent packets have been successfully received, so that the introduced redundancy can be exploited for decoding.

B. Fountain Codes

A fountain code is a class of erasure codes that has the property that a (potentially) infinite sequence of encoded symbols can be generated. Conventional block erasure codes on the other hand, such as Reed–Solomon codes, have a fixed structure. If a block of k data symbols is encoded into n symbols, recovery of the data is possible if any k encoded symbols are received. Fountain codes have the ability to continuously generate different encoded symbols from which data recovery is possible with probability $1 - 2^{-k\epsilon}$ if $k(1 + \epsilon)$ symbols have been received.

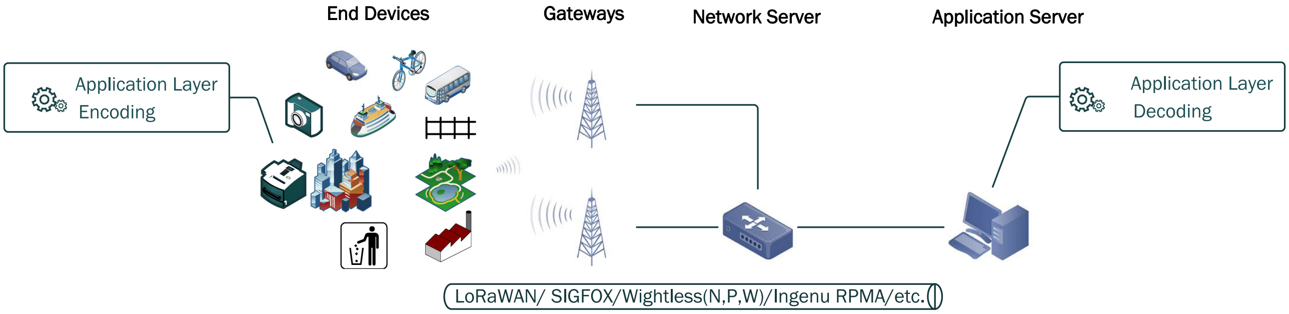
End Devices
Gateways
Network Server
Application Server
Application Layer Encoding
Application Layer Decoding
LoRaWAN/ SIGFOX/Wightless(N,P,W)/Ingenu RPMA/etc.

Fig. 1. Application layer coding over LPWA technologies.

This property makes them a *rateless code* since the code rate is not fixed. The small extra overhead ϵ makes it exceedingly likely that the receiver can recover lost symbols, and hence, the receiver only needs to wait until a sufficient number of symbols are available.

Each encoded symbol is a random combination of the data symbols. Mathematically speaking, the encoded symbols c_j are formed from the data symbols d_i by the matrix multiplication

$$\mathbf{c} = \mathbf{d}\mathbf{G}$$
$$(c_1, \ldots, c_n) = (d_1, \ldots, d_k)\,(\mathbf{g}_1, \ldots, \mathbf{g}_k)\,. \tag{1}$$

The ones[1] in the vectors $\mathbf{g}_j \in \{0,1\}^{k\times 1}$ indicate which data symbols are used to create the encoded symbol c_j. Note that the encoding process can be done by simple exclusive OR (XOR) operations. The received symbols will then correspond to columns of the generator matrix $\mathbf{G}$ and if k linearly independent columns are available (the formed submatrix $\mathbf{G}'$ has full rank), it can be inverted and the lost data symbol is recovered. Although this can be achieved with high probability, the decoding complexity of an optimal decoder would be $\mathcal{O}\left(k^3\right)$, making it impractical (decoding and complexity are discussed further in Section V). What is desired is instead linear (in k) encoding and decoding cost.

The first practical fountain code achieving this was the Luby transform (LT) code [14]. The decoder is based on a message-passing algorithm (for more details, see Section V), which can find the lost symbols in linear time. To allow the decoder to process received symbols with high probability, the distribution of which data symbols are combined as encoded symbols (the ones in $\mathbf{g}_j$) is optimized. The number of data symbols D is first randomly chosen from a certain distribution and then D data symbols are chosen at random. The distribution of D, also known as the degree distribution, was derived in [14] and is known as the robust Soliton distribution.

Further developments on fountain codes have led to the introduction of Raptor codes. These are similar to LT codes but feature a precoding step with (usually) a conventional erasure code. The idea is that the LT code can recover a large fraction of the missing data with high probability, which then allows the outer code to recover the rest with high probability. These codes are very efficient and have been chosen for several standards, such as 3GPP MBMS [15] (streaming services), DVB-H IPDC [16] (IP services over DVB networks), and DVB-IPTV [17] (TV services over IP networks).

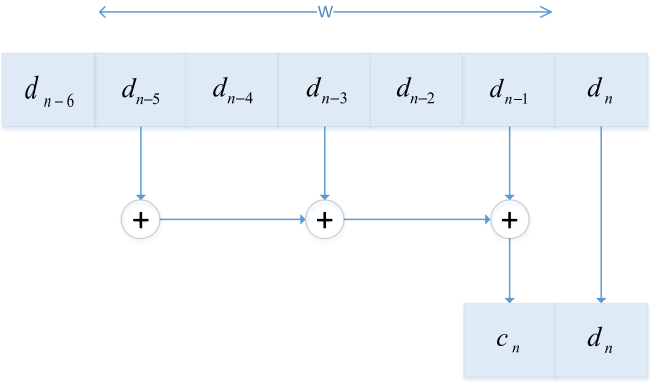

Fig. 2. Data encoding principle with memory $W = 5$ and degree $D = 3$. The transmitted symbols are d_n and c_n.

III. Application Layer Coding

To improve the performance of LoRaWAN without changing its specification, a fountain code based scheme on the application layer was proposed in [1]. In order to recover packets lost due to collisions, fading, and/or shadowing, packets are amended with redundancy. This basic principle is shown in Fig. 2 [1].

Note that we assume a systematic code, i.e., the data appear as one of the transmitted symbols and the redundancy is purely in the added parity symbol. Each packet carries, apart from its data, a parity symbol that is generated as a linear combination of previous data symbols. To simplify operations, all encoding is done on a bit level with XOR; this can then be repeated for all bits in the packet. Following the principle of fountain codes, the data symbols used for the redundancy change between the different parity symbols. This random coding approach allows the receiver (on average) to receive sufficiently different linear combinations of data symbols to be able to recover the missing ones with high probability.

A few operational differences to conventional fountain codes are worth pointing out. Since the end devices continuously deliver data, there is no fixed data size k since it grows with time. This means that we need to limit the "memory" of the encoder by using a sliding window. If the encoded parity symbol

[1] In general, other Galois fields than GF (2) can be used; however, in this paper, we limit the discussion to binary elements.

is allowed to depend on all previous data symbols, the end device would need to buffer all data symbols from the past. This is obviously not possible in practice, so a parity symbol p_n can only depend on data symbols in a subset of $\{d_{n-W}, \ldots, d_{n-1}\}$, where W denotes the memory. Note that it does not make sense to make c_n dependent on d_n since if packet n is lost, parity symbol c_n is also lost and cannot, hence, be used to recover d_n. One consequence of the finite memory W is that the generator matrix $\mathbf{G}$ will be banded, i.e., only W rows above the diagonal can have nonzero values. Because of this restriction, creating a degree distribution according to the LT code is difficult; in [1], a fixed degree D was used. It is also worth noting that the degree distribution is designed to optimize the reduced complexity decoder; if an optimal decoder is used, this is not necessary (for more details on the decoding, see Section V). Due to the finite memory W, the coding appears as a combination of fountain and convolutional codes. However, it can also be viewed as a special case of windowed erasure codes [18].

It is worth noting that these random combinations of data symbols must be known to both the transmitter and receiver. Rather than transmitting side information for this purpose, a pseudorandom number generator can be used [1]. By using the same one on both the encoder and decoder side with the same seed, a synchronized pseudorandom number can be generated for each packet. This is then used to determine which data symbols make up the parity symbols.

For example, in Fig. 2, we have used memory $W = 5$ and degree $D = 3$ to produce one parity symbol. It is possible to extend the coding idea to produce more parity symbols that are created by different random linear combinations; with $p - 1$ parity symbols, the systematic would have a code rate of $R = 1/p$. It is also possible to create other fractional code rates by splitting the data into l segments per packet. If these are used to create m parity symbols per packet, the overall code rate would be $l/(l + m)$.

IV. Simulations and Results

In this section, we present the performance of the application layer coding scheme with respect to memory size, latency, and code rate.

A. Memory Size

The size of the memory W will clearly have an effect on both performance and complexity. The end devices will need to keep the last W data symbols to produce the parity symbols, but at the same time, larger memory offers a higher probability that a received parity symbol (column of the generator matrix) is linearly independent of the other symbols. As a performance measure, we will use the DRR [1], which is defined as

$$\text{DRR} \triangleq \frac{\text{Number of recovered data units}}{\text{Number of transmitted data units}}. \quad (2)$$

In Fig. 3, the DRR is shown for a few memory sizes; clearly there are diminishing returns and very little is gained by using excessively large memories. The parity density, defined as

$$\Delta = \frac{D}{W} \quad (3)$$

is set to $\Delta = 0.5$. In the next section, we will discuss its impact.

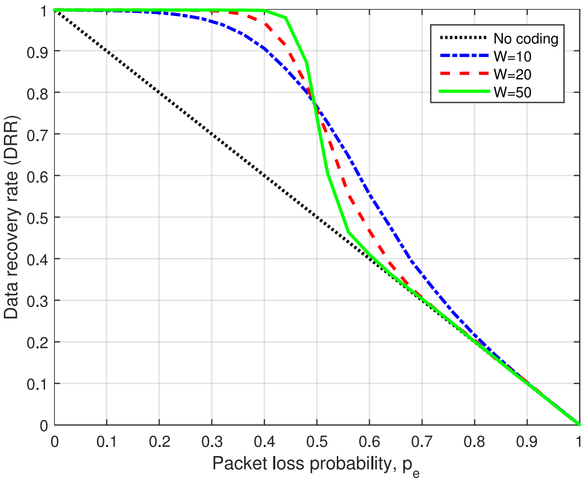

Fig. 3. Data recovery rate as a function of packet loss probability ($R = 1/2$, $\Delta = 0.5$).

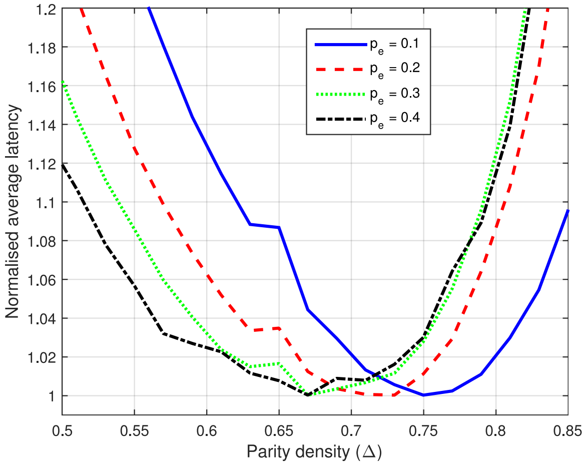

Fig. 4. Average normalized latency of recovered data (rate $R = 1/2$, memory $W = 50$).

B. Latency

The most important metric is, of course, the DRR, as it reflects the number of symbols recovered through the coding. However, this does not take into account another important aspect: the latency. While coding can recover lost data symbols, these could be quite old by the time they are decoded. If the data are time-sensitive, the recovery might be unnecessary. While the parity density Δ does not play a major part for the DRR, it does have an impact on the latency. The range of Δ for optimum DRR was shown in [1] to be quite large; however, when adding the latency metric, we can show that there are optimal values. In Fig. 4, the normalized[2] average latency is shown as a function of the

[2] The latency is normalized by its minimal value since the absolute value varies with the packet loss p_e, which would make the curves difficult to compare.

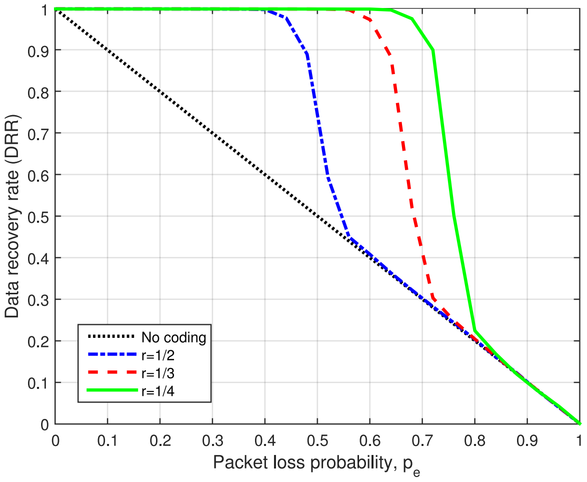

Fig. 5. Data recovery rate as a function of packet loss probability for different code rates ($W = 50, \Delta = 0.5$).

density Δ. The average latency is only measured over recovered (decoded) data symbols; data symbols that are received over the channel (and hence have latency zero) are not included. The optimal value of Δ depends on the packet loss rate p_e of the channel but a choice of $\Delta \approx 0.7$ works for most cases. It is worth noting that the latency can increase substantially if the density is chosen too small.

C. Code Rate

As mentioned earlier, it is possible to use lower code rates than $R = 1/2$ to offer more protection. The parity symbols are independently generated with the same memory and density (individual values do not seem to offer any advantages). The DRR for three different code rates are shown in Fig. 5; as expected, the performance improves with the lower code rate (more redundancy).

The asymptotic behavior of the coding scheme can actually be predicted by using existing results from fountain code theory. Full recovery of the missing packets is possible when the generator matrix $\mathbf{G}$ has full rank. In order for the probability of having a full rank random $K \times N$ binary matrix to be at least $1 - \delta$, we must have $N \geq K + \log_2 \frac{1}{\delta}$ columns [19]. This excess amount of packets $N - K$ makes it increasingly likely that $\mathbf{G}$ has full rank. Assume the above-mentioned coding scheme with rate R and that n packets have been transmitted, each with $1/R$ symbols. With p_e denoting the packet loss probability, the expected number of received symbols is $(1 - p_e)n/R$. Hence, we need

$$(1 - p_e)\,\frac{n}{R} \geq n + \log_2 \frac{1}{\delta}$$
$$\Rightarrow R \leq \frac{n\,(1 - p_e)}{n - \log_2 \delta} \rightarrow 1 - p_e \qquad (4)$$

where the last expression is the limit as the number of packets grows. Hence, it is clear that as the packet loss probability increases, the code rate must be reduced to maintain the same probability of successful decoding.

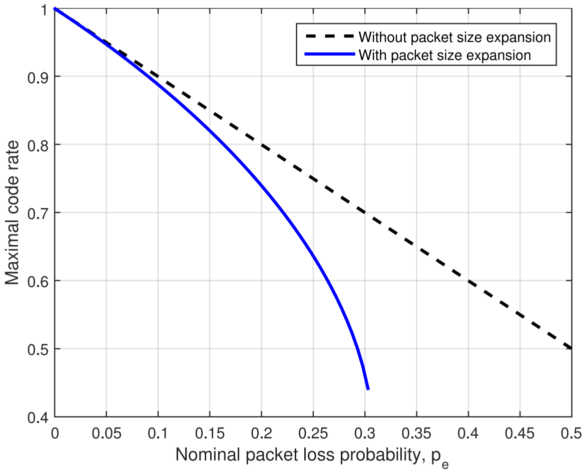

Fig. 6. Maximum effective code rate with and without packet size expansion.

D. System Performance Effects

Another aspect of the code rate is how the redundancy affects system performance. Since the payload of the packets is increased by a factor $1/R$, this could actually negatively influence the performance. For small payloads (relative to the overhead such as preamble, headers, etc.), it has marginal effects. However, if the payload is substantially larger than the overhead, this effectively makes the packet $1/R$ times larger. This, in turn, will increase the probability of packet collisions since it is more likely that two packets will overlap.

Consider a simplified model where there are l end devices and m slots in the time-frequency grid. A packet collision will occur if two or more packets occupy the same slot.[3] The probability that we have only one packet in a slot is then

$$\left(1 - \frac{1}{m}\right)^{l-1} \rightarrow e^{-(l-1)/m}, \quad l \gg 1 \qquad (5)$$

and consequently, the packet loss probability is

$$p_e \approx 1 - e^{-l/m}. \qquad (6)$$

By increasing the packet size by $1/R$ times, we effectively reduce the number of slots to mR. The packet loss probability now becomes

$$p'_e \approx 1 - e^{-l/mR} = 1 - (1 - p_e)^{1/R}. \qquad (7)$$

The condition for successful decoding (4) now becomes

$$R \leq 1 - p'_e(R) = (1 - p_e)^{1/R}. \qquad (8)$$

Hence, we can relate the nominal packet loss probability p_e to the maximum code rate when packet size expansion is taken into account. This means that the effective maximum code rate is

$$R_{\max} = \arg\max_{R} \left\{ R \,\middle|\, R < (1 - p_e)^{1/R} \right\}. \qquad (9)$$

This is shown in Fig. 6. It is worth noting that if the nominal packet loss probability exceeds $p_e > 0.3$, the coding introduces

[3] We ignore partial overlaps and relative signal strength. For a more in-depth outage analysis, see [7].

an unrecoverable increase in packet loss and no code rate exists that can successfully (on average) recover lost packets.

E. Observations

Based on the above-mentioned sections, a few things are worth highlighting. The parity density, which does not really affect complexity, can be used to optimize the latency although it has a very little impact on the DRR. Improving reliability by lowering the code rate only works up to a point; if the overhead in the packet is negligible compared to the data payload, then there is a nominal packet loss probability (≈ 0.3) beyond which reducing the code rate does not work. It simply makes the packets so large that packet collisions become too frequent for the application layer coding to work.

V. Decoding

Recovering the lost data symbols is possible if the generator matrix $\mathbf{G}$ has full rank, i.e., it is invertible. In this case, we can solve for the unknown symbols and recover all data symbols. In this section, we briefly discuss the principles and complexity of the different methods at our disposal. It is important to remember that all arithmetic is in $GF(2)$, which can sometimes simplify the job.

A. Optimal Decoding

Solving a system of linear equations can be done by Gaussian elimination [20]. For an $n \times n$ matrix, a straightforward implementation has complexity $\mathcal{O}\left(n^3\right)$; however, with optimized parallelized hardware, this can be brought down to $\mathcal{O}\left(n^2\right)$ [21]. For the systems described in this paper, it is important to note that due to the low duty cycle of the end devices, there might be quite sometime between packets. This can be used to process as much of the decoding matrix as possible, which alleviates the need for excessive processing when new parity symbols arrive. If fewer parity symbols than lost data symbols are available, decoding is not possible and the decoding matrix must be kept until new redundancy is obtained. By using elementary row and column operations [20], the decoding matrix $\mathbf{G}'$ of size s (missing data symbols) by t (linearly independent parity equations) can be arranged as

$$\mathbf{G}' = \begin{pmatrix} \mathbf{I}_t \\ \mathbf{A} \end{pmatrix} \tag{10}$$

where $\mathbf{I}_t$ is the $t \times t$ identity matrix and $\mathbf{A}$ is a $(s-t) \times t$ matrix. When a new parity equation is available through a received parity symbol, $\mathbf{G}'$ can easily be updated as

$$\mathbf{G}'' = \begin{pmatrix} \mathbf{I}_t & \\ \mathbf{A} & \mathbf{g} \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{I}_{t+1} \\ \mathbf{A}' \end{pmatrix}. \tag{11}$$

If the new vector $\mathbf{g}$ does not allow such a transformation, it is linearly dependent on the columns in $\mathbf{G}'$ and can be discarded. Decoding of symbols is now possible if any column of $\mathbf{G}''$ has only one nonzero element; this means that this equation has only one variable, and hence, it can be solved for. The trivial case is of course when the lower matrix $\mathbf{A}'$ is empty or all-zero, and thus, all $t+1$ variables can be solved. The rows and columns of $\mathbf{G}''$ corresponding to the solved variables and used equations, respectively, can be removed and the remaining matrix is kept for future decoding. Note that this "continuous" Gaussian elimination avoids duplicate operations and simplifies finding linearly dependent equations.

In the next section, we will discuss other methods to reduce complexity as well as a novel approach to the decoding problem in this paper.

B. Reduced Complexity Decoding

Fountain codes were initially designed to have linear encoding and decoding time [19]. This was achieved by using the LT decoder, which is a type of a *message passing* algorithm. Consider the following linear system of equations in $GF(2)$ [19]:

$$\begin{cases} x_1 = 1 \\ x_1 \bigoplus x_2 \bigoplus x_3 = 0 \\ x_2 \bigoplus x_3 = 1 \\ x_1 \bigoplus x_2 = 1. \end{cases} \tag{12}$$

Since the first equation only has one unknown, we solve this ($x_1 = 1$) and replace the variable with this value in the remaining equations as follows:

$$\begin{cases} x_2 \bigoplus x_3 = 1 \\ x_2 \bigoplus x_3 = 1 \\ x_2 = 0. \end{cases} \tag{13}$$

The last equation only has one variable, so solving this ($x_2 = 0$) and substituting it in the other equations give us

$$\begin{cases} x_3 = 1 \\ x_3 = 1. \end{cases} \tag{14}$$

The remaining equations now only have one variable, so we get $x_3 = 1$. Solving this system can clearly be done in linear time.

However, there is no guarantee that there will always be an equation with only one variable. If this does not happen, the decoding halts and no further decoding can take place (even if the system has a solution). Early work on fountain codes concerned designing the distribution of the chosen data symbols to form the parity symbols (nonzero values in a column of $\mathbf{G}$); this can be done to minimize the chances of the algorithm halting.

Another approach is that if the LT decoder halts, a particular variable can be solved. Wiedemann [22] designed a reduced complexity method for solving one (but not all) variables in a finite field. This was applied in [23] to the LT decoder to restart the algorithm whenever there were no more variables to solve directly. Although this increases the complexity, it was shown in [23] that it does not need to be applied very often with a carefully designed code.

A further modification of the message passing decoder is the *inactivation* method [24]. Instead of solving for one of the variables when the LT decoder halts, it is labeled inactive and assumed known. The decoder then continues until it halts again,

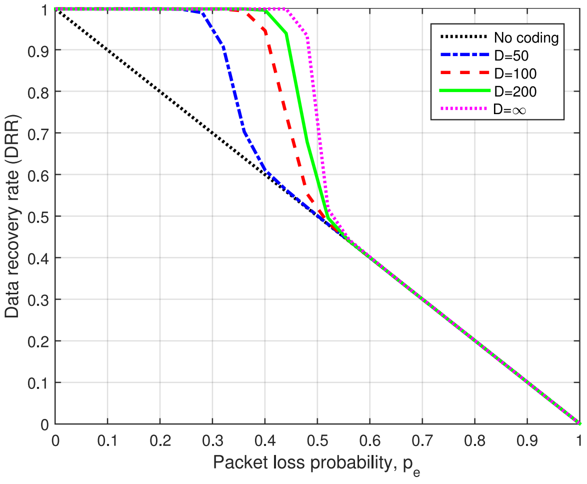

Fig. 7. DRR performance when a maximum decoding delay is introduced (rate $R = 1/2$, memory $W = 50$).

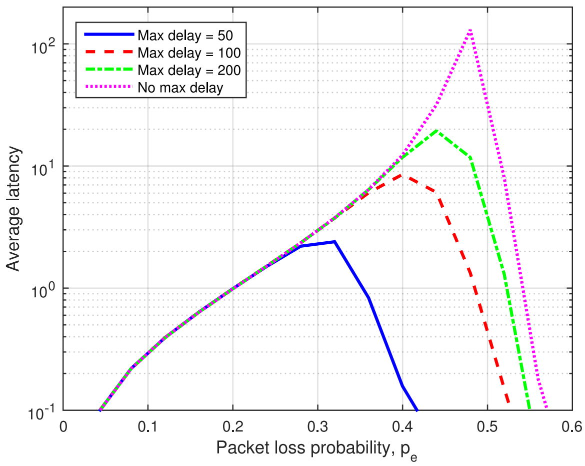

Fig. 8. Average latency of recovered data (rate $R = 1/2$, memory $W = 50$).

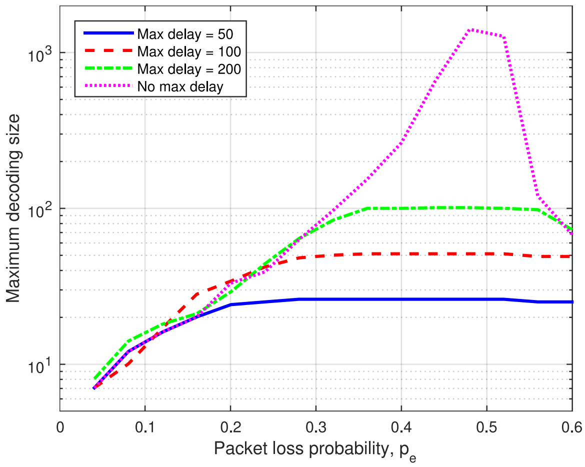

Fig. 9. Maximum size of the generator matrix for decoding (rate $R = 1/2$, memory $W = 50$).

when another variable is labeled inactive and assumed known. Eventually, all active variables are solved or rather, they are functions of the l inactive variables. These can be solved with Gaussian elimination (or similar methods) and back substituted into the other variables. The advantage is that a much smaller system needs to be solved of size $l \times l$, which can offer huge complexity reductions if $l \ll n$.

C. Suboptimal Decoding

All the methods described above find the solution, i.e., they are optimal in terms of performance. It is also possible to design suboptimal decoders that tradeoff performance for reduced complexity. The generator matrix is built up as more parity symbols are received. However, if some of the lost data symbols are very old, they could be discarded to reduce the size of the matrix to be inverted. This is done by removing the corresponding row and any columns with a one in this row. The latter step is necessary since we do not want parity symbols that depend on the discarded data symbol.

This novel complexity-reducing technique also has a practical side effect. If the end devices have time-sensitive data to send, it will not make sense to wait until a sufficient number of parity symbols have been received to allow decoding to start. Instead, these symbols can be discarded if they are too old as they are of little value. The downside is, of course, that the DRR is reduced, which is evident from Fig. 7.

The decoding delay is the age of a recovered data symbol in the decoding buffer; if symbol $n - d$ is decoded at time n, the delay is d. Note that this is measured in terms of packets. If the packets are sent on a regular basis, the actual delay is simply the d times the packet transmission interval. Otherwise, it must be measured using, e.g., time stamps. If a symbol in the decoding buffer has a delay that exceeds a predetermined value (it is too old to be valuable even if it is recovered), it is discarded and the decoding matrix is pruned as described above. Despite the loss in the DRR, it will have benefits in terms of latency and complexity. This is shown in Fig. 8, where the average latency can be limited by truncating the decoding process. As the maximum allowed delay is increased, the average latency is also increased. It is worth noting that the average latency drops as the packet loss increases; this is simply due to the fact that more symbols cannot be decoded and hence do not contribute toward the latency. For very high packet losses (above the code rate), the latency goes down to zero since no lost symbols can be recovered (only the ones received over the channel are available).

This also has a beneficial impact on the complexity, which is shown in Fig. 9; the complexity here is defined as the average size of the decoding buffer (number of symbols yet to decode). Regardless of decoder choice, this is an indication of the computational (and memory) burden of the data recovery. As can be seen, the novel scheme reduces the complexity as it limits the size of the decoding problem; this might have significant advantages when it comes to implementation.

The different decoders and their properties are compared in Table I, where we have listed their relative performance, complexity, and latency.

TABLE I
COMPARISON OF DIFFERENT DECODERS

Decoder	Performance	Complexity	Latency
Gaussian elimination	Optimal	High	Medium
LT-W [23]	Optimal	Medium	Medium
Inactivation [24]	Optimal	Medium	Medium
Message passing [14]	Suboptimal	Low	High
Truncation (Section V-C)	Suboptimal	Medium	Low

VI. CONCLUSION

In this paper, we have analyzed the application layer coding scheme for LPWA networks introduced in [1]. We have extended their study to include latency and the effects of decreased code rates as well as decoder complexity. The latency was shown to increase exponentially with the packet loss rate but a novel decoding scheme can reduce this with a small loss in data recovery. This new scheme can also limit the decoding complexity and memory requirements; a quantitative comparison between different decoding options was also given. We also showed that increased packet loss cannot be solely combated by introducing more redundancy; at some point, the increased packet size will cause an irreparable number of packet collisions.

ACKNOWLEDGMENT

The authors would like to acknowledge the fruitful discussions with their colleagues at Toshiba Research Europe and the support of its directors.

REFERENCES

[1] P. Marcelis, V. Rao, and R. Prasad, "DaRe: Data recovery through application layer coding for LoRaWAN," in *Proc. 2017 IEEE/ACM 2nd Int. Conf. Internet-of-Things Des. Implementation*, Pittsburgh, PA, USA, Apr. 2017, pp. 97–108.

[2] N. Sornin, M. Luis, T. Eirich, and T. Kramp, "LoRaWAN specification," LoRa Alliance, Beaverton, OR, USA, Tech. Rep., Jan. 2015. [Online]. Available: https://www.lora-alliance.org/portals/0/specs/LoRaWANSpecification1R0.p df

[3] "SIGFOX." [Online]. Available: http://www.sigfox.com/

[4] "RPMA technology for the internet of things," Ingenu, San Diego, CA, USA, Tech. Rep., 2016. [Online]. Available: http://theinternetofthings.report/Resources/Whitepapers/4cbc5e5e-6ef8-4 455-b8cd-f6e3888624cb_RPMATechnology.pdf

[5] D. Flore, "3GPP standards for the internet-of-things," gSMA MIoT, Feb. 2016. [Online]. Available: http://www.3gpp.org/news-events/3gpp-news/1766-iot_progress

[6] WAVIoT NB-FI LPWAN technology: Products and tech description," WAVIoT, Houston, TX, USA, Tech. Rep., Jun. 2016, release 1.8. [Online]. Available: http://waviot.com/wp.pdf

[7] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?" *IEEE Commun. Lett.*, vol. 6, no. 2, pp. 162–165, Apr. 2017.

[8] L. Krupka, L. Vojtech, and M. Neruda, "The issue of LPWAN technology coexistence in IoT environment," in *Proc. Int. Conf. Mechatron.*, Dec. 2016, pp. 1–8.

[9] D. Bankov, E. Khorov, and A. Lyakhov, "On the limits of LoRaWAN channel access," in *Proc. Int. Conf. Eng. Telecommun.*, Nov. 2016, pp. 10–14.

[10] F. Adelantado, X. Vilajosana, P. Tuset, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, 2017.

[11] A.-I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, "Does bidirectional traffic do more harm than good in LoRaWAN based LPWA networks?" in *Proc. IEEE GLOBECOM*, Singapore, Dec. 2017.

[12] "*IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)–Amendment 5: Physical Layer Specifications for Low Energy, Critical Infrastructure Monitoring Networks*," IEEE Std. 802.15.4k-2013 (Amendment to IEEE Std. 802.15.4-2011 as amended by IEEE Std. 802.15.4e-2012, IEEE Std. 802.15.4f-2012, IEEE Std. 802.15.4g-2012, and IEEE Std. 802.15.4j-2013), Aug. 2013, pp. 1–149.

[13] 'Weightless." [Online]. Available: http://www.weightless.org/

[14] M. Luby, "LT codes," in *Proc. IEEE Symp. Found. Comput. Sci.*, Vancouver, BC, Canada, Nov. 2002, pp. 271–280.

[15] 3GPP TS 25.346, "Technical specification group radio access network: Introduction of the multimedia broadcast/multicast service (MBMS) in the radio access network (RAN)," 3rd Generation Partnership Project, Mar. 2017. [Online]. Available: http://www.3gpp.org/ftp/Specs/2017-03/Rel-14/25_series/25.356-e00.zip

[16] ETSI EN 302 304, "Digital video broadcasting (DVB): Transmission system for handheld terminals (DVB-H)," 3rd Generation Partnership Project, Nov. 2004. [Online]. Available: http://www.etsi.org/deliver/etsi_en/302300_302399/302304/01.01.01_60/en _302304v010101p.pdf

[17] ETSI TS 102 034, "Digital video broadcasting (DVB): Transport of MPEG-2 TS based DVB service over IP based network," 3rd Generation Partnership Project, Apr. 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_ts/102000_102099/102034/02.01.01_60/ts _102034v020101p.pdf

[18] C. Studholme and I. Blake, "Windowed erasure codes," in *Proc. Int. Symp. Inf. Theory*, Toronto, ON, Canada, Jul. 2006, pp. 509–513.

[19] D. MacKay, "Fountain codes," *IEE Proc. Commun.*, vol. 152, no. 6, pp. 1062–1068, Dec. 2005.

[20] G. Strang, *Linear Algebra and Its Applications*, 3rd ed. Orlando, FL, USA: Harcourt Brace Jovanovich, 1988.

[21] A. Bogdanov, M. C. Mertens, C. Paar, J. Pelzl, and A. Rupp, "A parallel hardware architecture for fast Gaussian elimination over GF(2)," in *Proc. IEEE Symp. Field-Program. Custom Comput. Mach.*, Apr. 2006, pp. 237–248.

[22] D. Wiedemann, "Solving sparse linear equations over finite fields," *IEEE Trans. Inf. Theory*, vol. 32, no. 1, pp. 54–62, Jan. 1986.

[23] H. Lu, F. Lu, J. Cai, and C. H. Foh, "LT-W: Improving LT decoding with Wiedemann solver," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7887–7897, Dec. 2013.

[24] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.

Magnus Sandell (SM'09) received the M.Sc. degree in electrical engineering and Ph.D. degree in signal processing from the Luleå University of Technology, Luleå, Sweden, in 1990 and 1996, respectively.

He joined Bell Labs, Lucent Technologies (U.K.) in 1997. Since 2002, he has been a Chief Research Fellow with Toshiba Research Europe Ltd., Bristol, U.K. His research interests include signal processing, digital communications theory, and coding for non-volatile memories.

Usman Raza received the Ph.D. degree in information and communication technologies from the University of Trento, Trento, Italy, in collaboration with the Bruno Kessler Foundation.

He is a Senior Research Engineer with Toshiba Research Europe Ltd., Bristol, U.K. Before joining Toshiba Research Europe Ltd., he was a Research Fellow with the University of Trento. His current research interests include cyber-physical systems, industrial wireless systems, low-power wide area networks, and energy neutral embedded networks.

Dr. Raza was the recipient of the Endeavour Research Fellowship of the University of New South Wales, Sydney, NSW, Australia, an offer of the William J. Fulbright Scholarship from the U.S. Department of State, and an NMF Gold Medal from the Lahore University of Management Sciences. Two of his papers received the Best Paper Award at IEEE venues.