Reduced-Redundancy Product Codes for Burst Error Correction

Ron M. Roth[*] Gadiel Seroussi[†]

Abstract

In a typical burst-error correction application of a product code of $n_{\mathrm{v}} \times n_{\mathrm{h}}$ arrays, one uses an $[n_{\mathrm{h}}, n_{\mathrm{h}}-r_{\mathrm{h}}]$ code $\mathcal{C}_{\mathrm{h}}$ that detects corrupted rows, and an $[n_{\mathrm{v}}, n_{\mathrm{v}}-r_{\mathrm{v}}]$ code $\mathcal{C}_{\mathrm{v}}$ that is applied to the columns while regarding the detected corrupted rows as erasures. Although this conventional product code scheme offers very good error protection, it contains excessive redundancy, due to the fact that the code $\mathcal{C}_{\mathrm{h}}$ provides the code $\mathcal{C}_{\mathrm{v}}$ with information on many error patterns that exceed the correction capability of $\mathcal{C}_{\mathrm{v}}$. In this work, a coding scheme is proposed in which this excess redundancy is eliminated, resulting in significant savings in the overall redundancy compared to the conventional case, while offering the same error protection. The redundancy of the proposed scheme is $n_{\mathrm{h}} r_{\mathrm{v}} + r_{\mathrm{h}}(\ln r_{\mathrm{v}} + O(1)) + r_{\mathrm{v}}$, where the parameters r_{h} and r_{v} are close in value to their counterparts in the conventional case, which has redundancy $n_{\mathrm{h}} r_{\mathrm{v}} + n_{\mathrm{v}} r_{\mathrm{h}} - r_{\mathrm{h}} r_{\mathrm{v}}$. In particular, when the codes $\mathcal{C}_{\mathrm{h}}$ and $\mathcal{C}_{\mathrm{v}}$ have the same rate and $r_{\mathrm{h}} \ll n_{\mathrm{h}}$, the redundancy of the proposed scheme is close to one half of that of the conventional product code counterpart. Variants of the scheme are presented for channels that are mostly bursty, and for channels with a combination of random errors and burst errors.

Keywords: array codes, generalized concatenated codes, product codes, superimposed codes.

[*]Computer Systems Laboratory, Hewlett-Packard Laboratories, Palo Alto, California, and Hewlett-Packard Israel Science Center, Haifa, Israel. On sabbatical leave from the Computer Science Department, Technion, Haifa 32000, Israel. e-mail: ronny@cs.technion.ac.il.

[†]Computer Systems Laboratory, Hewlett-Packard Laboratories, Palo Alto, California. e-mail: seroussi@hpl.hp.com.

1 Introduction

Product codes [6][13] are a popular choice of error correction mechanism in magnetic recording due to their ability to offer good protection against both random and burst errors. Figure 1 depicts a typical $n_{\mathrm{v}} \times n_{\mathrm{h}}$ array Γ over a field $F = GF(q)$ which is encoded by a product code consisting of two codes: an $[n_{\mathrm{h}}, k_{\mathrm{h}}{=}n_{\mathrm{h}}{-}r_{\mathrm{h}}, d_{\mathrm{h}}]$ row code $\mathcal{C}_{\mathrm{h}}$ over F and an $[n_{\mathrm{v}}, k_{\mathrm{v}}{=}n_{\mathrm{v}}{-}r_{\mathrm{v}}, d_{\mathrm{v}}]$ column code $\mathcal{C}_{\mathrm{v}}$ over F. Hereafter we will refer to this product-code construction as *Construction 0*. The overall redundancy of Construction 0 is given by

$$n_{\mathrm{h}} r_{\mathrm{v}} + n_{\mathrm{v}} r_{\mathrm{h}} - r_{\mathrm{h}} r_{\mathrm{v}}. \tag{1}$$

In many applications, the codes $\mathcal{C}_{\mathrm{h}}$ and $\mathcal{C}_{\mathrm{v}}$ are taken to be maximum-distance separable (MDS) codes such as Reed-Solomon (RS) codes, in which case $d_{\mathrm{h}} = r_{\mathrm{h}}{+}1$ and $d_{\mathrm{v}} = r_{\mathrm{v}}{+}1$. This requires having code lengths n_{h} and n_{v} which do not exceed $q{+}1$, a condition that is met in practice in cases where the codes are naturally symbol- (e.g., byte-) oriented, and where burst correction is a major objective. Therefore, we will assume throughout this work that the codes used are MDS.

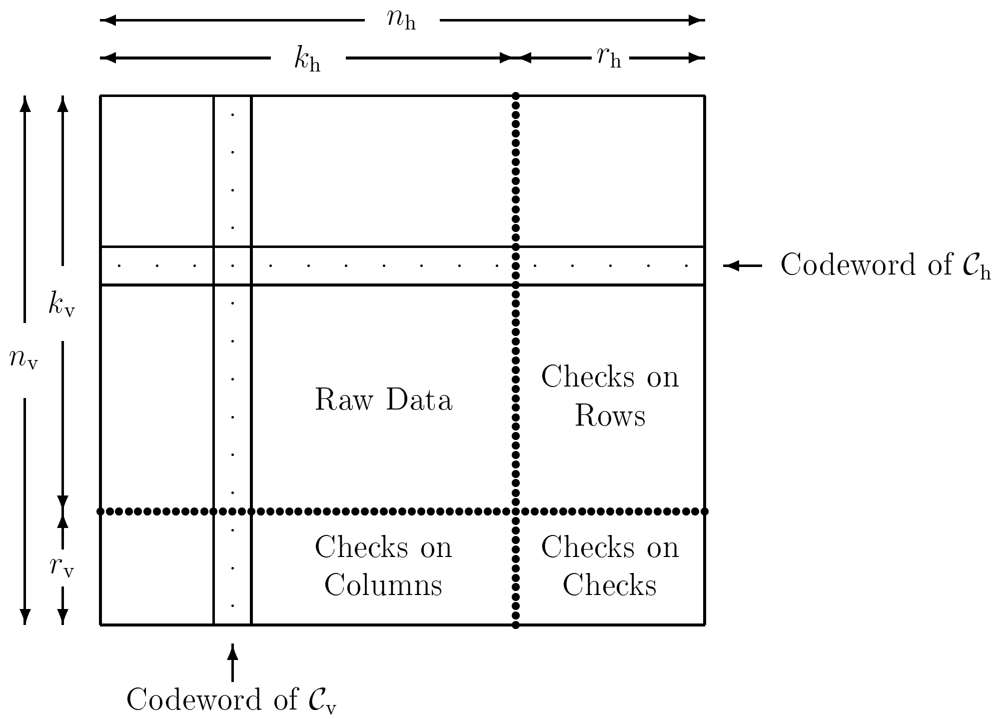

Figure 1: Array Γ of a product code.

In our model of error *values*, we will assume that entries in a transmitted array Γ are *affected*, resulting in an array $\tilde{\Gamma}$. An affected entry is replaced by a value of $GF(q)$ which is uniformly distributed over the elements of $GF(q)$, independently of the original contents of Γ or of the other error values. Such a model is approximated in practice through the use of scramblers. Notice that, in particular, an affected entry may still keep the correct value with probability $1/q$. If its value has been changed, we say that this entry is *corrupted*. The *error array* is defined by $E = \tilde{\Gamma} - \Gamma$.

The error *patterns* that will be considered in this work are mainly *burst errors* [13]. Assuming that the encoded array Γ is transmitted row by row, then, by the nature of burst errors, we expect the affected entries in the received array $\tilde{\Gamma}$ to be confined to a number T of rows, where T is governed by some probability measure $\mathsf{Prob}\{\, T = t \,\}$ which depends on the channel and on the choice of n_{h} and n_{v} (see below). An affected (respectively, corrupted) row in $\tilde{\Gamma}$ is a row that contains at least one affected (respectively, corrupted) entry. With the exception of Section 6, we will not assume any particular model on the patterns of affected entries within an affected row. If the ith row in $\tilde{\Gamma}$ has been affected, then the respective *error vector* is given by the ith row of the error array E. An error vector is nonzero if and only if the respective row in $\tilde{\Gamma}$ has been corrupted.

A typical burst decoding strategy for Construction 0 is as follows: The code $\mathcal{C}_{\mathrm{h}}$ is first used to *detect* the corrupted rows in a way that we describe shortly. Having found the corrupted rows, the decoder of $\mathcal{C}_{\mathrm{v}}$ is applied column by column, now regarding the corrupted entries in each given column as *erasures*. If p is the acceptable probability of array miscorrection, we will allocate half (say) of this probability to the event that the number of errors exceeds the correction capability of $\mathcal{C}_{\mathrm{v}}$. In particular, if $\mathcal{C}_{\mathrm{v}}$ is an MDS code, then r_{v} can be taken so that

$$\mathsf{Prob}\{\, T > r_{\mathrm{v}} \,\} \;\leq\; p/2 \;. \tag{2}$$

This guarantees that the erasure correction capability of $\mathcal{C}_{\mathrm{v}}$ is acceptable. (In fact, condition (2) can be slightly relaxed, since it is sufficient to require that, with probability $\geq 1-(p/2)$, the number of corrupted — as opposed to affected — rows in $\tilde{\Gamma}$ does not exceed r_{v}.)

Now, the code $\mathcal{C}_{\mathrm{h}}$ detects the corrupted rows by computing, for each row, its syndrome with respect to $\mathcal{C}_{\mathrm{h}}$. Let ℓ be the number of affected entries in a given affected row. If $\ell < d_{\mathrm{h}} = r_{\mathrm{h}}{+}1$, then the computed syndrome for that row must be nonzero in case the row is corrupted. Otherwise, suppose that $\ell > r_{\mathrm{h}}$ for a given affected row. Since every r_{h} columns in any $r_{\mathrm{h}} \times n_{\mathrm{h}}$ parity-check matrix of $\mathcal{C}_{\mathrm{h}}$ are linearly independent (by virtue of $\mathcal{C}_{\mathrm{h}}$ being MDS), the probability that such an affected row has an all-zero syndrome is $q^{-r_{\mathrm{h}}}$ (furthermore, the probability that such a row has got corrupted in addition to having an all-zero syndrome is $q^{-r_{\mathrm{h}}} - q^{-\ell} < q^{-r_{\mathrm{h}}}$). Therefore, regardless of the number of affected entries in a corrupted row, the probability of misdetecting a corrupted row is less than $q^{-r_{\mathrm{h}}}$. It follows that the probability that a row in a given array is both corrupted and misdetected is less than $\sum_t \mathsf{Prob}\{T = t\} \cdot t \cdot q^{-r_{\mathrm{h}}} = \tau q^{-r_{\mathrm{h}}}$, where τ stands for the expected value $\mathsf{E}_T\{T\}$. In fact, since we assume that (2) holds, then it is sufficient to require that r_{h} is such that

$$\sum_t \mathsf{Prob}\{\, T = t \mid T \leq r_{\mathrm{v}} \,\} \cdot t \cdot q^{-r_{\mathrm{h}}} \;\leq\; p/2 \;,$$

or

$$\tau(r_{\mathrm{v}}) \cdot q^{-r_{\mathrm{h}}} \;\le\; p/2 \;, \tag{3}$$

where $\tau(r) = \mathsf{E}_T\{\, T \,|\, T \le r \,\}$ (and where we assume that r_{h} does not exceed n_{h}).

We point out that the choice of r_{h} through (3) is rather conservative (and therefore robust) in the sense that we require that the overall probability of misdetecting a row will be not greater than $p/2$. For instance, in the event that the number of affected rows T is much smaller than r_{v}, we could in fact allow the decoder of $\mathcal{C}_{\mathrm{h}}$ to misdetect some of the corrupted rows and take advantage of the remaining $r_{\mathrm{v}} - T$ redundancy symbols (in excess of T) in $\mathcal{C}_{\mathrm{v}}$ to locate the misdetected corrupted rows. Such tuning, however, will depend much more substantially on the behavior of the probability measure $\mathsf{Prob}\{\, T = t \,\}$, whereas (2) and (3) depend only on the (conditional) expected value of T and the point where the tail probability drops below $p/2$. Indeed, in Appendix B we demonstrate how a finer tuning of the parameters can be made through a more extensive dependence on the probability measure $\mathsf{Prob}\{\, T = t \,\}$. The conservative approach, however, is warranted in many practical applications where the characterization of the channel statistics is often rather poor.

Many variations on the decoding strategy of Construction 0 are possible, offering a trade-off between random and burst error correction. The considerations for determining the values of n_{h} and n_{v} in the burst model case are roughly as follows. On the one hand, we would like n_{h} to be as small as possible so that the number of entries that will be marked as erased by the decoder of $\mathcal{C}_{\mathrm{h}}$ will be close to the number of entries that are affected by the bursts. On the other hand, we would like n_{h} to be large enough so that the ratio $r_{\mathrm{h}}/n_{\mathrm{h}}$ — and hence the relative redundancy — be as small as possible. Also, n_{v} — and therefore r_{v} — must be small enough so that, by the law of large numbers, we will be able to maintain a sufficiently small value for the ratio $r_{\mathrm{v}}/n_{\mathrm{v}}$ while still satisfying (2). This however makes the decoder of $\mathcal{C}_{\mathrm{v}}$ more complex, as it needs to be able to correct more erasures. An upper bound on $n_{\mathrm{h}} n_{\mathrm{v}}$ is dictated by the amount of memory and latency that we can afford.

In this work, we observe that although Construction 0 offers very good error protection, it contains excessive redundancy, due to the fact that the "inner" code $\mathcal{C}_{\mathrm{h}}$ provides the "outer" code $\mathcal{C}_{\mathrm{v}}$ with information on many error patterns that exceed the correction capability of $\mathcal{C}_{\mathrm{v}}$. More specifically, we allocate redundancy r_{h} of $\mathcal{C}_{\mathrm{h}}$ for *each row* of the array to determine whether the row is corrupted. This way, the decoder of $\mathcal{C}_{\mathrm{h}}$ can inform the decoder of $\mathcal{C}_{\mathrm{v}}$ about *any* combination of up to n_{v} corrupted rows. However, the code $\mathcal{C}_{\mathrm{v}}$ can correct only up to r_{v} erased locations, namely, it can only handle up to r_{v} corrupted rows. Any information about combinations of $r_{\mathrm{v}}{+}1$ corrupted rows or more is therefore useless for $\mathcal{C}_{\mathrm{v}}$. Nevertheless, we are paying in redundancy to provide this information. A coding scheme where this excess redundancy is eliminated is presented in Sections 2 and 3. Section 2 presents a basic construction, referred to as *Construction 1*, that illustrates the key ideas and achieves most of the redundancy reduction, while Section 3 presents a more refined construction, referred to as *Construction 2*, that attains further redundancy gains through the use of codes with varying rates.

In the early work by Kasahara et *al.* [11], they suggested an improvement on Construction 0 by a technique called *superimposition*. The objective in [11] was increasing the code dimension while maintaining the *minimum Hamming distance* of the code. The same motivation also led to the introduction of generalized concatenated codes by Blokh and Zyablov in [4]. In those codes, the savings in the overall redundancy were obtained by using inner and outer codes with varying rates. Generalized concatenated codes were further studied by Zinoviev [18], and Zinoviev and Zyablov [19], [20], where the latter paper also considered minimum-distance decoding of combined random and burst errors. Hirasawa et *al.* [7],[8] presented a similar construction which was shown to increase the code rate while maintaining the miscorrection probability of *random errors*. For related work, see also [12] and [16].

Our main objective in this paper is to increase the code dimension while maintaining the *miscorrection probability of bursts* (we do consider also a more general setting in *Construction 3* of Section 6 that includes combined burst and random errors). Our constructions differ significantly from that of Kasahara et *al.* [11] in the *decoding* mechanism (which we present in Section 4), although the schemes do bear some resemblance in their *encoding* mechanisms (our encoder is presented in Section 5). However, the different objective allows us to obtain a more substantial improvement on the code dimension over Construction 0 compared to the construction in [11]. Most aspects of our constructions also differ from those of Blokh and Zyablov [4] and Hirasawa et *al.* [7],[8]. Still, it is worth pointing out a feature which appears both in those construction and Construction 2, namely, that of using a sequence of codes of varying rates rather than a unique code — thereby increasing the overall code rate while maintaining the miscorrection probability.

We also mention here the recent work [17], where the model of crisscross errors is studied. That model is more general than the one we discuss here; however the construction for crisscross errors requires more redundancy.

2 Simple construction with reduced redundancy (Construction 1)

Let $\mathcal{C}_{\mathrm{h}}$ and $\mathcal{C}_{\mathrm{v}}$ be the MDS codes over $F = GF(q)$ which are used in Construction 0 and let $[n_{\mathrm{h}}, k_{\mathrm{h}}{=}n_{\mathrm{h}}{-}r_{\mathrm{h}}, d_{\mathrm{h}}{=}r_{\mathrm{h}}{+}1]$ and $[n_{\mathrm{v}}, k_{\mathrm{v}}{=}n_{\mathrm{v}}{-}r_{\mathrm{v}}, d_{\mathrm{v}}{=}r_{\mathrm{v}}{+}1]$ be their respective parameters. Also, let H_{h} be an $r_{\mathrm{h}} \times n_{\mathrm{h}}$ parity-check matrix of $\mathcal{C}_{\mathrm{h}}$.

Let Γ be an array which consists of n_{h} columns, each being a codeword of $\mathcal{C}_{\mathrm{v}}$. Unlike Construction 0, we do not assume at this point that the rows of Γ belong to any specific code. For each row of Γ, we compute its syndrome with respect to the parity-check matrix H_{h}, thus obtaining an $n_{\mathrm{v}} \times r_{\mathrm{h}}$ *syndrome array* $S = S(\Gamma)$; that is

$$S = \Gamma H_{\mathrm{h}}' .$$

(Note that each row in S can take arbitrary values in $F^{r_{\mathrm{h}}}$.)

Now, suppose that Γ is transmitted through a noisy channel, resulting in a (possibly corrupted) array $\tilde{\Gamma} = \Gamma + E$ at the receiving end. Let $\tilde{S}$ be the syndrome array $\tilde{\Gamma} H'_{\mathrm{h}}$ corresponding to the received array (see Figure 2). We compare $\tilde{S}$ to the syndrome array $S = S(\Gamma)$ for

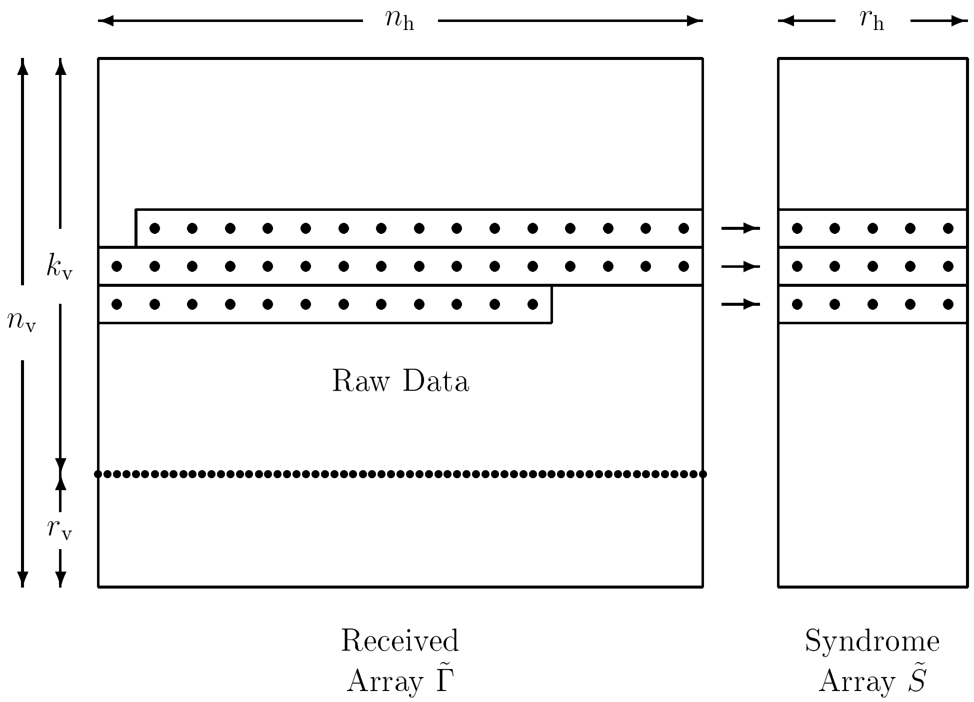

Figure 2: Syndrome array for received array.

the transmitted array Γ.[1] More specifically, we consider the *differential syndrome array*

$$\Delta S \;=\; \tilde{S} - S \;=\; E H'_{\mathrm{h}} \; .$$

The following two observations can be made:

- If a given row in ΔS is nonzero (namely, if the contents of a given row in $\tilde{S}$ has changed compared to that row in S), then the respective row in Γ has been corrupted.
- If a given row in ΔS is all-zero, then the probability that the respective row in Γ has been corrupted is less than $q^{-r_{\mathrm{h}}}$.

Hence, if condition (3) holds, then, with probability $\geq 1 - (p/2)$, the nonzero rows of ΔS point at all the corrupted rows in $\tilde{\Gamma}$.

[1] For the time being, this "comparison" is only conceptual, as the original syndrome array S is *not* sent along with the transmitted array Γ, and is not available on the decoding side.

Let $S_0, S_1, \ldots, S_{\mathrm{h}-1}$ denote the columns of $S = S(\Gamma)$. In order to allow the receiver to locate the nonzero rows in $\Delta S = \tilde{S} - S$, the transmitter encodes the raw data in Γ so that each column vector S_j in the resulting syndrome array $S = S(\Gamma)$ is a codeword of an $[n_{\mathrm{v}}, n_{\mathrm{v}}{-}r_j, r_j{+}1]$ MDS code $\mathcal{C}_j$ over $GF(q)$. The choice of the redundancy values r_j should allow the receiver to locate the nonzero rows in ΔS out of $\tilde{S}$, with an acceptably small probability of failure.

A simple choice for $\mathcal{C}_j$, which we assume for the remainder of the section, would be setting $r_j = 2r_{\mathrm{v}}$ for each j, where r_{v} satisfies condition (2). This condition implies that with probability $\geq 1 - (p/2)$, the number of nonzero rows in ΔS will not exceed r_{v}. Therefore, in this case, by decoding the columns of $\tilde{S}$, the decoders of $\mathcal{C}_j$ can locate the nonzero rows of ΔS and, thus, the corrupted rows of Γ with probability $> 1 - \tau q^{-r_{\mathrm{h}}} \geq 1 - (p/2)$.

Now, each column vector S_j is obtained as a linear combination of the columns of Γ. Each column of Γ, in turn, is a codeword of $\mathcal{C}_{\mathrm{v}}$. Since the code $\mathcal{C}_{\mathrm{v}}$ is linear, it follows that each column S_j is a codeword of the MDS code $\mathcal{C}_{\mathrm{v}}$ whose redundancy is r_{v}. However, we require that S_j belong to an MDS code $\mathcal{C}_j$ with redundancy $r_j = 2r_{\mathrm{v}}$, making the overall redundancy in S equal to $2r_{\mathrm{h}}r_{\mathrm{v}}$. If we choose each $\mathcal{C}_j$ to be a *subcode* of $\mathcal{C}_{\mathrm{v}}$, then we can fully exploit the redundancy inherited from $\mathcal{C}_{\mathrm{v}}$ due to linearity. The required additional redundancy of $r_{\mathrm{h}}r_{\mathrm{v}}$ in S will be achieved by imposing $r_{\mathrm{h}}r_{\mathrm{v}}$ additional linear constraints on the encoded array Γ. We refer to the resulting scheme as Construction 1, and from the above discussion, we readily obtain the following.

Proposition 1. *The redundancy of Construction 1 is*

$$n_{\mathrm{h}}r_{\mathrm{v}} + r_{\mathrm{h}}r_{\mathrm{v}} \ . \tag{4}$$

Hence, the redundancy of Construction 1 compares very favorably with (1) when $r_{\mathrm{v}} \ll n_{\mathrm{v}}$, as is usually the case in practical applications. In particular, when $r_{\mathrm{h}}/n_{\mathrm{h}} = r_{\mathrm{v}}/n_{\mathrm{v}}$ and $r_{\mathrm{h}}, r_{\mathrm{v}} \ll n_{\mathrm{h}}, n_{\mathrm{v}}$, the reduction in the redundancy is close to a factor of 2 compared to Construction 0.

3 Further redundancy reduction (Construction 2)

Additional savings in redundancy can be achieved by observing that for each $1 \leq j < r_{\mathrm{h}}$, the decoder of $\mathcal{C}_j$ can obtain *erasure* information from columns of $\tilde{S}$ that have already been decoded. This will result in a coding scheme which will be referred to as Construction 2. As we show next, the overall redundancy of Construction 2 is at most $n_{\mathrm{h}}r_{\mathrm{v}}{+}r_{\mathrm{h}}(\ln r_{\mathrm{v}}{+}O(1)){+}r_{\mathrm{v}}$, where the parameters r_{h} and r_{v} are close in value to their counterparts in Construction 0.

In Construction 2, we encode the array Γ as before, so that each column is a codeword of the $[n_{\mathrm{v}}, n_{\mathrm{v}}{-}r_{\mathrm{v}}, r_{\mathrm{v}}{+}1]$ code $\mathcal{C}_{\mathrm{v}}$ and, for $0 \leq j < r_{\mathrm{h}}$, each column vector S_j in $S = S(\Gamma)$ is a codeword of an $[n_{\mathrm{v}}, n_{\mathrm{v}} - r_j, r_j + 1]$ code $\mathcal{C}_j$. We will determine the parameters r_{v}, r_{h}, and r_j later on. Let $\tilde{S}_0, \tilde{S}_1, \ldots, \tilde{S}_{r_{\mathrm{h}}-1}$ denote the columns of the possibly corrupted syndrome array $\tilde{S}$. We assume that the receiver decodes those columns in a consecutive order, starting with $\tilde{S}_0$. Since the code $\mathcal{C}_0$ does not have any a priori erasure information, its redundancy will be set to $r_0 = 2r_{\mathrm{v}}$ in order to locate up to r_{v} errors in $\tilde{S}_0$.

As mentioned, if a row in $\Delta S = \tilde{S} - S$ is nonzero, then the respective row in $\tilde{\Gamma}$ is corrupted (furthermore, the converse holds with probability $> 1 - \tau q^{-r_{\mathrm{h}}}$). However, there may be nonzero rows in ΔS that are missed by the $\mathcal{C}_0$-decoder: These are the nonzero rows in ΔS whose leading entry (i.e., their entry in $\Delta S_0 = \tilde{S}_0 - S_0$) is zero. Nevertheless, with high probability (which we compute next), most of the nonzero rows in ΔS will be found by the $\mathcal{C}_0$-decoder when applied to $\tilde{S}_0$, and the locations of those rows can be passed to the decoders of $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{r_{\mathrm{h}}-1}$ as erasure information. These decoders, in turn, can locate nonzero rows in ΔS that were missed by the $\mathcal{C}_0$-decoder.

In general, for $1 \leq j < r_{\mathrm{h}}$, the $\mathcal{C}_{j-1}$-decoder will pass erasure information to the $\mathcal{C}_j$-decoder, thus allowing the reduction of the redundancy of $\mathcal{C}_j$ which is required in order to decode $\tilde{S}_j$. Ultimately, the erasure information passed by the $\mathcal{C}_{r_{\mathrm{h}}-1}$-decoder to the $\mathcal{C}_{\mathrm{v}}$-decoder will include (with an acceptably small probability of failure) the locations of all the corrupted rows in $\tilde{\Gamma}$, leaving the $\mathcal{C}_{\mathrm{v}}$-decoder with the task of decoding erasures only. For the sake of uniformity, it will be convenient to define $\mathcal{C}_j$ and r_j for $j = r_{\mathrm{h}}$ as $\mathcal{C}_{\mathrm{v}}$ and r_{v}, respectively. Thus, we will have a gradual transition from full error correction for $\mathcal{C}_0$, through combined error–erasure correction for $\mathcal{C}_j$, $1 \leq j < r_{\mathrm{h}}$, to pure erasure correction for $\mathcal{C}_{r_{\mathrm{h}}} = \mathcal{C}_{\mathrm{v}}$. The determination of the redundancies r_j, $0 \leq j \leq r_{\mathrm{h}}$, in Construction 2 is discussed next.

3.1 Setting the constraints on the code parameters

Let H_{h} be the parity-check matrix of $\mathcal{C}_{\mathrm{h}}$ used to compute S and, for $1 \leq j \leq r_{\mathrm{h}}$, denote by $H_{\mathrm{h}}^{[j]}$ the $j \times n_{\mathrm{h}}$ matrix which consists of the first j rows of H_{h}.

We say that H_{h} satisfies the *MDS supercode property* if, for $j = 1, 2, \ldots, r_{\mathrm{h}}$, each matrix $H_{\mathrm{h}}^{[j]}$ is a parity-check matrix of an MDS code. A code $\mathcal{C}_{\mathrm{h}}$ is said to satisfy the MDS supercode property if it has a parity-check matrix that satisfies the MDS supercode property. Examples of matrices that satisfy this property are $H_{\mathrm{h}} = [\,\alpha_\ell^k\,]_{k=0,\ell=0}^{r_{\mathrm{h}}-1,n_{\mathrm{h}}-1}$, where the α_ℓ are distinct elements of $GF(q)$; these are parity-check matrices of generalized RS codes. Notice that when $r_{\mathrm{h}} > 1$, every matrix that satisfies the MDS supercode property must be nonsystematic. Since every MDS code has a systematic parity-check matrix, it follows that codes that satisfy the MDS supercode property also have parity-check matrices that do not satisfy the property. We also point out that there are MDS codes, such as the $[q+1, q+1-r, r+1]$ (doubly-extended) RS codes with $r > 1$, that do not satisfy the MDS supercode property. We elaborate more on this in Appendix A.

We will assume in Construction 2 that H_{h} satisfies the MDS supercode property.

Consider the syndrome array $S = \Gamma H'_{\mathrm{h}}$. For $1 \le j \le r_{\mathrm{h}}$, denote by $S^{[j]}$ the matrix formed by the first j columns of S, i.e.,

$$S^{[j]} = [\, S_0 \; S_1 \; \ldots \; S_{j-1} \,] = \Gamma H_{\mathrm{h}}^{[j]\prime} .$$

We define $\tilde{S}^{[j]}$ in a similar manner and we let $\Delta S^{[j]}$ be $\tilde{S}^{[j]} - S^{[j]}$.

We say that a corrupted row in $\tilde{\Gamma}$ is *hidden* from $\tilde{S}^{[j]}$ if the corresponding row in $\Delta S^{[j]}$ is all-zero. For $1 \le j \le r_{\mathrm{h}}$, denote by X_j the random variable which equals the number of corrupted rows in $\tilde{\Gamma}$ that are hidden from $\tilde{S}^{[j]}$. We extend this definition to $j = 0$, letting X_0 denote the number of corrupted rows in $\tilde{\Gamma}$.

We will assume that for $1 \le j < r_{\mathrm{h}}$, each code $\mathcal{C}_j$ is a subcode of $\mathcal{C}_{\mathrm{v}}$ and so we can write $r_j = r_{\mathrm{v}} + a_j$ where $a_j \ge 0$. The overall redundancy of Construction 2 thus equals

$$r_{\mathrm{v}} n_{\mathrm{h}} + \sum_{j=0}^{r_{\mathrm{h}}-1} a_j \; . \tag{5}$$

We will also define $a_{r_{\mathrm{h}}} = 0$, in accordance with our previous convention that $\mathcal{C}_{r_{\mathrm{h}}} = \mathcal{C}_{\mathrm{v}}$.

We can now formulate our problem as follows: Given an acceptable probability p of miscorrection, find nonnegative integers $r_{\mathrm{v}}(\le n_{\mathrm{v}})$, $r_{\mathrm{h}}(\le n_{\mathrm{h}})$ and $a_0, a_1, \ldots, a_{r_{\mathrm{h}}-1}$ (and $a_{r_{\mathrm{h}}} = 0$) that

$$\text{minimize} \quad r_{\mathrm{v}} n_{\mathrm{h}} + \sum_{j=0}^{r_{\mathrm{h}}-1} a_j \; , \tag{6}$$

subject to the constraint

$$\mathsf{Prob}\Big\{ \bigcup_{j=0}^{r_{\mathrm{h}}} \{\, X_0 + X_j > r_{\mathrm{v}} + a_j \,\} \Big\} \le p \; . \tag{7}$$

The constraint (7) replaces conditions (2) and (3) and guarantees, with acceptable probability, that for each j, the number of errors, X_j, and erasures, $X_0 - X_j$, does not exceed the correction capability of the code $\mathcal{C}_j$ (i.e., the redundancy is at least $(X_0 - X_j) + 2X_j$).

In the sequel, we find an approximation to the solution of (6) and (7). By (7), we must have $\mathsf{Prob}\{\, X_0 + X_j > r_{\mathrm{v}} + a_j \,\} = O(p)$ for all j. In particular, for $j = r_{\mathrm{h}}$ we need to have $\mathsf{Prob}\{\, X_0 + X_{r_{\mathrm{h}}} > r_{\mathrm{v}} \,\} = O(p)$, which, in turn, implies that $\mathsf{Prob}\{\, X_0 > r_{\mathrm{v}} \,\} = O(p)$. On the other hand, for $j = 0$ we need to have $\mathsf{Prob}\{\, 2X_0 > r_{\mathrm{v}} + a_0 \,\} = O(p)$, which thus motivates us to choose $a_0 = r_{\mathrm{v}}$. This choice is also consistent with our strategy that the $\mathcal{C}_0$-decoder handles as many as r_{v} errors without having any a priori information on the locations of those errors.

Now,

$$\mathsf{Prob}\Big\{ \bigcup_{j=0}^{r_{\mathrm{h}}} \{\, X_0 + X_j > r_{\mathrm{v}} + a_j \,\} \Big\}$$

$$\leq \ \mathsf{Prob}\Big\{ \bigcup_{j=0}^{r_\mathrm{h}} \{ T + X_j > r_\mathrm{v} + a_j \} \ \Big| \ T \leq r_\mathrm{v} \Big\} \ + \ \mathsf{Prob}\{ T > r_\mathrm{v} \}$$

$$\leq \ \sum_{j=1}^{r_\mathrm{h}} \mathsf{Prob}\Big\{ T + X_j > r_\mathrm{v} + a_j \ \Big| \ T \leq r_\mathrm{v} \Big\} \ + \ \mathsf{Prob}\{ T > r_\mathrm{v} \} \ ,$$

where the second inequality follows from a union bound and the fact that $X_0 \leq T \leq r_\mathrm{v}$ given the conditioning event (and so the term that corresponds to $j = 0$ vanishes). It thus follows that (7) is implied by

$$\mathsf{Prob}\{ T > r_\mathrm{v} \} \ \leq \ p/2 \tag{8}$$

and

$$\sum_{j=1}^{r_\mathrm{h}} \mathsf{Prob}\Big\{ T + X_j > r_\mathrm{v} + a_j \ \Big| \ T \leq r_\mathrm{v} \Big\} \ \leq \ p/2 \ . \tag{9}$$

Satisfying the constraint (9) guarantees with acceptable probability that each $\mathcal{C}_j$-decoder will have enough redundancy to correct the number of full errors, X_j and erasures, $T - X_j$, that it will typically encounter.

The expressions $\mathsf{Prob}\{ T + X_j > r_\mathrm{v} + a_j \mid T \}$ will be bounded from above using Lemma 1 below.

Lemma 1. *For $1 \leq j \leq r_\mathrm{h}$ and every nonnegative integer b,*

$$\mathsf{Prob}\{ X_j > b \mid T \} \ \leq \ q^{-j(b+1)} \cdot \binom{T}{b+1} \ ,$$

where $\binom{t}{b+1} = 0$ if $b \geq t$.

Proof. Suppose that row i in $\tilde{\Gamma}$ contains at most j affected entries and let $\mathbf{e}$ be the respective error vector. Since H_h satisfies the MDS supercode property, $H_\mathrm{h}^{[j]}$ is the parity-check matrix of an $[n, n{-}j, j{+}1]$ code and, thus, row i in $\Delta S^{[j]}$ is all-zero if and only if $\mathbf{e} = \mathbf{0}$. Hence, if at least one of the (up to) j affected entries of row i in $\tilde{\Gamma}$ has been corrupted, then that row cannot be hidden from $\tilde{S}^{[j]}$.

Next consider the rows in $\tilde{\Gamma}$ that contain more than j affected entries, and let σ be the random variable which equals the number of those rows. For each such row, the respective row in $\Delta S^{[j]}$ will be zero with probability q^{-j}. Furthermore, the vector values of the rows in $\Delta S^{[j]}$ that correspond to distinct affected rows of $\tilde{\Gamma}$ are statistically independent. Hence, recalling that $\sigma \leq T$, we have,

$$\begin{aligned} \mathsf{Prob}\{ X_j > b \mid T, \sigma \} \ &\leq \ \sum_{z=b+1}^{\sigma} \binom{\sigma}{z} q^{-jz} (1 - q^{-j})^{\sigma - z} \\ &\leq \ \binom{\sigma}{b+1} q^{-j(b+1)} \ \leq \ \binom{T}{b+1} q^{-j(b+1)} \ , \end{aligned}$$

and, therefore,

$$\mathsf{Prob}\{\, X_j > b \mid T \,\} \;=\; \mathsf{E}_\sigma \Big\{ \mathsf{Prob}\{\, X_j > b \mid T, \sigma \,\} \Big\} \;\le\; \binom{T}{b+1} q^{-j(b+1)} \;,$$

as claimed. □

For nonnegative integers r, s, and ω, we define the quantity $\mathsf{B}_T(r,s,\omega)$ by

$$\mathsf{B}_T(r,s,\omega) \;=\; \mathsf{E}_T \left\{ \binom{T}{s+1-T} \cdot \omega^{T-r} \;\Big|\; T \le r \right\} \;. \tag{10}$$

Clearly, $\mathsf{B}_T(r,s,\omega) = 0$ when $s \ge 2r$. The following corollary follows readily from the definition of B_T and from Lemma 1.

Corollary 1. *For $1 \le j \le r_{\mathrm{h}}$ and every nonnegative integer a,*

$$\mathsf{Prob}\{\, T + X_j > r_{\mathrm{v}} + a \mid T \le r_{\mathrm{v}} \,\} \;\le\; \mathsf{B}_T(r_{\mathrm{v}}, r_{\mathrm{v}}{+}a, q^j) \cdot q^{-j(a+1)} \;.$$

By Corollary 1, we can replace the constraint (9) by the following stronger condition,

$$\sum_{1 \le j \le r_{\mathrm{h}} \,:\, a_j < r_{\mathrm{v}}} \mathsf{B}_T(r_{\mathrm{v}}, r_j, q^j) \cdot q^{-j(a_j+1)} \;\le\; p/2 \;, \tag{11}$$

where we recall that $r_j = r_{\mathrm{v}} + a_j$ and $a_{r_{\mathrm{h}}} = 0$. Notice that we have restricted the summation index set in (11) to those values of j for which $a_j < r_{\mathrm{v}}$, since $\mathsf{B}_T(r_{\mathrm{v}}, r_j, q^j) = 0$ otherwise. In fact, if $\mathsf{Prob}\{\, T = r_{\mathrm{v}} \,\} > 0$ (which is the case if r_{v} is the smallest integer that satisfies (8)), then we can also state conversely that $\mathsf{B}_T(r_{\mathrm{v}}, r_j, q^j) > 0$ whenever $a_j < r_{\mathrm{v}}$.

We next show a feasible solution for the a_j, satisfying the constraint (11). This solution will be the basis of Construction 2, since, together with the requirement (8), it will also satisfy constraint (7) and provide an approximation to (6).

3.2 Analysis of a feasible solution

For a nonnegative integer r, define $\beta_T(r)$ by

$$\beta_T(r) \;=\; q^{-r} \cdot \mathsf{E}_T\{\, q^T (2^T{-}1) \mid T \le r \,\} \;. \tag{12}$$

Theorem 1. *Given an acceptable probability p of miscorrection, let r_{v} be a positive integer (such as an integer that satisfies (8)) and let r_{h} be an integer such that*

$$\log_q \left(\frac{q}{q-1} \cdot \frac{\beta_T(r_{\mathrm{v}})}{p/2} \right) \;\le\; r_{\mathrm{h}} \;\le\; n_{\mathrm{h}} \tag{13}$$

(provided that such an integer $r_{\rm h}$ exists). Then the following values

$$a_j \;=\; \begin{cases} r_{\rm v} & \text{if } 0 \le j < r_{\rm h}/r_{\rm v} \\ \lceil r_{\rm h}/j \rceil - 1 & \text{if } r_{\rm h}/r_{\rm v} \le j \le r_{\rm h} \end{cases} \;, \qquad 0 \le j \le r_{\rm h} \;, \tag{14}$$

satisfy the constraint (11).

Proof. Let $j_1 < j_2 < \ldots < j_s$ be a sequence which consists of all indexes $0 < j \le r_{\rm h}$ such that $a_j \ne a_{j-1}$; note that $j_s = r_{\rm h}$, and define $j_{s+1} = j_s + 1$. Fix T to a value less than or equal to $r_{\rm v}$. For every $1 \le \ell \le s$ we have,

$$\begin{aligned}
\sum_{j=j_\ell}^{j_{\ell+1}-1} \binom{T}{r_j{+}1{-}T} \cdot q^{-j(r_j+1-T)} \;&=\; \binom{T}{r_{j_\ell}{+}1{-}T} \cdot \sum_{j=j_\ell}^{j_{\ell+1}-1} q^{-j(r_{j_\ell}+1-T)} \\
&\le\; \frac{q}{q{-}1} \cdot q^{-j_\ell(r_{j_\ell}+1-T)} \cdot \binom{T}{r_{j_\ell}{+}1{-}T} \\
&\le\; \frac{q}{q{-}1} \cdot q^{-r_{\rm h}} \cdot q^{j_\ell(T-r_{\rm v})} \cdot \binom{T}{r_{j_\ell}{+}1{-}T} \;, \qquad (15)
\end{aligned}$$

where (15) follows from (14), namely, $j_\ell(r_{j_\ell}+1) = j_\ell(a_{j_\ell}+1+r_{\rm v}) \ge r_{\rm h} + j_\ell r_{\rm v}$. Hence, for every fixed $T \le r_{\rm v}$ we have,

$$\begin{aligned}
\sum_{1\le j\le r_{\rm h}\,:\,a_j<r_{\rm v}} \binom{T}{r_j{+}1{-}T} \cdot q^{-j(r_j+1-T)} \;&=\; \sum_{\ell=1}^{s} \sum_{j=j_\ell}^{j_{\ell+1}-1} \binom{T}{r_j{+}1{-}T} \cdot q^{-j(r_j+1-T)} \\
&\overset{(15)}{\le}\; \frac{q}{q{-}1} \cdot q^{-r_{\rm h}} \cdot q^{j_1(T-r_{\rm v})} \cdot \sum_{\ell=1}^{s} \binom{T}{r_{j_\ell}{+}1{-}T} \\
&\le\; \frac{q}{q{-}1} \cdot q^{-r_{\rm h}} \cdot q^{T-r_{\rm v}} \cdot \sum_{\ell=1}^{s} \binom{T}{r_{j_\ell}{+}1{-}T} \qquad (16) \\
&\le\; \frac{q}{q{-}1} \cdot q^{-r_{\rm h}} \cdot q^{-r_{\rm v}} \cdot q^{T} \cdot (2^T{-}1) \;.
\end{aligned}$$

Taking expected values with respect to the probability measure $\mathsf{Prob}\{\, T = t \mid T \le r_{\rm v} \,\}$ yields

$$\begin{aligned}
\sum_{1\le j\le r_{\rm h}\,:\,a_j<r_{\rm v}} \mathsf{B}_T(r_{\rm v}, r_j, q^j) \cdot q^{-j(a_j+1)} \;&\le\; \frac{q}{q{-}1} \cdot q^{-r_{\rm h}} \cdot q^{-r_{\rm v}} \cdot \mathsf{E}_T \left\{ q^T \cdot (2^T{-}1) \;\middle|\; T \le r_{\rm v} \right\} \\
&\le\; \frac{q}{q{-}1} \cdot q^{-r_{\rm h}} \cdot \beta_T(r_{\rm v}) \;\le\; p/2 \;,
\end{aligned}$$

where the last inequality follows from (13). $\square$

Note that for the values of a_j defined in (14) we have

$$r_{\rm v} \;=\; a_0 \;\ge\; a_1 \;\ge\; \ldots \;\ge\; a_{r_{\rm h}-1} \;\ge\; a_{r_{\rm h}} \;=\; 0 \;,$$

and so for $1 \le j \le r_{\mathrm{h}}$, the code $\mathcal{C}_{j-1}$ can be taken as a subcode of $\mathcal{C}_j$.

In order to compute r_{h} from (13) we need to get upper bounds on $\beta_T(r_{\mathrm{v}})$. We obtain such bounds in Appendix B, but we mention here the very simple bound

$$\beta_T(r) \; \le \; 2^r - 1 \; . \tag{17}$$

The condition $r_{\mathrm{h}} \le n_{\mathrm{h}}$ will be satisfied if the acceptable probability of error p is at least $\frac{2q}{q-1}\beta_T(r_{\mathrm{v}})q^{-n_{\mathrm{h}}}$; by (17), this lower bound on p is smaller than $\frac{2q}{q-1}2^{r_{\mathrm{v}}}q^{-n_{\mathrm{h}}}$. Now, if p is smaller than this bound, we will need to take $r_{\mathrm{h}} = n_{\mathrm{h}}$ and increase r_{v} so that $\mathcal{C}_{\mathrm{v}}$ will be able to correct a certain number of errors, in addition to erasures (note that a similar proviso on p is also implied by (3)). This situation, though, will be fairly atypical, and it will probably mean that the initial design parameters n_{h}, n_{v}, or q might need to be re-thought.

Remark. Inequality (16) in the proof of Theorem 1 holds with equality if $j_1 = 1$, which occurs if $r_{\mathrm{h}} \le r_{\mathrm{v}}$. Otherwise, an improvement of the left-hand side of (13) can be obtained by introducing the positive integer variable J, resulting in the following bound on r_{h},

$$\min_{J>0} \max \Big\{ \, (J{-}1) r_{\mathrm{v}} + 1 \; , \; \log_q \Big((q/(q{-}1)) \cdot \beta_T(r_{\mathrm{v}}, J)/(p/2) \Big) \Big\} \; \le \; r_{\mathrm{h}} \; \le \; n_{\mathrm{h}} \; , \tag{18}$$

where

$$\beta_T(r, J) \; = \; q^{-rJ} \cdot \mathsf{E}_T \{ \, q^{J \cdot T} (2^T{-}1) \mid T \le r \, \} \; . \tag{19}$$

By (18), the minimizing J is at most $j_1 = \lceil r_{\mathrm{h}}/r_{\mathrm{v}} \rceil$.

Lemma 2. *Let r_{h} and a_j be defined by (13) and (14). Then,*

$$\sum_{j=0}^{r_{\mathrm{h}}} a_j \; \le \; \lceil r_{\mathrm{h}}/r_{\mathrm{v}} \rceil r_{\mathrm{v}} \; + \; (r_{\mathrm{h}} - 1)\,(\ln \, r_{\mathrm{v}} \; + \; \gamma) \; ,$$

where $\gamma = 0.5772\ldots$ is the Euler constant [1, p. 255].

Proof. Write $j_1 = \lceil r_{\mathrm{h}}/r_{\mathrm{v}} \rceil$. By (14) we have,

$$\begin{aligned}
\sum_{j=0}^{r_{\mathrm{h}}} a_j \;\; &\le \;\; j_1 \cdot r_{\mathrm{v}} \; + \sum_{j=j_1}^{r_{\mathrm{h}}-1} \left(\left\lceil \frac{r_{\mathrm{h}}}{j} \right\rceil - 1 \right) \; \le \; j_1 \cdot r_{\mathrm{v}} \; + \sum_{j=j_1}^{r_{\mathrm{h}}-1} \frac{r_{\mathrm{h}} - 1}{j} \\
&= \;\; j_1 \cdot r_{\mathrm{v}} \; + \; (r_{\mathrm{h}} - 1) \Big(\sum_{j=1}^{r_{\mathrm{h}}-1} \frac{1}{j} \; - \; \sum_{j=1}^{j_1-1} \frac{1}{j} \Big) \\
&\le \;\; j_1 \cdot r_{\mathrm{v}} \; + \; (r_{\mathrm{h}} - 1)\,(\ln \, r_{\mathrm{h}} \; + \; \gamma \; - \; \ln \, j_1) \; ,
\end{aligned}$$

where we have used the inequalities $\ln x \le \sum_{j=1}^{x-1}(1/j) \le \ln x + \gamma$. Hence,

$$\sum_{j=0}^{r_{\mathrm{h}}} a_j \; \le \; j_1 \cdot r_{\mathrm{v}} \; + \; (r_{\mathrm{h}}{-}1) \, \Big(\ln\,(r_{\mathrm{h}}/j_1) + \gamma \Big) \; \le \; \lceil r_{\mathrm{h}}/r_{\mathrm{v}} \rceil r_{\mathrm{v}} \; + \; (r_{\mathrm{h}}{-}1)\,(\ln \, r_{\mathrm{v}} + \gamma) \; ,$$

as claimed. $\square$

We summarize the foregoing discussion by bounding the redundancy of Construction 2 in the following proposition, which follows from (5) and Lemma 2.

Proposition 2. *The redundancy of Construction 2 is bounded from above by*

$$n_{\rm h} r_{\rm v} + (r_{\rm h} - 1)(\ln r_{\rm v} + \gamma + 1) + r_{\rm v} + 1 \,, \tag{20}$$

where $r_{\rm v}$ and $r_{\rm h}$ are set according to (8) and (13).

3.3 Computing the code parameters

The bound (17) on $\beta_T(r)$ allows us to estimate the left-hand side of (13) and set $r_{\rm h}$ to

$$r_{\rm h} \;=\; \left\lceil \frac{r_{\rm v} - \log_2(p/2) - \log_2(q-1)}{\log_2 q} \right\rceil \;+\; 1 \,. \tag{21}$$

In comparison, the value of $r_{\rm h}$ for Construction 0 and Construction 1, as dictated by (3), is given by

$$r_{\rm h} \;=\; \left\lceil \frac{\log_2(\tau(r_{\rm v})) - \log_2(p/2)}{\log_2 q} \right\rceil \,. \tag{22}$$

Hence, when $\tau(r_{\rm v}) = E_T\{\, T \,|\, T \le r_{\rm v} \} \ge 1$, the value of $r_{\rm h}$ in (21) is larger than the one in (22) by an additive term which is at most $\lceil r_{\rm v}/\log_2 q \rceil + 1$. Therefore, applying Construction 0 or Construction 1 with the conservative approach (namely, a coding approach where we insist on keeping the row misdetection probability upper-bounded by $p/2$), the redundancy (20) of Construction 2 can be significantly smaller than the redundancy (4) of Construction 1 (and hence much smaller than the redundancy (1) of Construction 0).

The development leading to (21) and (22) was based, in both cases, on the conservative approach, which assumes very little on the behavior of the actual probability distribution $\mathsf{Prob}\{\, T = t \,\}$. A finer computation and comparison is possible when more detailed knowledge of the distribution is available. An example of such analysis is presented in Appendix B for a Bernoulli distribution. However, the conservative approach is the appropriate choice in the following so-called *cut-off row-error channel*. In this channel, we assume that there is a cut-off error count $r_{\rm c}$ such that $\mathsf{Prob}\{\, T > r_{\rm c} \,\} \le p/2$ and

$$\mathsf{Prob}\{\, T = t \mid T \le r_{\rm c} \,\} \;=\; \begin{cases} 1-\theta_{\rm c} & t = 0 \\ \theta_{\rm c} & t = r_{\rm c} \\ 0 & \text{otherwise} \end{cases} \,.$$

The cut-off row-error channel models (in a rather simplified manner) a case where the array may be susceptible to one long burst event occurring with probability $(1-(p/2))\theta_{\rm c}$, and such an event affects several rows in the array; in our simplified model we assume that the burst affects exactly r_{c} rows, which makes it more amenable to exact analysis.

By (2) and (8) we can take $r_{\mathrm{v}} = r_{\mathrm{c}}$ and have

$$\tau(r_{\mathrm{v}}) \;=\; \mathsf{E}_T\{\, T \mid T \le r_{\mathrm{v}} \,\} \;=\; r_{\mathrm{v}} \cdot \theta_{\mathrm{c}} \; .$$

The computation of $\beta(r_{\mathrm{v}})$ is rather straightforward and we obtain

$$\beta_T(r_{\mathrm{v}}) \;=\; q^{-r_{\mathrm{v}}} \cdot \mathsf{E}_T\{\, q^T (2^T - 1) \mid T \le r_{\mathrm{v}} \,\} \;=\; (2^{r_{\mathrm{v}}} - 1) \cdot \theta_{\mathrm{c}} \; .$$

Hence, by (13) we can choose

$$\begin{aligned} r_{\mathrm{h}} &= \left\lceil \frac{r_{\mathrm{v}} + \log_2 \theta_{\mathrm{c}} - \log_2(p/2) - \log_2(q-1)}{\log_2 q} \right\rceil \;+\; 1 \\ &= \left\lceil \frac{r_{\mathrm{v}} - \log_2 r_{\mathrm{v}} + \log_2(\tau(r_{\mathrm{v}})) - \log_2(p/2) - \log_2(q-1)}{\log_2 q} \right\rceil \;+\; 1 \end{aligned} \tag{23}$$

(note that this value can be smaller than the one in (21)).

On the other hand, it is easy to verify that, for this channel, (22) provides the right choice for r_{h} for Construction 0 and Construction 1. It is also easy to check that the redundancy (20) of Construction 2 can thus be significantly smaller than the redundancy (4) of Construction 1 for this channel. We illustrate this in the next numerical example.

Example. Consider the design of a code with $n_{\mathrm{h}} = 96$, $n_{\mathrm{v}} = 128$ over $F = GF(2^8)$, with a target array error rate of $p = 10^{-17}$. We assume a cut-off row-error model as described above, with $\theta_{\mathrm{c}} = 10^{-3}$ and $r_{\mathrm{c}} = 10$. We set $r_{\mathrm{v}} = r_{\mathrm{c}} = 10$ and by (23) we take $r_{\mathrm{h}} = 8$. Next, we use (14) to obtain $a_0 = 10$, $a_1 = 7$, $a_2 = 3$, $a_3 = 2$, $a_4 = a_5 = a_6 = a_7 = 1$, $a_8 = 0$. The total redundancy is $n_{\mathrm{h}} r_{\mathrm{v}} + \sum_j a_j \;=\; 986$ symbols. In comparison, for Construction 0 and Construction 1, we take by (22) $r_{\mathrm{h}} = 7$, resulting in a redundancy of 1030 symbols for the latter and a redundancy of 1786 symbols for Construction 0. □

In Appendix B, we analyze the Bernoulli row-error channel. In this channel, each row gets affected with probability $\theta = \tau / n_{\mathrm{v}}$, independently of the other rows. It turns out that for typical values of q, n_{v}, and τ we can take

$$r_{\mathrm{h}} \;=\; \left\lceil \frac{r_{\mathrm{v}} + \log_2(r_{\mathrm{v}}+1) - \log_2 \tau}{\log_2 q} \right\rceil \;+\; 1 \; . \tag{24}$$

On the other hand, in Construction 0 and Construction 1 we need to take

$$r_{\mathrm{h}} \;=\; \left\lceil \frac{\frac{3}{2} \log_2(r_{\mathrm{v}}+1) - \log_2 \tau + O(1)}{\log_2 q} \right\rceil \; , \tag{25}$$

and such a value of r_{h} is required also if we do not insist on the conservative approach. Hence, the redundancy of Construction 2 will be typically smaller than that of Construction 1 for the Bernoulli row-error channel, and therefore typically much smaller than that of Construction 0.

Remark. When comparing our construction with Construction 0, we have chosen a fidelity criterion which is the probability p of having a miscorrection in any given $n_{\mathrm{v}} \times n_{\mathrm{h}}$ array. The performance of a coding scheme can be measured also in terms of the effective 'symbol error probability' after decoding, which equals the average fraction of erroneous entries among the decoded entries. For Construction 0, the strategy that we have outlined used the code $\mathcal{C}_{\mathrm{h}}$ for detection only. Therefore, $\mathcal{C}_{\mathrm{h}}$ will never miscorrect, namely, it will never identify an unaffected row as corrupted (on the other hand, it might misdetect corrupted rows). Therefore, the dominant failure event for these codes is one in which $r_{\mathrm{v}}+1$ rows are corrupted, and the condition is detected by $\mathcal{C}_{\mathrm{h}}$, which prevents $\mathcal{C}_{\mathrm{v}}$ from doing any further "damage." Under the constraint (8), the effective symbol error probability after decoding in this case is approximately $(p/2)(r_{\mathrm{v}}+1)/n_{\mathrm{v}}$. In Construction 2, however, the codes $\mathcal{C}_j$ might miscorrect; by the constraint (9), this will happen with probability $p/2$ whenever $T \leq r_{\mathrm{v}}$. Such a miscorrection, in turn, might introduce up to r_{v} false corrupted rows in the decoded array, amounting to an increase of $(p/2)r_{\mathrm{v}}/n_{\mathrm{v}}$ in the effective symbol error probability after decoding. To resolve this, we need to choose a value for p which is one half of the value chosen in Construction 0. Since the dependence of the parameters on p is logarithmic, such a choice of p has a small (if any) effect on the code parameters.

3.4 Summary of Construction 2

To summarize, Construction 2 is obtained as follows:

- Given n_{h}, n_{v}, and p, set the parameter r_{v} to be the smallest positive integer such that (8) holds.
- Set the parameter r_{h} so that (13) holds.
- Set the code $\mathcal{C}_{\mathrm{h}}$ to be an $[n_{\mathrm{h}}, n_{\mathrm{h}}-r_{\mathrm{h}}, r_{\mathrm{h}}+1]$ code over F with an $r_{\mathrm{h}} \times n_{\mathrm{h}}$ parity-check matrix H_{h} which satisfies the MDS supercode property.
- For $0 \leq j \leq r_{\mathrm{h}}$, set $\mathcal{C}_j$ to be an $[n_{\mathrm{v}}, n_{\mathrm{v}}-r_j, r_j+1]$ code over F such that $r_j = r_{\mathrm{v}} + a_j$ and a_j is given by (14). Furthermore, each code $\mathcal{C}_{j-1}$ is a subcode of $\mathcal{C}_j$: the $r_j \times n_{\mathrm{v}}$ parity-check matrix H_j of $\mathcal{C}_j$ consists of the first r_j rows of the $r_{j-1} \times n_{\mathrm{v}}$ parity-check matrix H_{j-1} of $\mathcal{C}_{j-1}$. We let $\mathcal{C}_{\mathrm{v}}$ and H_{v} be $\mathcal{C}_{r_{\mathrm{h}}}$ and $H_{r_{\mathrm{h}}}$, respectively.

Let $\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{r_{\mathrm{h}}-1}$ denote the rows of H_{h}. In Construction 2, the raw data is encoded into an $n_{\mathrm{v}} \times n_{\mathrm{h}}$ array Γ such that the following holds:

- For $0 \le j < r_{\rm h}$,
$$H_j \Gamma \mathbf{h}'_j = \mathbf{0} \ , \tag{26}$$
namely, when the syndrome of each row of Γ is computed with respect to the parity-check matrix $H_{\rm h}$, an $n_{\rm v} \times r_{\rm h}$ array $S = [\, S_0 \ S_1 \ \ldots \ S_{r_{\rm h}-1} \,] = \Gamma H'_{\rm h}$ is obtained in which each column S_j is a codeword of $\mathcal{C}_j$.

- Each column in Γ is a codeword of $\mathcal{C}_{\rm v} = \mathcal{C}_{r_{\rm h}}$, namely
$$H_{\rm v} \Gamma \ = \ 0 \ . \tag{27}$$

We can rewrite (26) and (27) as
$$H_j \Gamma H_{\rm h}^{[j+1]'} = 0 \ , \quad 0 \le j \le r_{\rm h} \ ,$$
where $H_{\rm h}^{[j+1]}$ is the matrix which consists of the first $j{+}1$ rows of $H_{\rm h}$ and $H_{\rm h}^{[r_{\rm h}+1]}$ is defined as the $n_{\rm h} \times n_{\rm h}$ identity matrix.

4 Decoding

The decoding procedure of Construction 2 can be summarized as follows. Let Γ be the transmitted array and let $\tilde{\Gamma}$ be the received array. For each j, $0 \le j \le r_{\rm h}{-}1$, η_j will denote the number of erased locations input to the decoder of $\mathcal{C}_j$ from previous stages.

1. Compute the syndrome array $\tilde{S} = [\, \tilde{S}_0 \ \tilde{S}_1 \ \ldots \ \tilde{S}_{r_{\rm h}-1} \,] = \tilde{\Gamma} H'_{\rm h}$.
2. Set $\eta_0 = 0$. For $j = 0, 1, \ldots, r_{\rm h}{-}1$, do
   a Given the locations of η_j erasures in column $\tilde{S}_j$, apply an error-erasure-correcting decoder for $\mathcal{C}_j$ to locate up to $\lfloor (r_{\rm v}{+}a_j{-}\eta_j)/2 \rfloor$ additional full errors in column $\tilde{S}_j$. For each full error found in $\tilde{S}_j$, mark the corresponding rows in $\tilde{S}$ and $\tilde{\Gamma}$ as erased.
   b Let ϵ_j be the number of full errors found in Step 2a. Set $\eta_{j+1} = \eta_j + \epsilon_j$.
   c If $\eta_{j+1} > r_{\rm v}$, declare the array **undecodable** and **stop**.
3. Apply an erasure-correcting decoder for $\mathcal{C}_{\rm v}$ to recover a total number of up to $r_{\rm v}$ erasures in each column of $\tilde{\Gamma}$.

We point out that the probability of miscorrection will be bounded from above by p also if we limit the number of full errors that we attempt to correct in Step 2a to $\min\{a_j, \lfloor (r_{\rm v}{+}a_j{-}\eta_j)/2 \rfloor\}$.

Steps 2 and 3 can be implemented by choosing the codes $\mathcal{C}_j$ to be RS codes and using any of the known decoding algorithms for these codes, designed to handle both errors and erasures.

The basis of those algorithms is computing an error-locator polynomial $\Lambda(z)$ over the field F (see [2, Ch. 7], [3], [15]) in an iterative manner, such that the roots in F of the computed polynomial $\Lambda(z)$ indicate where the locations of the errors are. If the locations of some errors are initially known (i.e., if some of the errors are actually erasures), then this information can be incorporated into the RS decoding algorithm by a proper initialization of the polynomial $\Lambda(z)$.

In the array decoding procedure outlined above, each stage j, $0 \le j \le r_{\mathrm{h}} - 1$, produces an error locator polynomial Λ_j, which is then fed to the next stage as the initial value of its error locator polynomial Λ_{j+1}. More specifically, we first compute an error-locator polynomial $\Lambda_0(z)$ for the column $\tilde{S}_0$ of $\tilde{S}$. By Lemma 1, the probability of having any corrupted row which is hidden from $\tilde{S}^{[1]} = [\,\tilde{S}_0\,]$ satisfies

$$\mathsf{Prob}\{\, X_1 > 0 \,\} \;\le\; \tau/q$$

(where $\tau = \mathsf{E}_T\{T\}$). Since the redundancy of $\mathcal{C}_0$ is $2r_{\mathrm{v}}$, we will experience a *decoding failure* on $\tilde{S}_0$ only when the number of corrupted rows in $\tilde{\Gamma}$ exceeds r_{v}. Indeed, the constraint (8) guarantees that this will occur only with an acceptably small probability. For subsequent columns of $\tilde{S}$, we compute an error-locator polynomial $\Lambda_j(z)$ that points at the erroneous rows in $\tilde{S}^{[j+1]} = [\,\tilde{S}_0\;\tilde{S}_1\;\ldots\;\tilde{S}_j\,]$. The $\mathcal{C}_j$-decoder will fail on the column $\tilde{S}_j$ only when for some $\ell \le j$, the number of corrupted rows that were hidden from $\tilde{S}^{[\ell]}$ exceeds the correction capability of $\mathcal{C}_\ell$; this occurs when $X_0 + X_\ell > r_\ell$. However, the constraints (8) and (9) (which imply (7)) guarantee that this probability is acceptably small for all j. By Lemma 1, the probability that $\mathcal{C}_j$ will need to correct proper errors (in addition to erasures) satisfies

$$\mathsf{Prob}\{\, X_j > 0 \,\} \;\le\; \tau q^{-j} \;.$$

We can therefore conclude that when $\tau \le r_{\mathrm{v}} \ll n_{\mathrm{v}} \le q$, most of the error-locating effort will typically fall on the $\mathcal{C}_0$-decoder while computing the error-locator polynomial $\Lambda_0(z)$. The role of the rest of the columns of $\tilde{S}$ amounts, in most cases, to *verifying*, with an acceptably small probability of error, that $\Lambda_0(z)$ is the true error-locator polynomial. If the polynomial $\Lambda_0(z)$ turns out to be inconsistent with any of the subsequent columns in $\tilde{S}$, then it will be updated by the decoding algorithm when applied to those columns. At any rate, by well-known properties of linear-recurring sequences [15], it can be shown that the number of such updates is bounded from above by the number of actual corrupted rows, assuming that no failure has occurred in the decoding of any of the columns of $\tilde{S}$. Thus, the total number of operations performed in a typical execution of the array decoding procedure will be significantly smaller than the number of operations in r_{h} independent RS decodings.

5 Encoding

In this section, we outline an encoding procedure for Construction 2. The encoder described here resembles the one in [11], with the following two major differences:

- The new encoder is systematic, namely, the raw data is included, as is, in the encoded array Γ. The encoder in [11], on the other hand, encodes part of the data non-systematically.
- The new encoder is more general in the sense that the codes $\mathcal{C}_j$ have different redundancies.

The raw data is assumed to be entered into Γ column by column, starting at the column $\Gamma_{n_\mathrm{h}-1}$ and ending with Γ_0. We denote the resulting reversed array by $\overleftarrow{\Gamma}$.

We break the encoding procedure into two main steps:

Step A: Encoding raw data into the subarray

$$\Gamma^A \;=\; [\,\Gamma_{r_\mathrm{h}}\;\Gamma_{r_\mathrm{h}+1}\;\ldots\;\Gamma_{n_\mathrm{h}-1}\,]$$

of Γ and computing an $n_\mathrm{v} \times r_\mathrm{v}$ *redundancy array* $V = [\,V_0\;V_1\;\ldots\;V_{r_\mathrm{h}-1}\,]$.

Step B: Encoding the remaining part of the raw data into the subarray

$$\Gamma^B \;=\; [\,\Gamma_0\;\Gamma_1\;\ldots\;\Gamma_{r_\mathrm{h}-1}\,]$$

through the computation of an $n_\mathrm{v} \times r_\mathrm{v}$ syndrome array $S = [\,S_0\;S_1\;\ldots\;S_{r_\mathrm{h}-1}\,]$.

Step B makes use of the redundancy array V that is computed in Step A. The computation of the columns of V can be carried out on-line while reading the data into Γ^B. Therefore, no latency will be caused during encoding. The arrays Γ^A, Γ^B, and V will be generated in reverse form. The reversed arrays will be denoted by $\overleftarrow{\Gamma}^A$, $\overleftarrow{\Gamma}^B$, and $\overleftarrow{V}$.

5.1 Encoding Step A

The computation of the columns of the subarray Γ^A and the redundancy array V is carried out as follows:

Step A1: For $j = n_{\mathrm{h}-1}, n_{\mathrm{h}-2}, \ldots, r_\mathrm{h}$, insert the raw data into the first $n_\mathrm{v}-r_\mathrm{v}$ entries of Γ_j.

Step A2: For $j = n_{\mathrm{h}-1}, n_{\mathrm{h}-2}, \ldots, r_\mathrm{h}$, set the last r_v entries of Γ_j so that Γ_j becomes a codeword of $\mathcal{C}_\mathrm{v} = \mathcal{C}_{r_\mathrm{h}}$.

Steps A1 and A2 are interleaved, and they amount to applying a conventional RS encoder to obtain each column of Γ^A.

Step A3: Set the entries of V so that each row of $[\, V \mid \Gamma^A \,]$ is a codeword of $\mathcal{C}_{\mathrm{h}}$. The computation of V can be done through accumulation of redundancy symbols while reading the data into Γ^A.

Step A2 guarantees that $H_{\mathrm{v}}\Gamma_A = 0$, in accordance with (27). By Step A3 we have

$$[\, V \mid \Gamma^A \,](H_{\mathrm{h}}^*)' = 0 \tag{28}$$

for *any* parity check matrix H_{h}^* of $\mathcal{C}_{\mathrm{h}}$ (in particular, the matrix used here does not have to satisfy the MDS supercode property defined in Section 3.1). Hence, Step A3 can be easily implemented using a *systematic* parity-check matrix of $\mathcal{C}_{\mathrm{h}}$. In this case, the redundancy array V can be computed column by column, while reading the data into Γ^B.

5.2 Encoding Step B

Encoding Step B does depend on the specific choice of the parity-check matrix H_{h} of $\mathcal{C}_{\mathrm{h}}$. In particular, H_{h} will need to satisfy the MDS supercode property, namely, for $1 \leq j \leq r_{\mathrm{h}}$, the matrix $H_{\mathrm{h}}^{[j]}$ is a parity-check matrix of an MDS code.

Let $H_{\mathrm{h}} = [\, h_{k,\ell} \,]_{k=0,\ell=0}^{r_{\mathrm{h}}-1,n_{\mathrm{h}}-1}$ be such a parity-check matrix. Now, for encoding purposes, we usually prefer to have matrices that are systematic (and, indeed, we did choose a systematic matrix in Step A). However, when $r_{\mathrm{h}} > 1$, matrices that satisfy the MDS supercode property must be nonsystematic. Hence, we will require instead the weaker condition $h_{k,\ell} = 0$ for $0 \leq \ell < k < r_{\mathrm{v}}$ and $h_{k,k} = 1$ for $0 \leq k < r_{\mathrm{h}}$. We will refer to such a parity-check matrix as *upper-triangular* (borrowing the term from square matrices). Notice that for each j, the first j rows of such an H_{h} generate an $[n_{\mathrm{h}}, j, n_{\mathrm{h}}{-}j{+}1]$ MDS code [14, Ch. 11]; hence, for any upper-triangular parity-check matrix H_{h} that satisfies the MDS supercode property, we must have $h_{k,\ell} \neq 0$ for $\ell > k$.

The following lemma is an immediate consequence of the fact that every $[n, n{-}r', r'{+}1]$ MDS code has a minimum-weight codeword with zeroes in the first $n{-}r'{-}1$ coordinates.

Lemma 3. *Let $\mathcal{C}$ be an $[n, n{-}r, r{+}1]$ code over $F = GF(q)$ that satisfies the MDS supercode property. Then $\mathcal{C}$ has an $r \times n$ upper-triangular parity-check matrix that satisfies the MDS supercode property.*

We explain next how the columns of Γ^B are encoded. Let H_{h} be an upper-triangular parity-check matrix of $\mathcal{C}_{\mathrm{h}}$ and such that H_{h} satisfies the MDS supercode property. Let $Q = [\, Q_{j,\ell} \,]_{j,\ell=0}^{r_{\mathrm{h}}-1}$ be the inverse of the matrix formed by the first r_{h} columns of H_{h}. Clearly, Q is an $r_{\mathrm{h}} \times r_{\mathrm{h}}$ upper-triangular matrix. Now,

$$\Gamma \;=\; [\, \Gamma^B \mid \Gamma^A \,] \;=\; [\, \Gamma^B{-}V \mid 0 \,] \;+\; [\, V \mid \Gamma^A \,] \,.$$

By (28) we can express the syndrome array $S = S(\Gamma)$ in the following manner:

$$S \;=\; \Gamma H_{\mathrm{h}}' \;=\; [\,\Gamma^B - V \mid 0\,] H_{\mathrm{h}}' \;.$$

Hence,

$$\Gamma^B \;=\; SQ' \;+\; V \;,$$

or

$$\Gamma_j \;=\; S_j \;+\; \sum_{\ell=j+1}^{r_{\mathrm{h}}-1} Q_{j,\ell} S_\ell \;+\; V_j \;, \quad 0 \le j < r_{\mathrm{h}} \;. \tag{29}$$

Letting $(\cdot)_i$ denote the ith component of a vector, we can rewrite (29) in scalar notation as follows:

$$(\Gamma_j)_i \;=\; (S_j)_i \;+\; \sum_{\ell=j+1}^{r_{\mathrm{h}}-1} Q_{j,\ell} (S_\ell)_i \;+\; (V_j)_i \;, \quad 0 \le j < r_{\mathrm{h}} \;, \tag{30}$$

where $0 \le i < n_{\mathrm{v}}$.

Suppose that S_ℓ is known to the encoder for $j < \ell < r_{\mathrm{h}}$. The encoder computes Γ_j and S_j using (30) as follows:

Step B1: Write the raw data into the first $n_{\mathrm{v}} - r_j$ entries in Γ_j.

Step B2: Set the first $n_{\mathrm{v}} - r_j$ entries in S_j so that (30) holds for $0 \le i < n_{\mathrm{v}} - r_j$.

Step B3: Set the last r_j values in S_j so that S_j becomes a codeword of $\mathcal{C}_j$.

Step B4: Set the last r_j values in Γ_j so that (30) holds for $n_{\mathrm{v}} - r_j \le i < n_{\mathrm{v}}$.

Steps B1 through B4 guarantee the following two properties: (a) Γ_j is systematic, namely, its first $n_{\mathrm{v}} - r_j$ entries consist of raw data, and (b) $S_j \in \mathcal{C}_j$.

The encoding procedure is described in Figure 3 in terms of the portions of the array Γ that are computed in each encoding step. An auxiliary $n_{\mathrm{v}} \times r_{\mathrm{h}}$ array is added for the computation of the syndrome array S. The redundancy array V, on the other hand, can be computed in the same area where Γ^B is written. The dotted line separates between the raw data and the redundancy symbols. The encoding steps that are applied in the computation of each particular area of the array are indicated in parentheses.

6 Applying row error-correction (Construction 3)

So far in the constructions, we have used only the detection capability of the code $\mathcal{C}_{\mathrm{h}}$ to mark the corrupted rows. This is a consequence of the fact that we did not assume any model on the number of affected entries in a row, thus assuming in effect the worst-case scenario

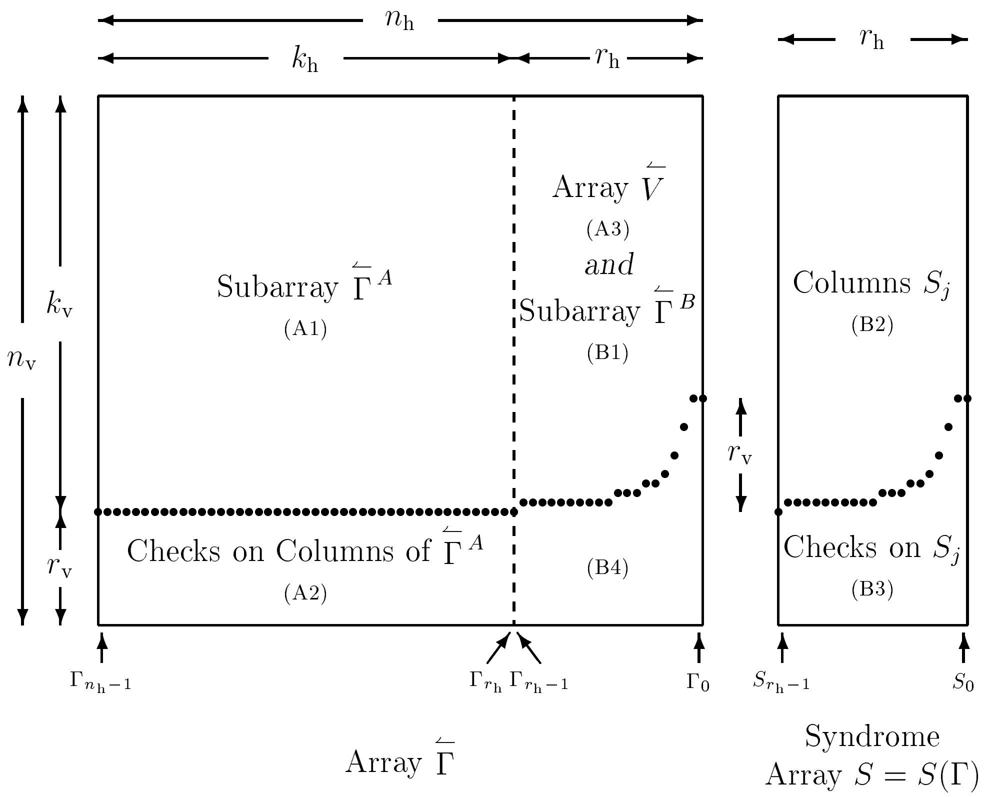

Figure 3: Encoding procedure of Construction 2.

where all the entries in an affected row may get corrupted. Indeed, in such a worst-case event, there is really no use in attempting to correct errors along rows.

In this section, we incorporate partial knowledge on the distribution of the number of affected entries in a row and extend Construction 2 to include some error correction (on top of error detection) on the rows. This approach may be advantageous in cases where there is a significant probability to have only a limited number of affected entries in one row. This is typically the case where the channel inserts both burst and random errors. The resulting extended coding scheme will be referred to as *Construction 3*.

We introduce a design parameter, λ, which marks the number of errors that $\mathcal{C}_h$ will attempt to correct. The ultimate design should optimize over that parameter. The parameter λ will be implicit in all forthcoming notations. The random variable T^- will stand for the number of affected rows each containing no more than λ affected entries. The random variable T^+ will denote the number of rows that contain more than λ affected entries. Clearly, $T = T^- + T^+$.

As before, each column in the array will be a codeword of an $[n_v, n_v - r_v]$ code $\mathcal{C}_v$, except that here we set r_v so that

$$\mathsf{Prob}\{\, T^+ > r_v \,\} \;\le\; p/4 \,. \tag{31}$$

The reasoning here is that the code $\mathcal{C}_{\mathrm{v}}$ will need to correct erasures only in those rows that contain more than λ affected (rather, corrupted) entries. We will introduce another parameter, r'_{v}, which stands for the overall number of affected rows that Construction 3 should be able to handle. The parameter r'_{v} will be determined by the inequality

$$\mathsf{Prob}\{\, T > r'_{\mathrm{v}} \,\} \;\leq\; p/4 \,, \tag{32}$$

which is the analog of (2) or (8).

The code $\mathcal{C}_{\mathrm{h}}$ is chosen to be an $[n_{\mathrm{h}}, n_{\mathrm{h}}-r_{\mathrm{h}}]$ code that satisfies the MDS supercode property, where r_{h} is set so that $\mathcal{C}_{\mathrm{h}}$ can correct any pattern of up to λ full errors or less and detect, with sufficiently high probability, any pattern of more than λ errors. Assuming that the decoder indeed attempts to correct up to λ errors in each row, the probability that a row containing more than λ corrupted entries will be misdetected or miscorrected by $\mathcal{C}_{\mathrm{h}}$ is bounded from above by

$$q^{-r_{\mathrm{h}}} \cdot \sum_{i=0}^{\lambda} \binom{n_{\mathrm{h}}}{i} (q-1)^i \;\leq\; q^{-r_{\mathrm{h}}+2\lambda} \,,$$

where the inequality holds whenever $n_{\mathrm{h}} \leq q$ (we show in Appendix A that this is always the case when the MDS supercode property holds and $r_{\mathrm{h}} > 1$). This bound on the probability takes into account the worst-case scenario where all n_{h} entries in that row may get affected. Given a value of T^+, a decoding failure will occur only if the number of rows that were miscorrected by $\mathcal{C}_{\mathrm{h}}$ exceeds $r_{\mathrm{h}} - T^+$; the probability of this to happen is bounded from above by $\binom{T^+}{r_{\mathrm{v}}+1-T^+} \cdot q^{(-r_{\mathrm{h}}+2\lambda)(r_{\mathrm{v}}+1-T^+)}$. To guarantee the acceptably small probability of decoding failure, we require that r_{h} is chosen so that

$$\mathsf{E}_{T^+} \left\{ \binom{T^+}{r_{\mathrm{v}}+1-T^+} \cdot q^{(-r_{\mathrm{h}}+2\lambda)(r_{\mathrm{v}}+1-T^+)} \;\middle|\; T^+ \leq r_{\mathrm{v}} \right\} \;\leq\; p/4 \,,$$

or, equivalently,

$$\mathsf{B}_{T^+}(r_{\mathrm{v}}, r_{\mathrm{v}}, q^{r_{\mathrm{h}}-2\lambda}) \cdot q^{-r_{\mathrm{h}}+2\lambda} \;\leq\; p/4 \tag{33}$$

(recall the definition in (10)).

The idea behind Construction 3 is that we code a given $n_{\mathrm{v}} \times n_{\mathrm{h}}$ array Γ in a way that makes the respective $n_{\mathrm{v}} \times r_{\mathrm{h}}$ syndrome array $S(\Gamma)$ a mini-array in which we can recover up to r'_{v} affected rows using the decoder of Construction 2 (when designed for $n_{\mathrm{v}} \times r_{\mathrm{h}}$ arrays). Those affected rows are in fact the syndrome vectors of the rows of Γ with respect to the code $\mathcal{C}_{\mathrm{h}}$. Now, the rows of $S(\Gamma)$ are already 'scrambled' versions of the rows of Γ through the use of the code $\mathcal{C}_{\mathrm{h}}$. Therefore, there will be no need to introduce another row-code (i.e., an analog of $\mathcal{C}_{\mathrm{h}}$) for the rows of the mini-array $S(\Gamma)$. We will, however, need to define a parameter r'_{h} and codes $\mathcal{C}'_0, \mathcal{C}'_1, \ldots, \mathcal{C}'_{r'_{\mathrm{h}}-1}, \mathcal{C}'_{r'_{\mathrm{h}}} = \mathcal{C}'_{\mathrm{v}}$ that will be applied to the columns of $S(\Gamma)$ as follows: The codes $\mathcal{C}'_j$, $j = 0, 1, \ldots, r'_{\mathrm{h}}-1$, will be applied to $r'_{\mathrm{h}}-1$ columns of $S(\Gamma)$ (say, the last columns), and $\mathcal{C}'_{\mathrm{v}}$ will be applied to the remaining columns. Note that r'_{h} is not an actual redundancy of any code in this scheme.

Following Theorem 1, we set $r'_{\rm h}$ so that

$$\log_q \left(\frac{q}{q-1} \cdot \frac{\beta_T(r'_{\rm v})}{p/4} \right) \;\le\; r'_{\rm h} \;\le\; r_{\rm h} \;, \tag{34}$$

where $\beta_T(r)$ is as in (12), with $T = T^+ + T^-$. (If $r'_{\rm h} > r_{\rm h}$, then $r_{\rm h}$ should be increased to have $r_{\rm h} = r'_{\rm h}$.) Each code $\mathcal{C}'_j$ is an $[n_{\rm v}, n_{\rm v} - r'_j]$ MDS code where $r'_j = r'_{\rm v} + a'_j$ and a'_j is given by

$$a'_j \;=\; \left\{ \begin{array}{ll} r'_{\rm v} & \text{if } 0 \le j < r'_{\rm h}/r'_{\rm v} \\ \lceil r'_{\rm h}/j \rceil - 1 & \text{if } r'_{\rm h}/r'_{\rm v} \le j \le r'_{\rm h} \end{array} \right. \;, \quad 0 \le j \le r'_{\rm h} \;. \tag{35}$$

The overall redundancy of Construction 3 equals

$$r_{\rm v} n_{\rm h} + (r'_{\rm v} - r_{\rm v}) r_{\rm h} + \sum_{j=0}^{r'_{\rm h}-1} a'_j \;,$$

(compare with (5)), and this redundancy should be minimized over λ.

Given (by (32)) that the number of affected rows is $r'_{\rm v}$ or less, it follows from Theorem 1 that the probability of failing to decode the syndrome array $S(\Gamma)$ is bounded from above by $p/4$. Note that the overall probability of the 'bad events' in (31), (32), and (33), does not exceed $3p/4$.

Figure 4 illustrates the structure of a coded array in which $\mathcal{C}_{\rm h}$ is enhanced to correct λ errors, where the area below the dotted line represents the redundancy symbols. Encoding is carried out in a manner which is similar to the description in Section 5. In fact, the encoding algorithm is *exactly* the same if we define the codes $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{r_{\rm h}}$ as follows:

$$\mathcal{C}_j \;=\; \left\{ \begin{array}{ll} \mathcal{C}'_j & \text{if } 0 \le j < r'_{\rm h} \\ \mathcal{C}'_{\rm v} & \text{if } r'_{\rm h} \le j < r_{\rm h} \\ \mathcal{C}_{\rm v} & \text{if } j = r_{\rm h} \end{array} \right. \;.$$

At the decoding side, we proceed as follows: We use the decoders of $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{r_{\rm h}-1}$ to recover the differential syndrome array ΔS for the received array $\tilde{\Gamma}$. However, unlike the decoding procedure in Section 4, we do need here to recover the full contents of ΔS and not just the locations of the nonzero rows; this is done through the iterative computation of an error-evaluator polynomial $\Omega(z)$ for each column of ΔS, together with the error-locator polynomial $\Lambda(z)$. Once we have the array ΔS, we regard each row in ΔS as a syndrome and apply the decoder of $\mathcal{C}_{\rm h}$ to attempt to correct up to λ errors in the respective row in $\tilde{\Gamma}$. Decoding will succeed if there are at most λ corrupted entries in that row in $\tilde{\Gamma}$. If there are more, then, by (33), the decoder will detect that with sufficiently high probability and mark that row as an erasure. The erasures will then be recovered by $\mathcal{C}_{\rm v}$.

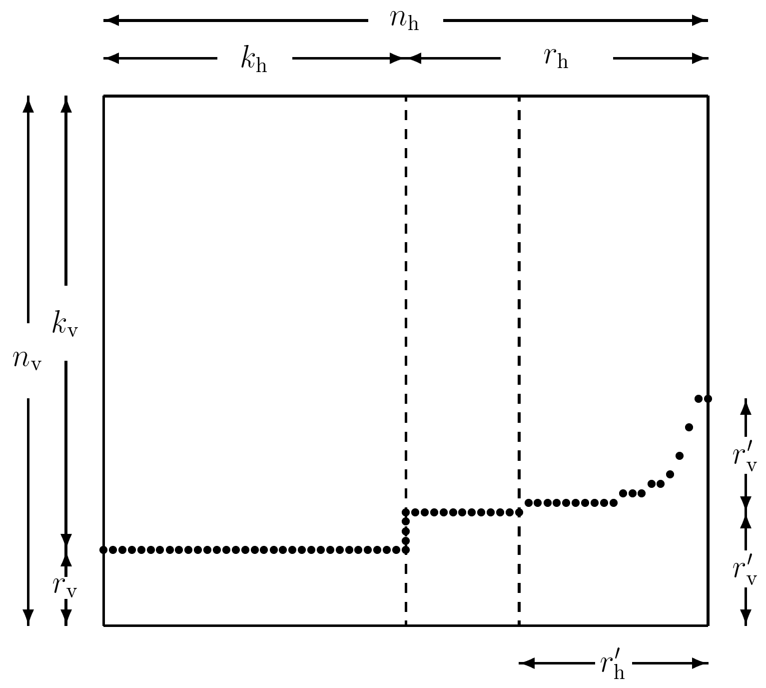

Figure 4: Array structure in Construction 3.

Appendix A

We summarize here several properties of codes that satisfy the MDS supercode property. Recall that a linear $[n, n-r]$ code $\mathcal{C}$ satisfies the MDS supercode property if there exist codes

$$\mathcal{C} = \mathcal{C}_r \subset \mathcal{C}_{r-1} \subset \ldots \subset \mathcal{C}_0 = F^n \tag{36}$$

such that each $\mathcal{C}_j$ is a linear $[n, n-j]$ MDS code.

We first make a connection between such codes and covering radius [14, p. 172]. We denote the covering radius of a code $\mathcal{C}$ by $\rho(\mathcal{C})$.

Lemma 4. (The Supercode Lemma [5, Proposition 1].) *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two distinct codes such that $\mathcal{C}_1 \subset \mathcal{C}_2$ and let d_2 be the minimum Hamming distance of $\mathcal{C}_2$. Then $\rho(\mathcal{C}_1) \geq d_2$.*

Corollary 2. *Let $\mathcal{C}$ be a linear $[n, n-r]$ MDS code. Then $\rho(\mathcal{C}) = r$ if $\mathcal{C}$ is a subcode of some linear $[n, n-r+1]$ MDS code, and $\rho(\mathcal{C}) < r$ otherwise.*

Proof. First, it is well-known [5, Proposition 2] that the covering radius of every linear $[n, n-r]$ code is at most r. This, with Lemma 4, implies the first part of the corollary.

On the other hand, suppose that $\mathcal{C}$ is not contained in any linear $[n, n-r+1, r]$ code. Then, for every vector $\mathbf{e}$ of length n, the coset $\mathcal{C} + \mathbf{e}$ contains a word of Hamming weight less than r. This, in turn, implies that $\rho(\mathcal{C}) < r$. $\square$

It follows from Corollary 2 that a linear $[n, n-r]$ MDS code $\mathcal{C}$ satisfies the MDS supercode property if and only if all codes in the chain (36) except $\mathcal{C}_0$ have covering radii that equal their redundancy; as such, each of these codes has the largest (and therefore the *worst*) covering radius among all codes with the same length and dimension.

A linear $[n, n-r]$ MDS code $\mathcal{C}$ is called *maximum-length* if adding any column to a parity-check of $\mathcal{C}$ results in a parity-check matrix of a linear $[n+1, n+1-r]$ code which is not MDS. Maximum-length MDS codes are extensively studied in projective geometry over finite fields, where they are called *complete arcs* [9],[10].

We next make a connection between the MDS supercode property and maximum-length MDS codes.

Lemma 5. *A linear $[n, n-r]$ MDS code $\mathcal{C}$ is maximum-length if and only if $\rho(\mathcal{C}) < r$.*

Proof. Let $\mathcal{C}$ be an $[n, n-r]$ MDS code over a field F and let H be an $r \times n$ parity-check matrix of $\mathcal{C}$ over F. Clearly, $\rho(\mathcal{C}) < r$ if and only if every (syndrome) vector $\mathbf{h} \in F^r$ can be obtained as a linear combination over F of less than r columns in H. On the other hand, $[\, H \; \mathbf{h} \,]$ is a parity-check of an $[n+1, n+1-r]$ MDS code if and only if $\mathbf{h}$ cannot be obtained as a linear combination of r columns in H. $\square$

Combining Corollary 2 and Lemma 5, we conclude that a linear $[n, n-r]$ MDS code $\mathcal{C}$ satisfies the MDS supercode property if and only there is no code in the chain (36) (except $\mathcal{C}_0$) which is maximum-length.

It is known that every linear $[q+1, q-1]$ MDS code over $GF(q)$ is maximum-length [14, Ch. 11]. Hence, there exist linear $[n, n-r]$ MDS codes over $GF(q)$ that satisfy the MDS supercode property only if $n \le q$ or $r \le 1$. The equality $n = q$ can be attained by extended Reed-Solomon codes.

Appendix B

We analyze here the Bernoulli row-error channel that was considered in Section 3.3. We first improve on the value of r_{h} in (21) by obtaining bounds on $\beta_T(r)$ which are tighter than (17). We do this next using the well-known Chernoff bounding technique.

Let $x \mapsto \mathcal{U}(x)$ be the step function which equals 1 when $x \ge 0$ and equals 0 otherwise. Clearly, $\mathcal{U}(x) \le \delta^{-x}$ for every $0 < \delta \le 1$. Recalling the definition of $\beta_T(r)$ in (12), we have

$$\beta_T(r) \;=\; q^{-r} \cdot \mathsf{E}_T\{\, q^T(2^T-1) \mid T \le r \,\}$$

$$
\begin{aligned}
&= q^{-r} \cdot \tfrac{1}{\mathsf{Prob}\{\,T \le r\,\}} \cdot \mathsf{E}_T\{\, q^T (2^T{-}1) \cdot \mathcal{U}(r-T)\} \\
&\le q^{-r} \cdot \tfrac{1}{\mathsf{Prob}\{\,T \le r\,\}} \cdot \min_{0<\delta\le 1} \mathsf{E}_T\{\, q^T (2^T{-}1) \cdot \delta^{T-r}\} \\
&\le \tfrac{1}{\mathsf{Prob}\{\,T \le r\,\}} \cdot \min_{0<\delta\le 1} \left\{ \, (q\delta)^{-r} \left(\mathsf{E}_T\{\, (2q\delta)^T\} - \mathsf{E}_T\{\, (q\delta)^T\} \right) \right\} \; .
\end{aligned}
$$

Hence, we obtain,

$$
\beta_T(r) \cdot \mathsf{Prob}\{\,T \le r\,\} \;\le\; \min_{0<z\le q} z^{-r} \left(\mathsf{E}_T\{\, (2z)^T\} - \mathsf{E}_T\{\, z^T\} \right) \; . \tag{37}
$$

(The same upper bound holds also for $\beta_T(r, J)$ in (19), except that the range of z becomes $0 \le z \le q^J$.)

We now demonstrate the application of (37) to the Bernoulli row-error channel. Recall that in this channel, each row gets affected with probability $\theta = \tau/n_{\mathrm{v}}$, independently of the other rows. Therefore, T is the sum of $n = n_{\mathrm{v}}$ independent Bernoulli random variables $Y_1, Y_2, \ldots, Y_n$, where $\mathsf{Prob}\{Y_i = 1\} = \theta$ and $\mathsf{Prob}\{Y_i = 0\} = 1{-}\theta$. By (37) we have

$$
\begin{aligned}
\beta_T(r) \cdot \mathsf{Prob}\{\,T \le r\,\} &< \min_{0<z\le q} z^{-r} \mathsf{E}_T\{\, (2z)^T\} \\
&= \min_{0<z\le q} z^{-r} \cdot \mathsf{E}_{Y_1,Y_2,\ldots,Y_n}\{\, (2z)^{Y_1+Y_2+\ldots+Y_n}\} \\
&= \min_{0<z\le q} z^{-r} \cdot \prod_{i=1}^{n} \left(\mathsf{E}_{Y_i}\{(2z)^{Y_i}\} \right) \\
&= \min_{0<z\le q} z^{-r} \cdot (1{-}\theta + 2z\theta)^n \; .
\end{aligned} \tag{38}
$$

The minimum in (38) is attained at $z = \min\{z_{\min}, q\}$, where

$$
z_{\min} \;=\; \frac{r}{n{-}r} \cdot \frac{1{-}\theta}{2\theta} \; .
$$

For the sake of having simpler expressions in the sequel, we will substitute for z the following value

$$
z'_{\min} \;=\; \frac{r{+}1}{n{-}r{-}1} \cdot \frac{1{-}\theta}{2\theta} \; ,
$$

which is slightly larger than $z_{\min}$. We will also assume from now on that $z'_{\min} \le q$, since this is the case for typical values of q, $n = n_{\mathrm{v}}$, $r = r_{\mathrm{v}}$ and $\theta = \tau/n$. We thus obtain the bound

$$
\begin{aligned}
\beta_T(r) \cdot \mathsf{Prob}\{\,T \le r\,\} &< (z'_{\min})^{-r} \cdot (1{-}\theta + 2z'_{\min}\theta)^n \\
&= \frac{n^n}{(r{+}1)^r (n{-}r{-}1)^{n-r}} \cdot \theta^r (1{-}\theta)^{n-r} \cdot 2^r \; .
\end{aligned} \tag{39}
$$

Next we identify portions of the bound (39) with the Chernoff bound on $\mathsf{Prob}\{\,T > r\,\}$. The latter bound is very similar to (38) and is obtained as follows. Recalling the definition of

$x \mapsto \mathcal{U}(x)$ herein,

$$\begin{aligned}
\mathsf{Prob}\{ T > r \} &= \mathsf{E}_T\Big\{ \mathcal{U}(T - (r{+}1)) \Big\} \\
&\le \min_{0<\delta\le 1} \delta^{r+1} \cdot \mathsf{E}_T\{ \delta^{-T} \} \\
&= \min_{0<\delta\le 1} \delta^{r+1} \cdot \prod_{i=1}^{n} \Big(\mathsf{E}_{Y_i}\{\delta^{-Y_i}\} \Big) \\
&= \min_{0<\delta\le 1} \delta^{r+1} \cdot (1{-}\theta + \theta\delta^{-1})^n \,.
\end{aligned}$$

The minimum here is attained at

$$\delta_{\min} = \frac{n{-}r{-}1}{r{+}1} \cdot \frac{\theta}{1{-}\theta}$$

(in our case $r{+}1$ will be at least $\tau = n\theta$ and so $\delta_{\min} \le 1$). Substituting $\delta = \delta_{\min}$ yields

$$\mathsf{Prob}\{ T > r \} \le \frac{n^n}{(r{+}1)^{r+1}(n{-}r{-}1)^{n-r-1}} \cdot \theta^{r+1}(1{-}\theta)^{n-r-1} \,. \tag{40}$$

Note that the bound (40) is rather tight, since, by the Stirling formula we have

$$\begin{aligned}
&\mathsf{Prob}\{ T = r{+}1 \} = \binom{n}{r{+}1} \theta^{r+1}(1{-}\theta)^{n-r-1} \\
&= O(1) \cdot \sqrt{\frac{n}{(r{+}1)(n{-}r{-}1)}} \cdot \frac{n^n}{(r{+}1)^{r+1}(n{-}r{-}1)^{n-r-1}} \cdot \theta^{r+1}(1{-}\theta)^{n-r-1} \,.
\end{aligned}$$

Hence, we set r_{v} to a value r for which the right-hand side of (40) is at most $p/2$. In this case the bound (39) implies

$$\beta_T(r_{\mathrm{v}}) < \frac{p/2}{1{-}(p/2)} \cdot \frac{1{-}\theta}{\theta} \cdot \frac{r_{\mathrm{v}}{+}1}{n_{\mathrm{v}}{-}r_{\mathrm{v}}{-}1} \cdot 2^{r_{\mathrm{v}}} \,, \tag{41}$$

provided that the value of r_{v} that we have chosen is such that $z'_{\min} \le q$, namely,

$$\tau \le r_{\mathrm{v}}{+}1 \le \frac{2q\tau}{1{-}\theta + 2q\theta} \,. \tag{42}$$

Furthermore, since typically $p/2 \le \theta$ and $\frac{1}{n_{\mathrm{v}}} + \frac{1}{q-1} \le \frac{1}{r_{\mathrm{v}}+1}$, then, under those assumptions and (42), it follows from (13) and (41) that we can take r_{h} to be the value in (24) in Section 3.3.

On the other hand, for Construction 0 and Construction 1 we require

$$\mathsf{Prob}\{ T = r_{\mathrm{v}} \} \cdot r_{\mathrm{v}} \cdot q^{-r_{\mathrm{h}}} \le p/2 \,. \tag{43}$$

By the Stirling formula we have

$$\begin{aligned}\mathsf{Prob}\{\,T=r\,\} \;&=\; \binom{n}{r}\theta^r(1{-}\theta)^{n-r} \;=\; \frac{r{+}1}{n{-}r}\cdot\frac{1{-}\theta}{\theta}\cdot\binom{n}{r{+}1}\theta^{r+1}(1{-}\theta)^{n-r-1}\\ &=\; O(1)\cdot\frac{1{-}\theta}{\theta}\cdot\frac{r{+}1}{n{-}r}\cdot\sqrt{\frac{n}{(r{+}1)(n{-}r{-}1)}}\cdot\frac{n^n}{(r{+}1)^{r+1}(n{-}r{-}1)^{n-r-1}}\cdot\theta^{r+1}(1{-}\theta)^{n-r-1}\,.\end{aligned}$$

Hence, if the right-hand side of (40) *equals* $p/2$ for $r = r_{\mathrm{v}}$ then

$$\mathsf{Prob}\{\,T=r_{\mathrm{v}}\,\} \;=\; O(1)\cdot\frac{1{-}\theta}{\theta}\cdot\frac{(n_{\mathrm{v}}(r_{\mathrm{v}}{+}1))^{1/2}}{(n_{\mathrm{v}}{-}r_{\mathrm{v}}{-}1)^{3/2}}\cdot(p/2)\,.$$

This, with (43) implies that we need to set r_{h} in Construction 0 and Construction 1 to the value in (25) in Section 3.3.

7 References

[1] M. Abramowitz, I.A. Stegun, *Handbook of Mathematical Functions,* National Bureau of Standards Applied Mathematics Series 55, 1964.

[2] E.R. Berlekamp, *Algebraic Coding Theory,* Second Edition, Laguna Hills, Englewood Cliffs, California, 1984.

[3] R.E. Blahut, *Theory and Practice of Error Control Codes,* Addison-Wesley, Reading, Massachusetts, 1983.

[4] E.L. Blokh, V.V. Zyablov, *Coding of generalized concatenated codes, Problems of Inform. Trans.,* 10 (1974), 218–222.

[5] G.D. Cohen, M.G. Karpovsky, H.F. Mattson, Jr., J.R. Schatz, *Covering radius — survey and recent results, IEEE Trans. Inform. Theory,* 31 (1985), 328–343.

[6] P.G. Farrell, *A survey of array error control codes, Europ. Trans. Telecommun. Rel. Tech.,* 3 (1992), 441–454.

[7] S. Hirasawa, M. Kasahara, Y. Sugiyama, T. Namekawa, *Certain generalizations of concatenated codes — exponential error bounds and decoding complexity, IEEE Trans. Inform. Theory,* 26 (1980), 527–534.

[8] S. Hirasawa, M. Kasahara, Y. Sugiyama, T. Namekawa, *Modified product codes, IEEE Trans. Inform. Theory,* 30 (1984), 299–306.

[9] J.W.P. Hirschfeld, *Projective Geometries over Finite Fields,* Oxford University Press, Oxford, 1979.

[10] J.W.P. Hirschfeld, J.A. Thas, *General Galois Geometries,* Oxford University Press, Oxford, 1991.

[11] M. Kasahara, S. Hirasawa, Y. Sugiyama, T. Namekawa, *New classes of binary codes constructed on the basis of concatenated codes and product codes, IEEE Trans. Inform. Theory,* 22 (1976), 462–467.

[12] T. Inoue, Y. Sugiyama, K. Ohnishi, T. Kanai, K. Tanaka, *A new class of burst-error-correcting codes and its application to PCM tape recording systems, Proc. IEEE National Communications Conf.*, Part II, Birmingham, Alabama (1978), 20.6/1–5.

[13] S. Lin, D.J. Costello, Jr., *Error Control Coding, Fundamentals and Applications,* Prentice-Hall, Englewoord Cliffs, New Jersey, 1983.

[14] F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes,* North-Holland, Amsterdam, 1977.

[15] J.L. Massey, *Shift-register synthesis and BCH decoding, IEEE Trans. Inform. Theory,* 15 (1969), 122-127.

[16] T. Nishijima, H. Inazumi, S. Hirasawa, *A further improvement of the performance for the original iterated codes, Trans. IEICE,* E-72 (1989), 104–110.

[17] R.M. Roth, *Probabilistic crisscross error correction*, submitted to *IEEE Trans. Inform. Theory.*

[18] V.A. Zinoviev, *Generalized cascade codes, Problems of Inform. Trans.,* 12 (1976), 2–9.

[19] V.A. Zinoviev and V.V. Zyablov, *Decoding of nonlinear generalized cascade codes, Problems of Inform. Trans.,* (1978), 110–114.

[20] V.A. Zinoviev and V.V. Zyablov, *Correcting bursts and independent errors by generalized concatenated codes, Problems of Inform. Trans.,* 15 (1979), 125–134.