Reduced-Redundancy Product Codes for Burst Error Correction*

Ron M. Roth†
Computer Science Department
Technion
Haifa 32000, Israel
e-mail: ronny@cs.technion.ac.il

Gadiel Seroussi
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304, USA.
e-mail: seroussi@hpl.hp.com

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.

---

*Work presented in part at the IEEE International Symposium on Information Theory, Ulm, Germany (June 1997).

†This work was done while R.M. Roth was on sabbatical leave at Hewlett-Packard Laboratories, Palo Alto, California. Part of this work was also done at Hewlett-Packard Laboratories–Israel in Haifa, Israel.

1 Introduction

Product codes [5][10] are a popular choice of error correction mechanism in magnetic recording and other applications 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 $F = GF(q)$ that is encoded by a product code consisting of two linear 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.

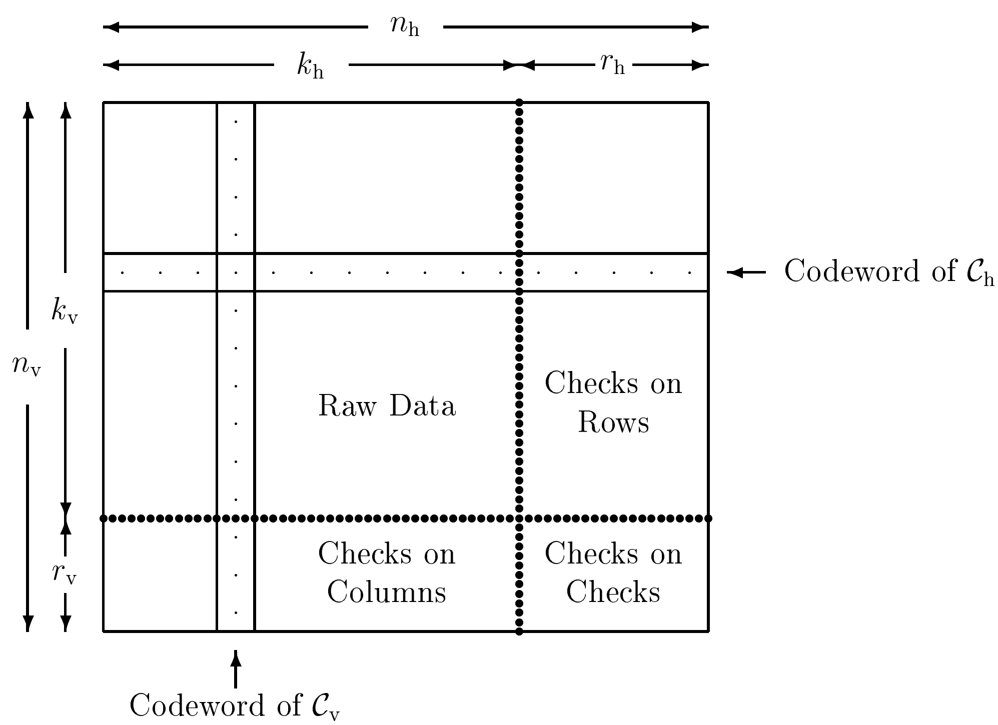

Figure 1: Array Γ of a product code.

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* [10]. 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 $n_{\rm h}$ and $n_{\rm v}$ (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 $\mathcal{C}_{\rm h}$ is first used to *detect* the corrupted rows in a way that we describe shortly. Having found the corrupted rows, the decoder of $\mathcal{C}_{\rm v}$ is applied column by column, now regarding the corrupted entries in each given column as *erasures*. If p is the acceptable probability of array *decoding failure* (i.e. an event in which the original encoded array is *not* properly reconstructed by the decoder), we will allocate half (say) of this probability to the event that the number of errors exceeds the correction capability of $\mathcal{C}_{\rm v}$. In particular, if $\mathcal{C}_{\rm v}$ is an MDS code, then $r_{\rm v}$ can be taken so that

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

This guarantees that the erasure correction capability of $\mathcal{C}_{\rm v}$ is acceptable.

The code $\mathcal{C}_{\rm h}$ detects the corrupted rows by computing, for each row, its syndrome with respect to $\mathcal{C}_{\rm h}$. Let ℓ be the number of affected entries in a given affected row. If $\ell < d_{\rm h} = r_{\rm h}{+}1$, then the computed syndrome for that row must be nonzero in case the row is corrupted. Otherwise, suppose that $\ell > r_{\rm h}$ for a given affected row. Since every $r_{\rm h}$ columns in any $r_{\rm h} \times n_{\rm h}$ parity-check matrix of $\mathcal{C}_{\rm h}$ are linearly independent (by virtue of $\mathcal{C}_{\rm h}$ being MDS), the probability that such an affected row has an all-zero syndrome is $q^{-r_{\rm h}}$ (furthermore, the probability that such a row is corrupted in addition to having an all-zero syndrome is $q^{-r_{\rm h}} - q^{-\ell} < q^{-r_{\rm h}}$). Therefore, regardless of the number of affected entries in a corrupted row, the probability of misdetecting a corrupted row is less than $q^{-r_{\rm h}}$. 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

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

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).[1]

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 $p/2$. 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 large 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

[1]The equal allocation of the probability p between equations (2) and (3) is arbitrary, and other allocations could be considered that might yield slight redundancy improvements for specific parameter values. However, the general results will not be affected by this allocation, as the redundancies are ultimately proportional to $\log p$.

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 5. 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 5 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.* [8], they suggested an improvement on Construction 0 by a technique called *superimposition*. The objective in [8] 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 [15], and Zinoviev and Zyablov [16], [17], where the latter paper also considered minimum-distance decoding of combined random and burst errors. Hirasawa et *al.* [6],[7] presented a similar construction which was shown to increase the code rate while maintaining the decoding failure probability for *random errors*. For related work, see also [9] and [13].

Our main objective in this paper is to increase the code dimension while maintaining the *decoding failure probability for 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.* [8] in the *decoding* mechanism (which we present in Section 3), although the schemes are similar in their *encoding* mechanisms (our encoder is presented in Section 4, where we also discuss in more detail the main differences with [8]). However, the different objective allows us to obtain a more substantial improvement on the code dimension over Construction 0 compared to the construction in [8]. Most aspects of our constructions also differ from those of Blokh and Zyablov [4] and Hirasawa et *al.* [6],[7]. Still, it is worth pointing out a feature which appears both in those constructions and in 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 decoding failure probability.

2 Simple construction with reduced redundancy (Construction 1)

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

Let Γ be an array that consists of $n_{\rm h}$ columns, each being a codeword of $\mathcal{C}_{\rm v}$. 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 $H_{\rm h}$, thus obtaining an $n_{\rm v} \times r_{\rm h}$ *syndrome array* $S = S(\Gamma)$; that is

$$S \;=\; \Gamma H_{\rm h}' \; .$$

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

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 $\tilde{\Gamma} H_{\rm h}'$ corresponding to the received array (see Figure 2). We compare $\tilde{S}$ to the syndrome array $S = S(\Gamma)$ for the transmitted array Γ.[2] More specifically, we consider the *differential syndrome array*

$$\Delta S \;=\; \tilde{S} - S \;=\; E H_{\rm h}' \; .$$

The following two observations can be made:

- If a given row in ΔS is nonzero, 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 $q^{-r_{\rm h}}$.

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

Let $S_0, S_1, \ldots, S_{r_{\rm h}-1}$ denote the columns of S. In order to allow the receiver to locate the nonzero rows in ΔS, the transmitter encodes the raw data in Γ so that each column vector S_j in the resulting syndrome array S is a codeword of an $[n_{\rm v}, n_{\rm v}{-}r_j, r_j{+}1]$ 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.

[2]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.

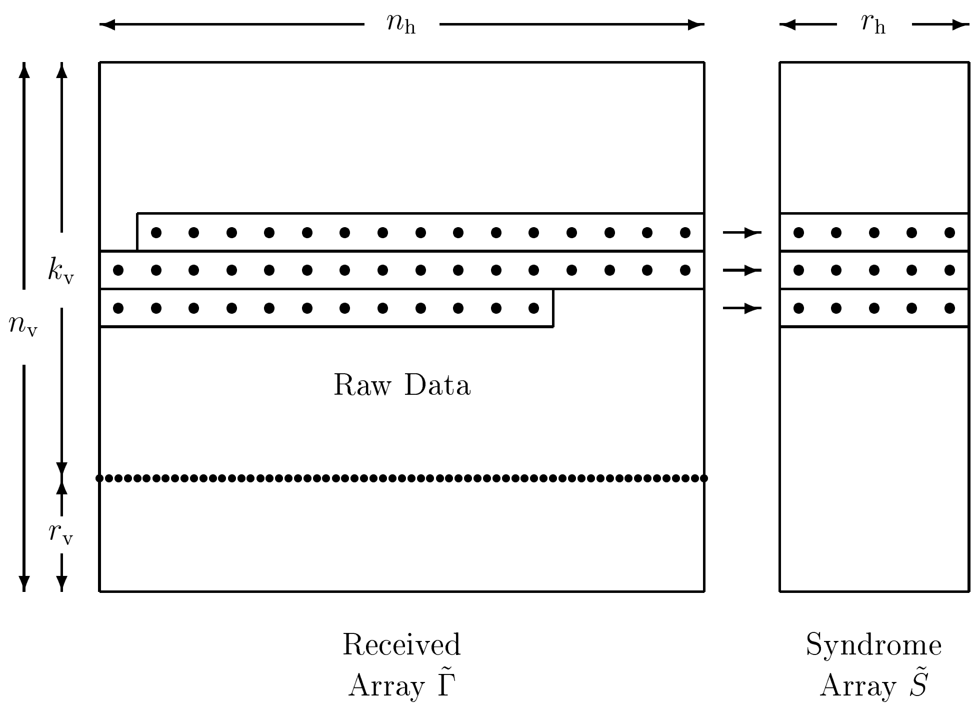

Figure 2: Syndrome array for received array.

A simple choice for $\mathcal{C}_j$, which we assume for the remainder of this section, is to set $r_j = 2r_{\mathrm{v}}$ 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 $r_{\mathrm{h}}/n_{\mathrm{h}} \approx r_{\mathrm{v}}/n_{\mathrm{v}}$ 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.

The following is a summary of Construction 1. Let $\mathcal{C}_{\mathrm{h}}$ and $\mathcal{C}_{\mathrm{v}}$ be as in Construction 0 with parity-check matrices H_{h} and H_{v}, respectively, and denote by $\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{r_{\mathrm{h}}-1}$ the rows of H_{h}. For $0 \le j < r_{\mathrm{h}}$, let $\mathcal{C}_j \subseteq \mathcal{C}_{\mathrm{v}}$ be $[n_{\mathrm{v}}, n_{\mathrm{v}}-r_j, r_j+1]$ MDS codes over $GF(q)$ with $r_j = 2r_{\mathrm{v}}$, and let H_j denote a parity-check matrix of $\mathcal{C}_j$. The code of Construction 1 consists of all $n_{\mathrm{v}} \times n_{\mathrm{h}}$ arrays Γ such that the following holds:

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

- Each column in Γ is a codeword of $\mathcal{C}_{\mathrm{v}} = \mathcal{C}_{r_{\mathrm{h}}}$, namely
$$H_{\mathrm{v}} \Gamma \ = \ 0 \ . \tag{5}$$

3 Decoding

The discussion in Section 2 implies a decoding procedure for Construction 1 which we outline next. Let Γ be the transmitted array and let $\tilde{\Gamma}$ be the received array. For each j, $0 \le j \le r_{\mathrm{h}}-1$, η_j will denote the number of erased locations input to the decoder of $\mathcal{C}_j$ from previous stages. Even though the codes $\mathcal{C}_j$ can in fact be taken as identical in Construction 1, we will maintain the index j; this will emphasize which column of $\tilde{S}$ is being decoded. In addition, this notation will make the decoding algorithm suitable for the generalization presented in Construction 2 of Section 5.

1. Compute the syndrome array $\tilde{S} = [\, \tilde{S}_0 \ \tilde{S}_1 \ \ldots \ \tilde{S}_{r_{\mathrm{h}}-1} \,] = \tilde{\Gamma} H'_{\mathrm{h}}$.

2. Set $\eta_0 = 0$. For $j = 0, 1, \ldots, r_{\mathrm{h}}-1$, 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 $\lfloor (r_j - \eta_j)/2 \rfloor$ 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 $\eta_{j+1} > r_{\mathrm{v}}$, declare the array **undecodable** and **stop**.

3. Apply an erasure-correcting decoder for $\mathcal{C}_{\mathrm{v}}$ to recover a total number of up to r_{v} erasures in each column of $\tilde{\Gamma}$.

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 (see [2, Ch. 7], [3]).

The discussion so far has made no assumptions on the form of the parity-check matrices used for the various codes in Construction 1. We will now impose a constraint on the structure of the parity-check matrix H_{h}, which, while not strictly required for Construction 1, will have some beneficial effects. First, it will guarantee that most of the corrupted rows in $\tilde{\Gamma}$ are detected early in the loop of Step 2 of the decoding procedure. More precisely, the probability that a corrupted row is *not* detected by the j-th iteration of Step 2 will decrease exponentially with j.[3] This feature allows for *adaptive* coding strategies, where the number of stages (and, thus, the computational complexity) of Step 2 can be dynamically adapted to the actual channel noise conditions, or to varying requirements on the parameter p. Second, the exponential decay of the misdetection probability will be a requirement for Construction 2 of Section 5. On the other hand, the additional constraint on H_{h} turns out not to be too restrictive in practice, as it is easily met for Reed-Solomon codes of length $n_{\mathrm{h}} \le q$.

For $1 \le j \le r_{\mathrm{h}}$, let $H_{\mathrm{h}}^{[j]}$ denote the $j \times n_{\mathrm{h}}$ matrix consisting 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 RS codes (possibly shortened or singly-extended).

Notice that when $r_{\mathrm{h}} > 1$, every matrix that satisfies the MDS supercode property must be nonsystematic (i.e., it cannot have the form $[\, I \,|\, A \,]$, where I is an identity matrix of order r_{h}). However, to preserve some of the computational advantages of the systematic form, we will require instead that $H_{\mathrm{h}} = [\, h_{k,\ell} \,]_{k=0,\ell=0}^{r_{\mathrm{h}}-1, n_{\mathrm{h}}-1}$ satisfy the weaker condition $h_{k,\ell} = 0$ for $0 \le \ell < k < r_{\mathrm{v}}$ and $h_{k,k} = 1$ for $0 \le k < r_{\mathrm{h}}$. 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

[3] This property is not valid for all parity-check matrices, since we require it to hold for *all* possible error location configurations, with the probability space being defined just by the error values.

code [11, 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 1. *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 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. A more detailed discussion and characterizations of the MDS supercode property can be found in [14].

For a syndrome array $S = \Gamma H_{\mathrm{h}}'$ and $1 \leq j \leq r_{\mathrm{h}}$ we denote by $S^{[j]}$ the matrix formed by the first j columns of S, i.e.,

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

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 $1 \leq j \leq r_{\mathrm{h}}$, denote by X_j the random variable that 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 $\tilde{\Gamma}$.

The following lemma establishes the desired exponential decay, with j, of the row misdetection probability. The proof is deferred to the Appendix.

Lemma 2. *Suppose that H_{h} satisfies the MDS supercode property. Then, 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 $b \geq t$.

Assume that H_{h} satisfies the MDS supercode property. By Lemma 2 we have

$$\mathsf{Prob}\{\, X_j > 0 \,\} \leq \tau q^{-j} ,$$

which means that the probability of misdetecting the corrupted rows decreases exponentially with j. When $\tau \le r_{\rm v} \ll n_{\rm v} \le q$, most of the error-locating effort will typically fall on the $\mathcal{C}_0$-decoder. The role of the rest of the columns of $\tilde{S}$ amounts, in most cases, to *verifying*, with an acceptably small probability p of error, that we have located all the corrupted rows.

4 Encoding

In this section, we outline an encoding procedure for Construction 1. The encoder described here resembles the one in [8], 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 [8], on the other hand, encodes part of the data nonsystematically.
- The new encoder is more general in the sense that it allows the codes $\mathcal{C}_j$ to have different redundancies. This will turn out to be useful for Construction 2 in Section 5.

The raw data is assumed to be entered into Γ column by column, starting at the column $\Gamma_{n_{\rm h}-1}$ and ending with Γ_0. We denote the resulting reversed array $[\,\Gamma_{n_{\rm h}-1}\ \ldots\ \Gamma_1\ \Gamma_0\,]$ by $\overleftarrow{\Gamma}$.

The encoding procedure is divided into two main steps:

Step A: Encoding raw data into the subarray

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

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

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

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

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

Step B makes use of the redundancy array V 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 is incurred during encoding. The arrays Γ^A, Γ^B, and V are generated in reverse form, with the reversed arrays denoted, respectively, by $\overleftarrow{\Gamma}^A$, $\overleftarrow{\Gamma}^B$, and $\overleftarrow{V}$.

4.1 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 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 $\mathcal{C}_{\mathrm{v}} = \mathcal{C}_{r_{\mathrm{h}}}$.

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 (5). By Step A3 we have

$$[\, V \mid \Gamma^A \,](H_{\mathrm{h}}^*)' = 0 \tag{6}$$

for any parity-check matrix H_{h}^* of $\mathcal{C}_{\mathrm{h}}$. 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.

4.2 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, as discussed in Section 3, we will assume for the remaining of this section that $H_{\mathrm{h}} = [\, h_{k,\ell} \,]_{k=0,\ell=0}^{r_{\mathrm{h}}-1, n_{\mathrm{h}}-1}$ is an upper-triangular parity-check matrix satisfying the MDS supercode property.

We explain next how the columns of Γ^B are encoded. 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 (6) 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,

$$\Gamma^B \;=\; S Q' \;+\; V \,,$$

or

$$\Gamma_j \;=\; S_j \;+\; \sum_{\ell=j+1}^{r_{\mathrm{h}}-1} Q_{j,\ell} S_\ell \;+\; V_j \;, \quad 0 \le j < r_{\mathrm{h}} \;. \tag{7}$$

Letting $(\cdot)_i$ denote the ith component of a vector, we can rewrite (7) in scalar notation as follows:

$$(\Gamma_j)_i \;=\; (S_j)_i \;+\; \sum_{\ell=j+1}^{r_{\mathrm{h}}-1} Q_{j,\ell} (S_\ell)_i \;+\; (V_j)_i \;, \quad 0 \le j < r_{\mathrm{h}} \;, \tag{8}$$

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 (8) 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 (8) 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 (8) 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 illustrated 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 included 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.

5 Further redundancy reduction (Construction 2)

Referring to the decoding procedure of Section 3, we recall that for each $1 \le j < r_{\mathrm{h}}$, the $\mathcal{C}_{j-1}$-decoder passes to the $\mathcal{C}_j$-decoder *erasure* information obtained from columns of $\tilde{S}$ that have already been processed. Therefore, the error-correction "load" on $\mathcal{C}_j$ is lighter than that of its predecessors, and it might be possible to design the sequence of codes $\mathcal{C}_j$, $0 \le j \le r_{\mathrm{h}}-1$, so that the redundancy of the codes generally decreases as j

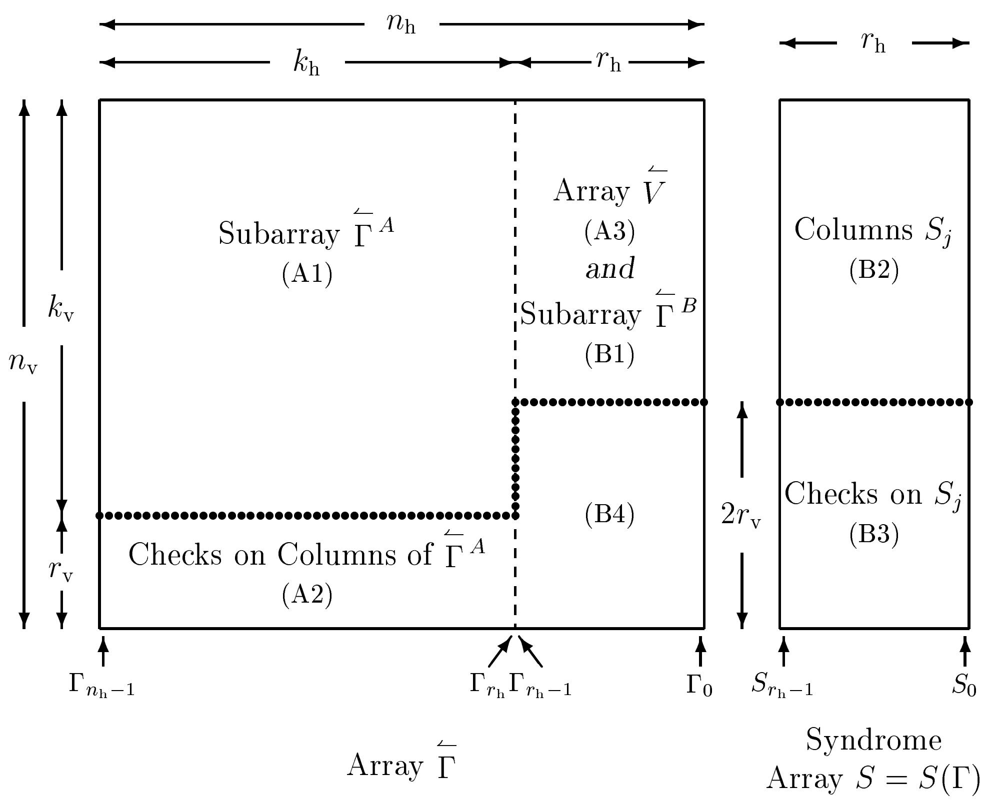

Figure 3: Encoding procedure of Construction 1.

increases. 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 resulting coding scheme will be referred to as *Construction 2*. As we show next, the code parameters can be designed so that 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. Recall that there may be as many as $n_{\mathrm{h}} r_{\mathrm{v}}$ affected entries in r_{v} affected rows, in which case we will require at least $n_{\mathrm{h}} r_{\mathrm{v}}$ redundant symbols to decode those entries. Hence, the improvement over Construction 1 is significant in the following sense: Construction 1 has a redundancy "overhead" of $r_{\mathrm{h}} r_{\mathrm{v}}$ symbols above $n_{\mathrm{h}} r_{\mathrm{v}}$, whereas the overhead in Construction 2 is reduced to $r_{\mathrm{h}}(\ln r_{\mathrm{v}} + O(1)) + r_{\mathrm{v}}$.

Codes with varying redundancies have also been used in [4],[6], and [7] for random errors. In those constructions, the conventional product code is modified so that the (explicit) codes $\mathcal{C}_{\rm h}$ and $\mathcal{C}_{\rm v}$ may have different redundancies at different rows or columns (see for example Figure 1 in [7]). Our varying-redundancy codes, $\mathcal{C}_j$, manifest themselves indirectly in the code array through constraints on an implicit syndrome array.

5.1 Computing the redundancies of the codes $\mathcal{C}_j$

We assume that the parity-check matrix $H_{\rm h}$ of $\mathcal{C}_{\rm h}$ used to compute S satisfies the MDS supercode property. We also assume that for $1 \le j < r_{\rm h}$, each code $\mathcal{C}_j$ is a subcode of $\mathcal{C}_{\rm v}$ and so we can write $r_j = r_{\rm v} + a_j$ where $a_j \ge 0$. The overall redundancy of Construction 2 thus equals

$$r_{\rm v} n_{\rm h} + \sum_{j=0}^{r_{\rm h}-1} a_j \ . \qquad (9)$$

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

Recalling the notations X_j from Section 3, we can now formulate our problem as follows: Given an acceptable probability p of decoding failure, find nonnegative integers $r_{\rm v}(\le n_{\rm v})$, $r_{\rm h}(\le n_{\rm h})$ and $a_0, a_1, \ldots, a_{r_{\rm h}-1}$ (and $a_{r_{\rm h}} = 0$) that minimize (9) subject to the constraint

$$\mathsf{Prob}\Big\{ \bigcup_{j=0}^{r_{\rm h}} \{ X_0 + X_j > r_{\rm v} + a_j \} \Big\} \ \le \ p \ . \qquad (10)$$

The constraint (10) 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 $(X_0 - X_j) + 2X_j$).

While the precise minimization of (9) subject to (10) appears quite intractable, we can construct a feasible solution that satisfies (10) and achieves the claimed savings in redundancy. We derive such a solution in the following discussion.

We choose $a_0 = r_{\rm v}$ (i.e., $r_0 = 2r_{\rm v}$), which is consistent with our strategy that the $\mathcal{C}_0$-decoder handles as many as $r_{\rm v}$ errors without having any a priori information on the locations of those errors.

Now,

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

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

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

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 (10) is implied by

$$\mathsf{Prob}\{ T > r_\mathrm{v} \} \ \leq \ p/2 \tag{11}$$

and

$$\sum_{j=1}^{r_\mathrm{h}} \mathsf{Prob}\Big\{ T + X_j > r_\mathrm{v} + a_j \ \Big| \ T \leq r_\mathrm{v} \Big\} \ \leq \ p/2 \ . \tag{12}$$

Satisfying the constraint (12) 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.

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

$$\mathsf{B}_T(r, s, \omega) \ = \ \mathsf{E}_T \left\{ \binom{T}{s+1-T} \cdot \omega^{T-r} \ \Big| \ T \leq r \right\} \ . \tag{13}$$

Clearly, $\mathsf{B}_T(r, s, \omega) = 0$ when $s \geq 2r$. The following corollary follows readily from an application of Lemma 2 to bound $\mathsf{Prob}\{ T + X_j > r_\mathrm{v} + a_j \mid T \}$, combined with the definition of B_T.

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

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

By Corollary 1, we can replace the constraint (12) by the following stronger condition,

$$\sum_{1 \leq j \leq r_\mathrm{h} \, : \, a_j < r_\mathrm{v}} \mathsf{B}_T(r_\mathrm{v}, r_j, q^j) \cdot q^{-j(a_j+1)} \ \leq \ p/2 \ , \tag{14}$$

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 (14) 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 (11)), then we can also state conversely that $\mathsf{B}_T(r_\mathrm{v}, r_j, q^j) > 0$ whenever $a_j < r_\mathrm{v}$.

A feasible solution for the a_j, satisfying the constraint (14), is given in Theorem 1 below. Combined with (11), this solution will also satisfy constraint (10).

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

$$\beta_T(r) \;=\; q^{-r} \cdot \mathsf{E}_T\{\, q^T (2^T - 1) \mid T \le r \,\} \; . \tag{15}$$

Theorem 1. *Given an acceptable probability p of miscorrection, let $r_{\rm v}$ be a positive integer (such as an integer that satisfies (11)) and let $r_{\rm h}$ be an integer such that*

$$\log_q \left(\frac{q}{q-1} \cdot \frac{\beta_T(r_{\rm v})}{p/2} \right) \;\le\; r_{\rm h} \;\le\; n_{\rm h} \tag{16}$$

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

$$a_j \;=\; \begin{cases} r_{\rm v} & \text{if } 0 \le j < r_{\rm h}/r_{\rm v} \\ \lceil r_{\rm h}/j \rceil - 1 & \text{if } r_{\rm h}/r_{\rm v} \le j \le r_{\rm h} \end{cases} \; , \quad 0 \le j \le r_{\rm h} \; , \tag{17}$$

satisfy the constraint (14).

The proof of Theorem 1 is given in the Appendix. For the values of a_j defined in (17) 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 $1 \le j \le r_{\rm h}$, the code $\mathcal{C}_{j-1}$ can be taken as a subcode of $\mathcal{C}_j$.

In order to compute $r_{\rm h}$ from (16) we need an upper bound on $\beta_T(r_{\rm v})$. From (15) we get the very simple bound

$$\beta_T(r) \;\le\; 2^r - 1 \; . \tag{18}$$

The condition $r_{\rm h} \le n_{\rm h}$ will be satisfied if the acceptable probability of error p is at least $\frac{2q}{q-1}\beta_T(r_{\rm v})q^{-n_{\rm h}}$; by (18), this lower bound on p is smaller than $\frac{2q}{q-1}2^{r_{\rm v}}q^{-n_{\rm h}}$. Now, if p is smaller than this bound, we will need to take $r_{\rm h} = n_{\rm h}$ and increase $r_{\rm v}$ so that $\mathcal{C}_{\rm v}$ 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 $n_{\rm h}$, $n_{\rm v}$, or q might need to be re-thought.

Lemma 3. *Let $r_{\rm h}$ and a_j be defined by (16) and (17). Then,*

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

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 (17) we have,

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

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

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

as claimed. □

We summarize the foregoing discussion by bounding the redundancy of Construction 2 in the following proposition, which follows from (9) and Lemma 3.

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

$$n_\mathrm{h} r_\mathrm{v} + (r_\mathrm{h}-1)(\ln r_\mathrm{v} + \gamma + 1) + r_\mathrm{v} + 1\;, \tag{19}$$

where r_v and r_h are set according to (11) and (16).

5.2 Computing the redundancy of $\mathcal{C}_\mathrm{h}$

The bound (18) on $\beta_T(r)$ allows us to estimate the left-hand side of (16) and set r_h to

$$r_\mathrm{h} \;=\; \left\lceil \frac{r_\mathrm{v} - \log_2(p/2) - \log_2(q-1)}{\log_2 q} \right\rceil \;+\; 1\;. \tag{20}$$

In comparison, the value of r_h for Construction 0 and Construction 1, as dictated by (3), is given by

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

Hence, when $\tau(r_\mathrm{v}) = E_T\{\,T \,|\, T \le r_\mathrm{v}\} \ge 1$, the value of r_h in (20) is larger than the one in (21) by an additive term which is at most $\lceil r_\mathrm{v}/\log_2 q \rceil + 1$. 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 (19) 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 (20) and (21) 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. 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_{c} such that $\mathsf{Prob}\{\, T > r_{\mathrm{c}} \,\} \le p/2$ and

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

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 $(1-(p/2))\theta_{\mathrm{c}}$, 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 (11) 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 (16) we can choose

$$\begin{aligned} r_{\mathrm{h}} \;&=\; \left\lceil \frac{r_{\mathrm{v}} + \log_2 \theta_{\mathrm{c}} - \log_2(p/2) - \log_2(q-1)}{\log_2 q} \right\rceil \;+\; 1 \\ &=\; \left\lceil \frac{r_{\mathrm{v}} - \log_2 r_{\mathrm{v}} + \log_2(\tau(r_{\mathrm{v}})) - \log_2(p/2) - \log_2(q-1)}{\log_2 q} \right\rceil \;+\; 1 \end{aligned} \tag{22}$$

(note that this value can be smaller than the one in (20)).

On the other hand, it is easy to verify that, for this channel, (21) provides the right choice for r_{h} for Construction 0 and Construction 1. It is also easy to check that the redundancy (19) 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 $n_h = 96$, $n_v = 128$ 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 $\theta_c = 10^{-3}$ and $r_c = 10$. We set $r_v = r_c = 10$ and by (22) we take $r_h = 8$. Next, we use (17) 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 $n_h r_v + \sum_j a_j = 986$ symbols. Note that any coding scheme that is designed to correct 10 corrupted rows of length 96 must have at least 960 redundant symbols. So, the redundancy overhead of Construction 2 above that number is only $\sum_j a_j = 986 - 960 = 26$ symbols. In comparison, for Construction 0 and Construction 1, we take by (21) $r_h = 7$, resulting in a redundancy of 1030 symbols for the latter (i.e., overhead of 70 symbols above 960) and a redundancy of 1786 symbols (overhead of 826 symbols) for the former. □

An analysis of the Bernoulli row-error channel can be found in [14]. In this channel, each row gets affected with probability $\theta = \tau / n_v$, independently of the other rows. As shown in [14], for typical values of q, n_v, and τ we can take

$$r_h = \left\lceil \frac{r_v + \log_2(r_v+1) - \log_2 \tau}{\log_2 q} \right\rceil + 1 . \tag{23}$$

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

$$r_h = \left\lceil \frac{\frac{3}{2}\log_2(r_v+1) - \log_2 \tau + O(1)}{\log_2 q} \right\rceil , \tag{24}$$

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.

5.3 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 (11) holds.
- Set the parameter r_h so that (16) holds.
- Set the code $\mathcal{C}_h$ to be an $[n_h, n_h - r_h, r_h + 1]$ code over F with an $r_h \times n_h$ parity-check matrix H_h that 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 (17). 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_h}, respectively.

The decoding algorithm of Section 3 and the encoding algorithm of Section 4 are suitable also for Construction 2. Figure 4 shows the various portions of the array Γ that are computed in each encoding step.

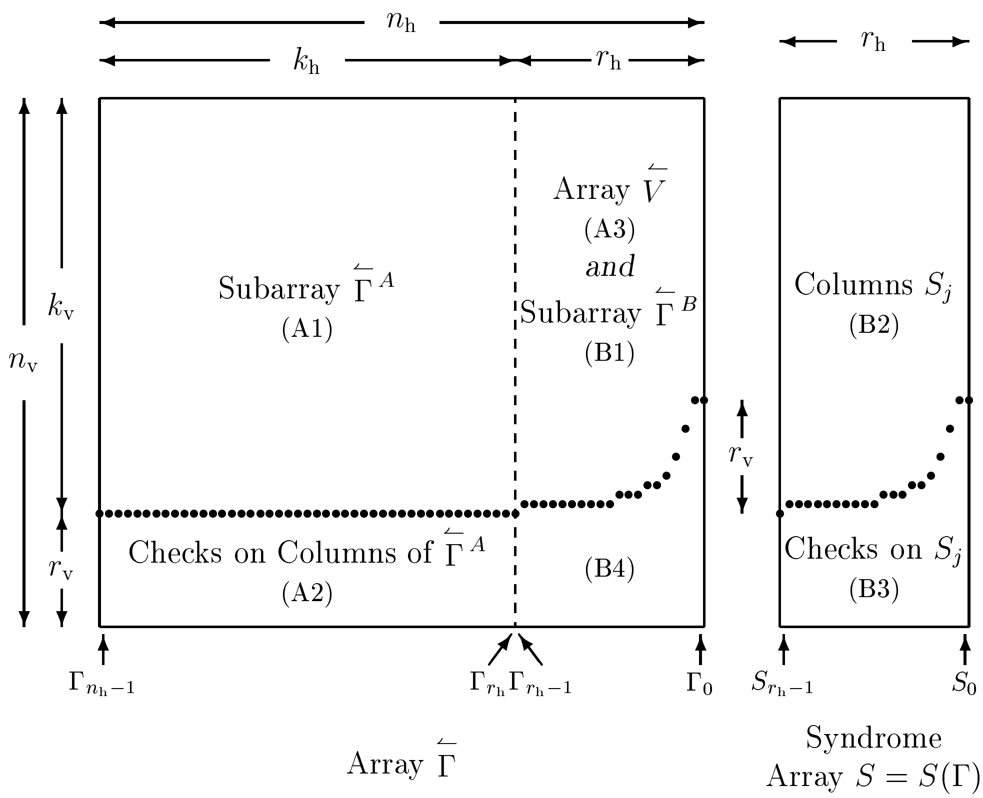

Figure 4: Encoding procedure of Construction 2.

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 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 $\mathcal{C}_{\rm h}$ 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 $[n_{\rm v}, n_{\rm v} - r_{\rm v}]$ code $\mathcal{C}_{\rm v}$, except that here we set $r_{\rm v}$ so that

$$\mathsf{Prob}\{\, T^+ > r_{\rm v} \,\} \;\le\; p/4 \,. \tag{25}$$

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

$$\mathsf{Prob}\{\, T > r'_{\rm v} \,\} \;\le\; p/4 \,, \tag{26}$$

which is the analog of (2) or (11).

The code $\mathcal{C}_{\rm h}$ is chosen to be an $[n_{\rm h}, n_{\rm h} - r_{\rm h}]$ code that satisfies the MDS supercode property, where $r_{\rm h}$ is set so that $\mathcal{C}_{\rm h}$ 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 for $\mathcal{C}_{\rm h}$ indeed attempts to correct up to λ errors in each row, the following argument provides an upper bound on the probability that a row containing more than λ corrupted entries be miscorrected (see also [12]). Let Y denote the set of indices of affected locations in a given row, for a given error event with $|Y| = \ell$. The row can be miscorrected only if there exists a set of locations W, with $|W| \le \lambda$, such that $\mathcal{C}_{\rm h}$ has a nonzero codeword with support included in $Y \cup W$. For a given set W, the number of such codewords is less than $q^{\ell+\lambda-r_{\rm h}}$. Taking a union over all possible sets W, it follows that the total number of "bad" codewords for a given Y is less than $\binom{n_{\rm h}}{\lambda} q^{\ell+\lambda-r_{\rm h}}$. Given the channel under consideration, the probability of the entries in Y matching a given "bad" codeword is $q^{-\ell}$. Thus, the probability of miscorrection for a given set Y of affected entries is bounded from above by $\binom{n_{\rm h}}{\lambda} q^{\lambda - r_{\rm h}}$. Since this bound holds for all Y, it is an upper bound on the miscorrection probability of a row, independent of Y. Writing $\binom{n_{\rm h}}{\lambda} < n_{\rm h}^{\lambda}$, it follows that whenever $n_{\rm h} \le q$ (as is the case if RS codes, possibly singly-extended, are used), the probability of row miscorrection is bounded from above by

$$q^{-r_{\rm h}+2\lambda} \,. \tag{27}$$

Given a value of T^+, a decoding failure will occur only if the number of rows that were miscorrected by $\mathcal{C}_{\rm h}$ exceeds $r_{\rm v} - T^+$; by the discussion leading to (27), the probability of this event is bounded from above by $\binom{T^+}{r_{\rm v}+1-T^+} \cdot q^{(-r_{\rm h}+2\lambda)(r_{\rm v}+1-T^+)}$. To guarantee the acceptably small probability of decoding failure, we require that $r_{\rm h}$ is chosen so that

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

or, equivalently,

$$\mathsf{B}_{T^+}(r_{\rm v}, r_{\rm v}, q^{r_{\rm h}-2\lambda}) \cdot q^{-r_{\rm h}+2\lambda} \;\le\; p/4 \tag{28}$$

(recall the definition in (13)).

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

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

$$\log_q \left(\frac{q}{q-1} \cdot \frac{\beta_T(r'_{\rm v})}{p/4} \right) \;\le\; r'_{\rm h} \;\le\; r_{\rm h} \,, \tag{29}$$

where $\beta_T(r)$ is as in (15), 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

$$a'_j \;=\; \begin{cases} r'_{\rm v} & \text{if } 0 \le j < r'_{\rm h}/r'_{\rm v} \\ \lceil r'_{\rm h}/j \rceil - 1 & \text{if } r'_{\rm h}/r'_{\rm v} \le j \le r'_{\rm h} \end{cases} , \quad 0 \le j \le r'_{\rm h} \,. \tag{30}$$

The overall redundancy of Construction 3 equals

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

(compare with (9)), and this redundancy should be minimized over λ.

Given, by (26), that the number of affected rows is r'_{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 (25), (26), and (28), does not exceed $3p/4$.

Figure 5 illustrates the structure of a coded array in which $\mathcal{C}_{\mathrm{h}}$ 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 4. In fact, the encoding algorithm is *exactly* the same if we define the codes $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{r_{\mathrm{h}}}$ as follows:

$$\mathcal{C}_j = \begin{cases} \mathcal{C}'_j & \text{if } 0 \le j < r'_{\mathrm{h}} \\ \mathcal{C}'_{\mathrm{v}} & \text{if } r'_{\mathrm{h}} \le j < r_{\mathrm{h}} \\ \mathcal{C}_{\mathrm{v}} & \text{if } j = r_{\mathrm{h}} \end{cases} .$$

At the decoding side, we proceed as follows: We use the decoders of $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{r_{\mathrm{h}}-1}$ to recover the differential syndrome array ΔS for the received array $\tilde{\Gamma}$. However, unlike the decoding procedure in Section 3, we do need here to recover the full contents of ΔS and not just the locations of the nonzero rows. Once we have the array ΔS, we regard each row in ΔS as a syndrome and apply the decoder of $\mathcal{C}_{\mathrm{h}}$ 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 (28), the decoder will detect that with sufficiently high probability and mark that row as an erasure. The erasures will then be recovered by $\mathcal{C}_{\mathrm{v}}$.

Appendix

Proof of Lemma 2. 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 that equals the number of those rows. For each such row, the respective

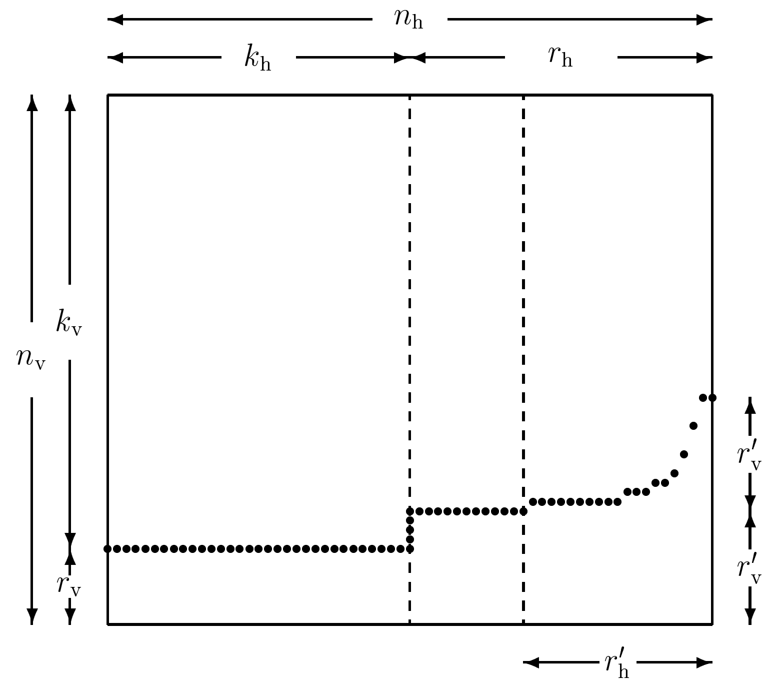

Figure 5: Array structure in Construction 3.

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 $\sigma \le T$, we have,

$$\begin{aligned} \mathsf{Prob}\{\, X_j > b \mid T, \sigma \,\} &\le \sum_{z=b+1}^{\sigma} \binom{\sigma}{z} q^{-jz}(1-q^{-j})^{\sigma-z} \\ &\le \binom{\sigma}{b+1} q^{-j(b+1)} \le \binom{T}{b+1} q^{-j(b+1)} . \end{aligned} \tag{31}$$

The second inequality in (31) follows by observing that the left-hand side of the inequality computes the exact probability of at least $b+1$ successes in σ independent Bernoulli trials with parameter q^{-j}, while the right-hand side bounds the same probability using a (possibly redundant) union of events containing $b+1$ successes, where each such event is counted at least once. It follows from (31) that

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

as claimed. □

Proof of Theorem 1. Let $j_1 < j_2 < \ldots < j_s$ be a sequence consisting 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,

$$\begin{aligned}
\sum_{j=j_\ell}^{j_{\ell+1}-1} \binom{T}{r_j+1-T} \cdot q^{-j(r_j+1-T)} &= \binom{T}{r_{j_\ell}+1-T} \cdot \sum_{j=j_\ell}^{j_{\ell+1}-1} q^{-j(r_{j_\ell}+1-T)} \\
&\le \frac{q}{q-1} \cdot q^{-j_\ell(r_{j_\ell}+1-T)} \cdot \binom{T}{r_{j_\ell}+1-T} \\
&\le \frac{q}{q-1} \cdot q^{-r_{\rm h}} \cdot q^{j_\ell(T-r_{\rm v})} \cdot \binom{T}{r_{j_\ell}+1-T} , \qquad (32)
\end{aligned}$$

where (32) follows from (17), 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,

$$\begin{aligned}
\sum_{1 \le j \le r_{\rm h}\,:\,a_j < r_{\rm v}} \binom{T}{r_j+1-T} \cdot q^{-j(r_j+1-T)} &= \sum_{\ell=1}^{s} \sum_{j=j_\ell}^{j_{\ell+1}-1} \binom{T}{r_j+1-T} \cdot q^{-j(r_j+1-T)} \\
&\overset{(32)}{\le} \frac{q}{q-1} \cdot q^{-r_{\rm h}} \cdot q^{j_1(T-r_{\rm v})} \cdot \sum_{\ell=1}^{s} \binom{T}{r_{j_\ell}+1-T} \\
&\le \frac{q}{q-1} \cdot q^{-r_{\rm h}} \cdot q^{T-r_{\rm v}} \cdot \sum_{\ell=1}^{s} \binom{T}{r_{j_\ell}+1-T} \\
&\le \frac{q}{q-1} \cdot q^{-r_{\rm h}} \cdot q^{-r_{\rm v}} \cdot q^{T} \cdot (2^T-1) .
\end{aligned}$$

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 (16). $\square$

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

[6] 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.

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

[8] 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.

[9] 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.

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

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

[12] R.J. McEliece, L. Swanson, *On the decoder error probability for Reed-Solomon codes, IEEE Trans. Inform. Theory,* 32 (1986), 701–703.

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

[14] R.M. Roth, G. Seroussi, *Reduced-redundancy product codes for burst error correction,* HP Laboratories Technical Report HPL-97-25, January 1997.

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

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

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