

Joint Decoding of Content-Replication Codes for Flash Memories

Qing Li*, Huan Chang*, Anxiao (Andrew) Jiang*, and Erich F. Haratsch[†]

[†] Seagate Technology, San Jose, CA, 95131

* Computer Sci. and Eng. Dept., Texas A&M University, College Station, TX, 77843

{qingli, hchang, ajiang}@cse.tamu.edu

Abstract—One serious challenge for flash memories is data reliability. In this work, we present the content-replication codeword problem, and it leads to our proposed joint decoder. We focus on joint decoding algorithms and study their theoretical decoding performances. The proposed scheme is novel for flash memories, and we show their reliability can be enhanced by increasing the diversity of error-correcting codes.

I. INTRODUCTION

One challenge for flash memories is the data reliability as several types of noise [1], [5] exist. Besides strong error correcting codes, e.g., LDPC codes, another mechanism to protect flash memories is *memory scrubbing* [10], i.e., while errors accumulate in a codeword, with the next block erasure, the codeword is corrected and a new error-free codeword is written back to the memory. However, in flash memory rewrites are made in an *out-of-place* fashion, i.e., an updated codeword is stored at a new physical address and the original codeword remains in the memory. Those mechanisms can lead to multiple copies of codewords, i.e., the *content-replicated codeword* problem. In addition to memory scrubbing, other factors also may cause the content-replication problem such as garbage collection, wear-leveling, etc, and it is estimated that on average $3 \sim 13$ (i.e., the exact number depends on the workload traffic and various Flash Translation Layer algorithms [2] used) copies of content-replicated codewords can be generated [3].

In this work, we enhance flash memory reliability by utilizing the existence of *two* content-replicated codewords for decoding, including an old codeword and a new codeword storing the same information. We aim at designing a *joint decoding* scheme having access to both content-replicated codewords, and explore its decoding performance. This leads to reliability improvement in flash memories. We further study a new paradigm where the two content-replicated codewords have different forms for better performance. The significance of this paper is two-fold: on the practical side, the new coding scheme utilizes the unique properties of flash memories; on the theoretical side, we show that increasing the diversity of error-correcting codes in the storage system

can improve the reliability of replicated data even if there exist constraints in their joint decoding algorithms.

II. PROBLEM STATEMENT

Let $\mathcal{D} = \{0, 1, \dots, M - 1\}$ be the message set for $M \in \mathbb{N}$, and let \mathcal{X} and \mathcal{Y} be two alphabets of the symbols stored in a cell. Let two encoders be $f_1 : \mathcal{D} \rightarrow \mathcal{X}^N$ and $f_2 : \mathcal{D} \rightarrow \mathcal{X}^N$, and the desired joint decoder be $h : \mathcal{Y}^N \times \mathcal{Y}^N \rightarrow \mathcal{D}$, where N is the length of codewords. Let $\mathbb{P} = (\mathcal{X}, \mathcal{Y}, \mathcal{P}_{Y|X})$ and $\mathbb{Q} = (\mathcal{X}, \mathcal{Y}, \mathcal{Q}_{Y|X})$ be two independent channels.

We illustrate the model in Fig. 1. Here, m is a common message to both encoders, the N -dimensional vectors $x_0^{N-1}(1), x_0^{N-1}(2) \in \mathcal{X}^N$ are two codewords obtained through two encoders (those encoders are not necessarily identical), and $y_0^{N-1}(1), y_0^{N-1}(2)$ are two noisy codewords through \mathbb{P} and \mathbb{Q} . The task is to design a joint decoder to give a *reliable* estimation of the message m , which is denoted as \hat{m} , giving $y_0^{N-1}(1)$ and $y_0^{N-1}(2)$.

The problem statement is presented below:

Definition 1. Given two $(N, 2^{NR})$ error-correcting codes, a message set $\mathcal{D} = \{0, 1, \dots, 2^{NR} - 1\}$, their encoding functions $f_1 : \mathcal{D} \rightarrow \mathcal{X}^N$ and $f_2 : \mathcal{D} \rightarrow \mathcal{X}^N$, and two independent channels \mathbb{P} and \mathbb{Q} , the task is to design a joint decoding scheme $h : \mathcal{Y}^N \times \mathcal{Y}^N \rightarrow \mathcal{D}$ such that $Pr(h(y_0^{N-1}(1), y_0^{N-1}(2)) \neq i | x_0^{N-1}(1) = f_1(i), x_0^{N-1}(2) = f_2(i)) \rightarrow 0$ for $i \in \mathcal{D}$ as $N \rightarrow \infty$.

We point out two implicit requirements for the joint decoder in the above definition: the first is the rate of the given code should be larger than the capacities of underlying two channels, i.e., $R > C(\mathbb{P})$ and $R > C(\mathbb{Q})$, therefore reliable decoding is impossible for separate decoders, i.e., $\nexists g_1 : \mathcal{Y}^N \rightarrow \mathcal{D}$ and $\nexists g_2 : \mathcal{Y}^N \rightarrow \mathcal{D}$ such that $Pr(g_1(y_0^{N-1}(1)) \neq i \text{ or } g_2(y_0^{N-1}(2)) \neq i | x_0^{N-1}(1) = f_1(i), x_0^{N-1}(2) = f_2(i)) \rightarrow 0$ for $i \in \mathcal{D}$ as $N \rightarrow \infty$. Otherwise, the joint decoder degenerates to the separate decoder in channel coding model; the second requirement is for the given encoders, when channels are not degrading too much, reliable separate decoders exist. More precisely, given the same parameters N, R , we require $f_1(\cdot)$ and $f_2(\cdot)$ meet the condition that when $R < C(\mathbb{P}_1)$ and $R < C(\mathbb{Q}_1)$ for some \mathbb{P}_1 and \mathbb{Q}_1 ,

there exist $g_1 : \mathcal{Y}^N \rightarrow \mathcal{D}$ and $g_2 : \mathcal{Y}^N \rightarrow \mathcal{D}$ such that $Pr(g_1(y_0^{N-1}(1)) \neq i, g_2(y_0^{N-1}(2)) \neq i | x_0^{N-1}(1) = f_1(i), x_0^{N-1}(2) = f_2(i)) \rightarrow 0$ for $i \in \mathcal{D}$ as $N \rightarrow \infty$.

The above requirements are due to the motivations of joint decoders: the joint decoder is not to replace existing individual decoders (as it is possible that individual decoders suffice to reliably decode when channels do not degrade too much, and also the content-replicated codewords cannot always be guaranteed to exist) but to replace individual decoders when they fail. It is also those requirements that differentiate the joint decoder from other coding models like Multiple Access Channels with correlated sources by Slepian and Wolf [11] and Fountain code [6].

In the following, we assume \mathbb{P} and \mathbb{Q} are identical Binary Erasure Channels (BEC) in Section III and identical Additive White Gaussian Noise (AWGN) channels in Section IV, and both encoders are *systematic* LDPC encoders. The following notations will be used: let the rate of two *systematic* LDPC codes be $\frac{K}{N}$, let $\mathbf{G}_1, \mathbf{G}_2$ be the encoding matrices, and $\mathbf{H}_1, \mathbf{H}_2$ denotes their parity check matrices. Let $y_0^{N-1}(1), y_0^{N-1}(2) \in \{0, 1, ?\}^N$ be two codewords received for BECs and let $y_0^{N-1}(1), y_0^{N-1}(2) \in \mathbb{R}^N$ also be those for AWGN channels.

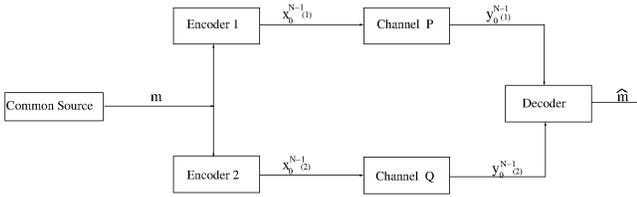


Fig. 1. Illustration of joint decoding content-replicated codewords.

III. JOINT DECODER FOR BECS

In this section, we present several joint decoder designs when \mathbb{P} and \mathbb{Q} are Binary Erasure Channels with the same parameter.

A. Joint decoder for identical content-replicated codes

The given codes are *identical* in this case, i.e., $\mathbf{G}_1 = \mathbf{G}_2$ and $\mathbf{H}_1 = \mathbf{H}_2$.

Given $y_0^{N-1}(1)$ and $y_0^{N-1}(2)$, a *combined* codeword y_0^{N-1} is obtained as follows, for $i = 0, 1, \dots, N-1$, $y_i =$

$$\begin{cases} ? & \text{if } y_i(1) = y_i(2) = ?, \\ y_i(1) & \text{if } y_i(2) = ? \text{ and } y_i(1) \neq ?, \\ y_i(2) & \text{else} \end{cases}$$

The parity check matrix for y_0^{N-1} is \mathbf{H}_1 . The decoding result is obtained by applying belief propagation to y_0^{N-1} with \mathbf{H}_1 and initial erasure probability ϵ^2 .

Let $\lambda(x)$ and $\rho(x)$ be degree distributions for the LDPC codes used, let $\epsilon^{BP}(\lambda, \rho)$ be its original threshold

as in [8], and $\epsilon_{iden}^{BP}(\lambda, \rho)$ be the threshold for our joint decoder. The comparison of $\epsilon_{iden}^{BP}(\lambda, \rho)$ and $\epsilon^{BP}(\lambda, \rho)$ for some regular LDPC codes is presented in the second and the third columns of Table I, and we have $\epsilon_{iden}^{BP} > \epsilon^{BP}$.

Note that the above scheme can be generalized to cases when \mathbb{P} and \mathbb{Q} are with different ϵ , and due to space limitation we do not present that here.

TABLE I
COMPARISON OF ϵ^{BP} , ϵ_{iden}^{BP} AND ϵ_{dif}^{BP}

(d_v, d_c)	ϵ^{BP}	ϵ_{iden}^{BP}	ϵ_{dif}^{BP}
(3,4)	0.6474	0.8046	0.8741
(3,5)	0.5176	0.7194	0.7594
(3,6)	0.4294	0.6553	0.6600
(4,6)	0.5061	0.7114	0.7335
(4,8)	0.3834	0.6192	0.5814

B. Joint decoder of different content-replicated codes

In the above subsection, the two codes are identical, which are effectively repetition codes, and this motivates us to explore another joint decoder design when the two encoders are different.

1) *Joint decoder design*: The given codes are *different* in this case, i.e., $\mathbf{G}_1 \neq \mathbf{G}_2$ and $\mathbf{H}_1 \neq \mathbf{H}_2$, but codewords carry identical systematic information bits, that is, two encoding functions are $x_0^{N-1}(1) = u_0^{K-1} \mathbf{G}_1$ and $x_0^{N-1}(2) = u_0^{K-1} \mathbf{G}_2$.

Let $\mathcal{I}_1, \mathcal{I}_2 \subseteq \{0, 1, \dots, N-1\}$ be the information bit index sets for $y_0^{N-1}(1)$ and $y_0^{N-1}(2)$, and let \mathcal{P}_1 and \mathcal{P}_2 be their parity check bit index sets. Let $y_0^{N-1}(1)_{\mathcal{I}_1} = (y_i(1) : i \in \mathcal{I}_1)$, i.e., information bits of $y_0^{N-1}(1)$, and similar notations apply to $y_0^{N-1}(2)_{\mathcal{I}_2}$, $y_0^{N-1}(1)_{\mathcal{P}_1}$ and $y_0^{N-1}(2)_{\mathcal{P}_2}$. Let $g(\cdot) : \mathcal{I}_1 \rightarrow \mathcal{I}_2$ be a one-to-one mapping such that $x_i(1) = x_{g(i)}(2)$ for $i \in \mathcal{I}_1$. Similar to the previous section, we define $(y_0^{N-1})_{\mathcal{I}_1}$, where $y_i =$

$$\begin{cases} ? & \text{if } y_i(1) = y_{g(i)}(2) = ?, \\ y_i(1) & \text{if } y_{g(i)}(2) = ? \text{ and } y_i(1) \neq ?, \\ y_{g(i)}(2) & \text{else} \end{cases}$$

Then, a constructed *combined* codeword is $y_0^{2N-K-1} = [(y_0^{N-1})_{\mathcal{I}_1}, y_0^{N-1}(1)_{\mathcal{P}_1}, y_0^{N-1}(2)_{\mathcal{P}_2}]$. That is, y_0^{2N-K-1} is constructed by appending information bits from $y_0^{N-1}(1)$ and $y_0^{N-1}(2)$ after preprocessing, and parity check bits from $y_0^{N-1}(1)$ and $y_0^{N-1}(2)$.

Let $\mathbf{H}_1 = [\mathbf{H}_{1,0}, \mathbf{H}_{1,1}, \dots, \mathbf{H}_{1,N-1}]$, let $\mathbf{H}_{1,\mathcal{I}_1} = [\mathbf{H}_{1,i} : i \in \mathcal{I}_1]$, and let $\mathbf{H}_{1,\mathcal{P}_1} = [\mathbf{H}_{1,i} : i \in \mathcal{P}_1]$. Similarly, we divide \mathbf{H}_2 into $\mathbf{H}_{2,\mathcal{I}_2}$ and $\mathbf{H}_{2,\mathcal{P}_2}$. Then, the parity check matrix \mathbf{H} for y_0^{2N-K-1} is of the form in Fig. 2.

An example for a combined codeword and its parity check matrix is illustrated in Fig. 3. In Fig. 3, (a) is the Tanner graph and \mathbf{H}_1 for $y_0^{N-1}(1)$, where information bits are black and parity check bits are red; (b) is the Tanner graph and \mathbf{H}_2 for $y_0^{N-1}(2)$, where information bits are black and parity check bits are green; (c) is the

$$\mathbf{H} = \begin{array}{c|cc|c} & \text{K} & \text{N-K} & \text{N-K} \\ \hline \text{N-K} & \mathbf{H}_{1,I} & \mathbf{H}_{1,P} & \mathbf{0} \\ \hline \text{N-K} & \mathbf{H}_{2,I} & \mathbf{0} & \mathbf{H}_{2,P} \\ \hline \end{array}$$

Fig. 2. Illustration of the parity check matrix \mathbf{H} .

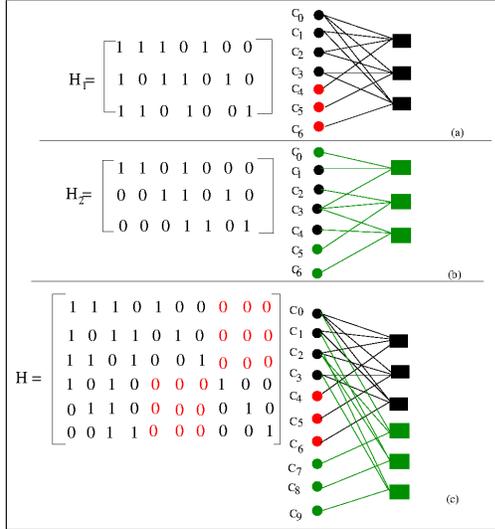


Fig. 3. Illustration of constructed y_0^{2N-K-1} and \mathbf{H} .

constructed Tanner graph and \mathbf{H} based on (a) and (b), where information bits are black, parity check bits from $y_0^{N-1}(1)$ are red, and parity check bits from $y_0^{N-1}(2)$ are green.

The decoding result is obtained by applying belief propagation to y_0^{2N-K-1} with \mathbf{H} , the initial erasure probability ϵ^2 for $(y_0^{N-1})_{\mathcal{I}_1}$, and ϵ for $y_0^{N-1}(1)_{\mathcal{P}_1}$ and $y_0^{N-1}(2)_{\mathcal{P}_2}$.

2) Performance analysis by density evolution:

a) *Notations:* In a Tanner graph of an LDPC code, for an edge if its one end connects to an information bit of variable nodes, we call it *information edge*; if it connects to a parity check bit of variable nodes, we call it *parity edge*. For example, in Fig. 3 (c), the edges connecting to c_0, c_1, c_2, c_3 are information edges and the remaining edges are parity edges.

For information edges (resp. parity edges), let $\lambda_i^{(i)}$ (resp. $\lambda_i^{(p)}$) be the fraction of edges connecting to an variable node with degree i . Let $\lambda^{(i)}(x) = \sum_{i=1}^{d_v} \lambda_i^{(i)} x^{i-1}$,

where $\sum_{i=1}^{d_v} \lambda_i^{(i)} = 1$, and $\lambda^{(p)}(x) = \sum_{i=1}^{d_v} \lambda_i^{(p)} x^{i-1}$, where $\sum_{i=1}^{d_v} \lambda_i^{(p)} = 1$, be the degree distribution functions from the edge perspective. For example, $\lambda^{(i)}(x) = \frac{6}{15}x^2 + \frac{4}{15}x^3 + \frac{5}{15}x^5$ and $\lambda^{(p)}(x) = 1$ in Fig. 3 (c).

Let $\rho_{j,k}$ be the fraction of edges connecting to a

check node with degree $j+k$, of which j edges are information edges and k edges are parity edges. Let $\rho(x, y) = \sum_{j,k} \rho_{j,k} x^j y^k$, where $\sum_{j,k} \rho_{j,k} = 1$, denote the edge degree distribution functions from the check node perspective. For example, $\rho(x, y) = \frac{12}{21}x^3y + \frac{9}{21}x^2y$ in Fig. 3 (c).

Let $\rho_{j,k}^{(p)} = \frac{\rho_{j,k}}{1-\rho_{0,j+k}}$ and $\rho_{j,k}^{(i)} = \frac{\rho_{j,k}}{1-\rho_{j+k,0}}$, let $\rho^{(i)}(y, x) = \sum_{j,k} \rho_{j,k}^{(i)} x^{j-1} y^k$ where $\sum_{j,k} \rho_{j,k}^{(i)} = 1$ and $j \geq 1, k \geq 0$, and $\rho^{(p)}(x, y) = \sum_{j,k} \rho_{j,k}^{(p)} x^j y^{k-1}$

where $\sum_{j,k} \rho_{j,k}^{(p)} = 1$ and $j \geq 0, k \geq 1$. For example,

$\rho^{(i)}(x, y) = \frac{12}{21}x^2y + \frac{9}{21}xy$ and $\rho^{(p)}(x, y) = \frac{12}{21}x^3 + \frac{9}{21}x$ in Fig. 3 (c), where $\rho^{(p)}(x, y)$ happens to be the same as $\rho^{(i)}(x, y)$ for this example.

b) Edge degree distributions:

Lemma 2. Given two regular (d_v, d_c) LDPC codes (which are not necessarily the same), the edge degree distributions of constructed the *combined* LDPC code are: $\lambda^{(i)}(x) = x^{2d_v-1}$, $\lambda^{(p)}(x) = x^{d_v-1}$, and $\rho_{j,k} = \binom{j+k}{j} \left(\frac{d_c-d_v}{d_c}\right)^j \left(\frac{d_v}{d_c}\right)^k$, where $j+k = d_c$.

Proof: Based on the construction presented, for the Tanner graph of y_0^{2N-K-1} , both check nodes of $y_0^{N-1}(1)$ and $y_0^{N-1}(2)$ connect to information bits of variable nodes of y_0^{2N-K-1} , thus those node degrees are doubled; The degree of parity check bit of variables nodes of y_0^{2N-K-1} remains the same as those of $y_0^{N-1}(1)$ and $y_0^{N-1}(2)$.

The result for $\rho_{j,k}$ follows from that for a random edge it is an information edge with probability $\frac{d_c-d_v}{d_c}$, a parity edge with probability $\frac{d_v}{d_c}$, and the probability distribution that j out of $j+k$ edges are from information edges is a binomial distribution. ■

c) *Density evolution:* From Lemma 2, we know that $\lambda^{(i)}(x)$ and $\lambda^{(p)}(x)$ are not identical, the initial effective erasure probability is ϵ^2 for information bits of y_0^{2N-K-1} and ϵ for parity bits of y_0^{2N-K-1} , thus the probabilities of a parity bit and an information bit being an erasure at the l -round of belief propagation decoding are not the same (we show this point in Fig.4 through a simulation with both (3,6) LDPC code and initial erasure probability 0.6).

Let $x_i^{(l)}$ be the average probability of an information bit of y_0^{2N-K-1} being an erasure after the l -round of belief propagation decoding, and similarly let $x_p^{(l)}$ be that for a parity check bit of y_0^{2N-K-1} .

Our main result based on density evolution [8] is presented below:

Theorem 3. For our joint decoding of different content-replicated codes, the average erasure probabilities after

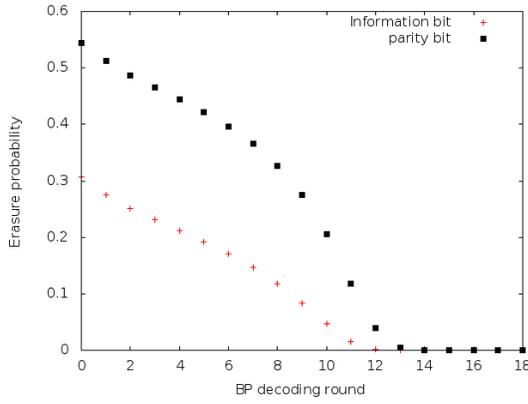


Fig. 4. Density evolution comparison of information bits and parity bits for joint decoder of different content-replicated (3, 6) LDPC codes with initial erasure probability 0.6.

l -round of belief-propagation decoding are given by

$$x_i^{(l)} = \epsilon^2 \lambda^{(i)}(1 - \rho^{(i)}(1 - x_p^{(l-1)}, 1 - x_i^{(l-1)})),$$

$$x_p^{(l)} = \epsilon \lambda^{(p)}(1 - \rho^{(p)}(1 - x_i^{(l-1)}, 1 - x_p^{(l-1)})),$$

where $\lambda^{(i)}(\cdot)$, $\lambda^{(p)}(\cdot)$, $\rho^{(i)}(\cdot)$ and $\rho^{(p)}(\cdot)$ are for the combined LDPC code.

Proof: We break the proof into two steps.

First, let $y_i^{(l)}$ be the average probability of being an erasure under belief-propagation decoding after l rounds for an output edge from a check node to an information bit of variable node. It is given by $y_i^{(l)} = \sum_{j,k} \rho_{j,k}^{(i)}(1 - (1 - x_i^{(l)})^{j-1}(1 - x_p^{(l)})^k) = 1 - \rho^{(i)}(1 - x_p^{(l)}, 1 - x_i^{(l)})$.

Similarly, let $y_p^{(l)}$ be the average probability of erasure under belief-propagation decoding after l -round for an output edge from a check node to a parity check bit of variable node. It is given by $y_p^{(l)} = 1 - \rho^{(p)}(1 - x_i^{(l)}, 1 - x_p^{(l)})$.

Second, the average probability of erasure for the output message of an information bit of variable nodes is given by $x_i^{(l)} = \epsilon^2 \sum_i \lambda_i^{(i)}(y_i^{(l-1)})^{i-1} = \epsilon^2 \lambda^{(i)}(y_i^{(l-1)})$.

Similarly, the average probability of erasure for the output message of an parity check bit of variable nodes is given by $x_p^{(l)} = \epsilon \lambda^{(p)}(y_p^{(l-1)})$.

Combining the above two steps, we obtain the desired results. ■

The following theorem presents us the existence of density evolution threshold.

Theorem 4. Based on Theorem 3, one sees density evolution updates are given by $f_i(\epsilon, x, y) = \epsilon^2 \lambda^{(i)}(1 - \rho^{(i)}(1 - y, 1 - x))$ and $f_p(\epsilon, x, y) = \epsilon \lambda^{(p)}(1 - \rho^{(p)}(1 - x, 1 - y))$. We observe the following:

- 1) $f_i(\epsilon, x, y)$ and $f_p(\epsilon, x, y)$ are non-decreasing in all arguments for $\epsilon, x, y \in [0, 1]$ and strictly increasing if $\epsilon, x, y \in (0, 1)$.
- 2) For any $x_0, y_0, \epsilon \in [0, 1]$, the sequence $x_{l+1} =$

$f_i(\epsilon, x_l, y_l)$ and $y_{l+1} = f_p(\epsilon, x_l, y_l)$ are monotonic in l .

- 3) Let $x_{l+1}(\epsilon)$ and $y_{l+1}(\epsilon)$ be defined recursively by $x_{l+1}(\epsilon) = f_i(\epsilon, x_l(\epsilon), y_l(\epsilon))$, $y_{l+1}(\epsilon) = f_p(\epsilon, x_l(\epsilon), y_l(\epsilon))$, $x_0(\epsilon) = \epsilon^2$ and $y_0(\epsilon) = \epsilon$. Then, $x_{l+1}(\epsilon)$ and $y_{l+1}(\epsilon)$ are non-decreasing in ϵ .
- 4) The function $x_\infty(\epsilon) = \lim_{l \rightarrow \infty} x_l(\epsilon)$ and $y_\infty(\epsilon) = \lim_{l \rightarrow \infty} y_l(\epsilon)$ exist and are non-decreasing for all $\epsilon \in [0, 1]$.

Proof: For 1), we observe that $\frac{d}{d\epsilon} f_i(\epsilon, x, y) = 2\epsilon \lambda^{(i)}(1 - \rho^{(i)}(1 - y, 1 - x))$ is not negative for $\epsilon, x, y \in [0, 1]$, and $\frac{d}{d\epsilon} f_p(\epsilon, x, y) = \lambda^{(p)}(1 - \rho^{(p)}(1 - x, 1 - y))$ are positive for $x, y \in [0, 1]$. $\frac{d}{dx} f_i(\epsilon, x, y) = \epsilon^2 \lambda^{(i)'}(1 - \rho^{(i)}(1 - y, 1 - x)) \rho^{(i)'}(1 - y, 1 - x)$ is positive for $\epsilon, x, y \in (0, 1)$ and $\frac{d}{dx} f_p(\epsilon, x, y) = \epsilon \lambda^{(p)'}(1 - \rho^{(p)}(1 - x, 1 - y)) \rho^{(p)'}(1 - x, 1 - y)$ is also positive for $\epsilon, x, y \in (0, 1)$. Similarly, we can prove $\frac{d}{dy} f_i(\epsilon, x, y)$ and $\frac{d}{dy} f_p(\epsilon, x, y)$ are also positive for $\epsilon, x, y \in (0, 1)$.

For 2), the monotonicity of $f_i(\epsilon, x, y)$ and $f_p(\epsilon, x, y)$ implies that $x_{l+1} = f_i(\epsilon, x_l, y_l) \stackrel{\geq}{\leq} x_l$ and $x_{l+2} = f_i(\epsilon, x_{l+1}, y_{l+1}) \stackrel{\geq}{\leq} x_{l+1}$. Therefore, monotonicity holds inductively and the direction of x_l depends only on the first step. Similarly, we can prove $y_{l+1} = f_p(\epsilon, x, y)$ are monotonic.

For 3), we first observe that $x_0(\epsilon)$ and $y_0(\epsilon)$ are non-decreasing in ϵ . Next, we proceed by induction, for any $\epsilon \leq \epsilon'$, to see that $x_{l+1}(\epsilon) = f_i(\epsilon, x_l(\epsilon), y_l(\epsilon)) \leq f_i(\epsilon', x_l(\epsilon'), y_l(\epsilon')) = x_{l+1}(\epsilon')$. Similarly, we can prove that $y_{l+1}(\epsilon)$ is non-decreasing in ϵ .

For 4), the limit exists because 2) implies the sequence $x_l(\epsilon)$ is monotonic and bounded for all $\epsilon \in [0, 1]$. The limit function is non-decreasing because 3) implies that, for any $\epsilon \leq \epsilon'$, we have $x_\infty(\epsilon) = \lim_{l \rightarrow \infty} x_l(\epsilon) \leq \lim_{l \rightarrow \infty} x_l(\epsilon') = x_\infty(\epsilon')$. The same process applies for the sequence $y_l(\epsilon)$. ■

Let $\epsilon_{diff}^{BP}(\lambda^{(i)}, \rho^{(i)}) = \sup\{\epsilon \in [0, 1] : x_\infty(\epsilon) = 0\}$ (which is clearly equal to $\sup\{\epsilon \in [0, 1] : y_\infty(\epsilon) = 0\}$) be the threshold defined by the density evolution. We compute $\epsilon_{diff}^{BP}, \epsilon_{iden}^{BP}, \epsilon_{diff}^{BP}$, where ϵ_{diff}^{BP} is based on the recursive functions defined in Theorem 3, for some regular LDPC codes in the fourth column of Table I. Comparing with previous results, we can see that $\epsilon_{diff}^{BP} > \epsilon_{iden}^{BP}$ is possible.

C. Joint decoder of related content-replicated codes

1) *Related encoder design:* Let \mathbf{G}_3 be an intermediate systematic LDPC generator matrix with rate $\frac{1}{2}$. Similarly, let \mathcal{I}_i and \mathcal{P}_i denote the information bit index set and the parity check bit index set for codes with $\mathbf{G}_i, i = 1, 2, 3$. The encoding algorithm is below, where $(x_0^{N-1})_{\mathcal{P}_3}$ denotes the subvector $(x_i : i \in \mathcal{P}_3)$.

- 1) $f_1: x_0^{N-1}(1) = u_0^{K-1} \mathbf{G}_1$.

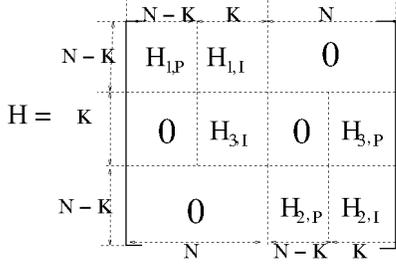


Fig. 5. Illustration of parity check matrix \mathbf{H}

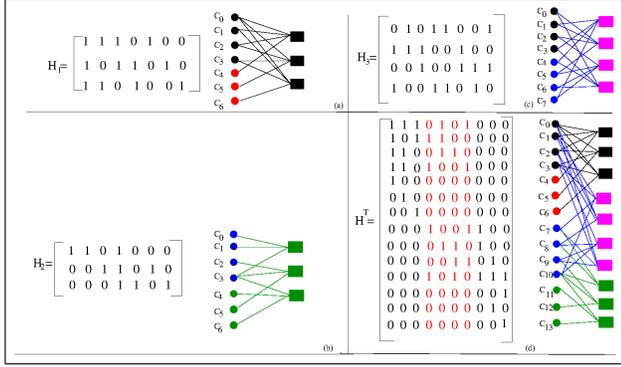


Fig. 6. Illustration of constructed y_0^{2N-1} and \mathbf{H} .

- 2) $v_0^{K-1} = (u_0^{K-1} \mathbf{G}_3) \mathcal{P}_3$.
- 3) $f_2: x_0^{N-1}(2) = v_0^{K-1} \mathbf{G}_2$,

where in the above f_1 and f_2 are the two encoders defined in Definition 1. That is, $(x_0^{N-1}(1))_{\mathcal{I}_1}$ and $(x_0^{N-1}(2))_{\mathcal{I}_2}$ are related through \mathbf{G}_3 .

2) *Joint decoder design:* A combined codeword is obtained by assembling $y_0^{N-1}(1)$ and $y_0^{N-1}(2)$ in the following way, $y_0^{2N-1} = (y_0^{N-1}(1))_{\mathcal{P}_1}, (y_0^{N-1}(1))_{\mathcal{I}_1}, (y_0^{N-1}(2))_{\mathcal{P}_2}, (y_0^{N-1}(2))_{\mathcal{I}_2}$.

Let \mathbf{H}_3 be the parity check matrix corresponding to \mathbf{G}_3 . Then, the parity check matrix \mathbf{H} for y_0^{2N-1} is of the form in Fig. 5.

An example for a combined codeword and its parity check matrix is illustrated in Fig. 6, where (a) is the Tanner graph and \mathbf{H}_1 for $y_0^{N-1}(1)$, where information bits are black and parity check bits are red; (b) is the Tanner graph and \mathbf{H}_2 for $y_0^{N-1}(2)$, where information bits are green and parity check bits are blue; (c) is the Tanner graph and \mathbf{H}_3 for v_0^{K-1} , where information bits are black and parity check bits are blue; (d) is the constructed Tanner graph and \mathbf{H} for y_0^{2N-1} .

The decoding result is obtained by applying belief propagation to y_0^{2N-1} with \mathbf{H} and initial erasure probability ϵ .

3) *Performance analysis by density evolution:* For $x_0^{N-1}(1)$ and $x_0^{N-1}(2)$ the two LDPC codes, we use same notations of previous subsection, $\lambda^{(i)}(x)$, $\lambda^{(p)}(x)$, $\rho^{(i)}(x, y)$ and $\rho^{(p)}(x, y)$, to denote the edge degree distributions from the variable nodes. For the intermediate LDPC code, v_0^{K-1} , we use the usual $\rho_3(x)$ and $\lambda_3(x)$

to denote its edge degree distributions.

We have the following results for density evolution, where we use the same notations, i.e., $x_p^{(l)}$ and $x_i^{(l)}$, as the previous subsection.

Theorem 5. For our joint decoding of related content-replicated codes, the average erasure probabilities after l -round of belief-propagation decoding are given by

$$\begin{aligned} x_p^{(l)} &= \epsilon \lambda^{(p)}(1 - \rho^{(p)}(1 - x_i^{(l-1)}, 1 - x_p^{(l-1)})), \\ x_i^{(l)} &= \epsilon^2 \lambda^{(i)}(1 - \rho^{(i)}(1 - x_p^{(l-1)}, 1 - x_i^{(l-1)})) \\ &\quad \cdot \lambda_3(1 - \rho_3(1 - x_i^{(l-1)})). \end{aligned}$$

where $\lambda^{(i)}(\cdot)$, $\lambda^{(p)}(\cdot)$, $\rho^{(i)}(\cdot)$ and $\rho^{(p)}(\cdot)$ are for the individual LDPC code, and $\lambda_3(\cdot)$ and $\rho_3(\cdot)$ are for the intermediate LDPC code.

Proof: The proof is the similar to that of the previous theorem, and thus we present its sketch as follows.

Let $y_i^{(l)}$ (resp. $y_p^{(l)}$) denote the average probability of being an erasure after the l th round of belief propagation decoding for an output edge from a check node of $y_0^{N-1}(1)$ or $y_0^{N-1}(2)$ to an information bit (resp. parity check bit) of variable node of y_0^{2N-1} . Clearly, they follow the same formulas as Theorem 3.

For $((y_0^{N-1}(1))_{\mathcal{I}_1}, (y_0^{N-1}(2))_{\mathcal{I}_2})$, let $y^{(l)}$ denote the probability that the message sent to a variable node is an erasure, and it is easy to know that $y^{(l)} = 1 - \rho_3(1 - x_i^{(l)})$.

Next, we focus on $x_i^{(l)}$, we know that an information bit of variable nodes receives both messages from parity bits of LDPC codes (ρ_1, λ_1) and (ρ_3, λ_3) , thus $x_i^{(l)} = \epsilon^2 \lambda^{(i)}(1 - \rho^{(i)}(1 - x_p^{(l-1)}, 1 - x_i^{(l-1)})) \cdot \lambda_3(1 - \rho_3(1 - x_i^{(l-1)}))$.

The equation for $x_p^{(l)}$ remains the same as that of the previous section, and the following conclusions hold immediately. ■

Let us verify one special case of Theorem 5: when $(d_{v'}, d_{c'})$ is $(1, 2)$ regular LDPC code, it should degenerate to the different content-replicated code case of Theorem 3, and this result coincides with this point.

Similarly, we obtain the following convergence results:

Theorem 6. Based on Theorem 5, one sees density evolution updates are given by $f_i(\epsilon, x, y) = \epsilon^2 \lambda^{(i)}(1 - \rho^{(i)}(1 - y, 1 - x)) \lambda_3(1 - \rho_3(1 - x))$ and $f_p(\epsilon, x, y) = \epsilon \lambda^{(p)}(1 - \rho^{(p)}(1 - x, 1 - y))$. We observe the following:

- 1) $f_i(\epsilon, x, y)$ and $f_p(\epsilon, x, y)$ are non-decreasing in all arguments for $\epsilon, x, y \in [0, 1]$ and strictly increasing if $\epsilon, x, y \in (0, 1)$.
- 2) For any $x_0, y_0, \epsilon \in [0, 1]$, the sequence $x_{l+1} = f_i(\epsilon, x_l, y_l)$ and $y_{l+1} = f_p(\epsilon, x_l, y_l)$ are monotonic in l .
- 3) Let $x_{l+1}(\epsilon)$ and $y_{l+1}(\epsilon)$ be defined recursively by $x_{l+1}(\epsilon) = f_i(\epsilon, x_l(\epsilon), y_l(\epsilon))$, $y_{l+1}(\epsilon) =$

$f_p(\epsilon, x_l(\epsilon), y_l(\epsilon))$, $x_0(\epsilon) = \epsilon^2$ and $y_0(\epsilon) = \epsilon$. Then, $x_{l+1}(\epsilon)$ and $y_{l+1}(\epsilon)$ are non-decreasing in ϵ .

- 4) The function $x_\infty(\epsilon) = \lim_{l \rightarrow \infty} x_l(\epsilon)$ and $y_\infty(\epsilon) = \lim_{l \rightarrow \infty} (y_l(\epsilon))$ exist and are non-decreasing for all $\epsilon \in [0, 1]$.

Let $\epsilon_{re}^{BP}(\lambda^{(i)}, \rho^{(i)}) = \sup\{\epsilon \in [0, 1] : x_\infty(\epsilon) = 0\}$ be the threshold defined by the density evolution. We calculate several ϵ_{re}^{BP} based on the recursive functions defined in Theorem 5 in Table II, where the first row indicates the regular LDPC for \mathbf{G}_3 , and the first column indicates the regular LDPC code for \mathbf{G}_1 and \mathbf{G}_2 . For example, the result 0.7976 is the threshold when LDPC codes for \mathbf{G}_1 and \mathbf{G}_2 are (4, 6) regular codes, and the intermediate LDPC code is (2, 4) code in Table II. From this table, we see that $\epsilon_{re}^{BP} > \epsilon_{dif}^{BP}$ is possible with appropriate \mathbf{G}_3 . That is the threshold can be improved by increasing the diversity of the underlying error-correcting codes.

TABLE II
CALCULATION OF ϵ_{re}^{BP}

(d_v, d_c)	(1,2)	(2,4)	(3,6)	(4,8)
(3,4)	0.8741	0.8918	0.8794	0.8754
(3,5)	0.7594	0.8169	0.7928	0.7771
(3,6)	0.6600	0.7569	0.7327	0.7085
(4,6)	0.7335	0.7976	0.772	0.7543
(4,8)	0.5814	0.7082	0.6917	0.662

IV. JOINT DECODERS FOR AWGN CHANNEL

In this section, we present the joint decoder designs for AWGN channel with the insight provided in previous sections. In the following, we assume that both \mathbb{P} and \mathbb{Q} are AWGN channels with the same parameters, let the rates of two LDPC codes still be $\frac{K}{N}$, let $\mathbf{G}_1, \mathbf{G}_2$ be the encoding matrices, and let $\mathbf{H}_1, \mathbf{H}_2$ denote their parity check matrices. Let $x_0^{N-1}(1)$ and $x_0^{N-1}(2)$ be all ones due to the channel symmetry, $y_0^{N-1}(1)$ and $y_0^{N-1}(2)$ are noisy codewords through \mathbb{P} and \mathbb{Q} , respectively, thus $y_i(1), y_i(2) \sim \mathcal{N}(1, \sigma^2)$ for $i = 0, \dots, N-1$.

A. Joint decoder of identical content-replicated codes

We first present the joint decoder design and its theoretical performance for the case when encoders are identical, i.e., $\mathbf{G}_1 = \mathbf{G}_2$ and $\mathbf{H}_1 = \mathbf{H}_2$.

Given noisy codewords $y_0^{N-1}(1), y_0^{N-1}(2) \in \mathbb{R}^N$ of the same codeword x_0^{N-1} , the log-likelihood-ratio (LLR) message from channel \mathbb{P} , denoted as $u_P(i)$, is $\ln \frac{p(y_i(1)|x_i=1)}{p(y_i(1)|x_i=0)} = \frac{2y_i(1)}{\sigma^2}$, i.e., Gaussian with mean $\frac{2}{\sigma^2}$ and variance $\frac{4}{\sigma^2}$, and so is for the LLR message from channel \mathbb{Q} , denoted as $u_Q(i)$. Therefore, by averaging of LLR messages from \mathbb{P} and \mathbb{Q} , we can obtain the combined LLR message as $u_0(i) = \frac{u_P(i) + u_Q(i)}{2}$, i.e., Gaussian with mean $\frac{2}{\sigma^2}$ and variance $\frac{2}{\sigma^2}$.

The decoding result is obtained by applying sum-product algorithm with \mathbf{H}_1 and initial LLR messages $u_0(i)$ for $i = 0, \dots, N-1$. That is, let v be a LLR message from a variable node (with initial LLR $u_0(i)$) to a check node, then $v = u_0(i) + \sum_{i=1}^{d_v-1} u_i$, where $u_i, i = 1, \dots, d_v-1$, are the incoming LLRs from the neighbors of the variable node except the check node that gets the message v , and u is updated by $\tanh(\frac{u}{2}) = \prod_{i=1}^{d_c-1} \tanh(\frac{v_i}{2})$, where $v_i, i = 1, \dots, d_c-1$, are the incoming LLRs from d_c-1 neighbors of a check node.

Let $u(l)$ be the average of LLRs sent to a variable node at the l -th round of sum-product decoding, let $\lambda(x)$ and $\rho(x)$ be degree distribution functions for the LDPC code used, and let $\sigma_{iden}^{BP}(\lambda, \rho) = \sup\{\sigma : u(l) \rightarrow \infty \text{ as } l \rightarrow \infty\}$ be the threshold for the joint decoder. $\sigma_{iden}^{BP}(\lambda, \rho)$ can be obtained through the methods provided by Fu [4], we compare it with $\sigma^{BP}(\lambda, \rho)$ in Table III, and we conclude $\sigma_{iden}^{BP}(\lambda, \rho) > \sigma^{BP}(\lambda, \rho)$.

B. Joint decoder for different content-replicated codes

In this part, we present the joint decoder design for different content-replicated codes. We use two (d_v, d_c) regular LDPC codes to simplify the analysis of decoding algorithm. The two content-replicated codes are different in this way, i.e., $\mathbf{G}_1 \neq \mathbf{G}_2, \mathbf{H}_1 \neq \mathbf{H}_2$.

1) *Joint decoder design:* Let $\mathcal{I}_1, \mathcal{I}_2, \mathcal{P}_1, \mathcal{P}_2$ and $g(\cdot)$ be the same notations as before, and a combined codeword is obtained as $y_0^{2N-K-1} = (y_{\mathcal{I}_1}, y_0^{N-1}(1)_{\mathcal{P}_1}, y_0^{N-1}(2)_{\mathcal{P}_2})$ with initial LLR from channel $u_{\mathcal{I}_1} \sim \mathcal{N}(\frac{2}{\sigma^2}, \frac{2}{\sigma^2})$ for $i \in \mathcal{I}_1$ (that is by combining LLRs from $y_0^{N-1}(1)_{\mathcal{I}_1}$ and $y_0^{N-1}(2)_{\mathcal{I}_2}$), and $u_{\mathcal{P}_1}, u_{\mathcal{P}_2} \sim \mathcal{N}(\frac{2}{\sigma^2}, \frac{4}{\sigma^2})$ for $i \in \mathcal{P}_1 \cup \mathcal{P}_2$. The decoding result is obtained by applying sum-product decoding algorithm on y_0^{2N-K-1} with \mathbf{H} demonstrated in Fig. 2 with $u_{\mathcal{I}_1}, u_{\mathcal{P}_1}$ and $u_{\mathcal{P}_2}$ specified as above.

2) *Theoretical performance analysis by density evolution:*

a) *Density Evolution:* For bits of y_0^{2N-K-1} , let $v_i^{(l)}$ be the average LLR from an information bit to parity nodes at the l -th round, and similarly let $v_p^{(l)}$ be that from a parity check bit of y_0^{2N-K-1} .

For a parity node connecting to j information edges and k parity edges, let $\mu_i^{(l)}(j, k)$ and $\mu_p^{(l)}(j, k)$ be its LLR sent to an information and a parity bit at l -round, respectively. Thus, we have

$$\begin{aligned} \mu_i^{(l)}(j, k) &= 2 \tanh^{-1} \left(\left(\tanh \frac{v_i^{(l)}}{2} \right)^{j-1} \left(\tanh \frac{v_p^{(l)}}{2} \right)^k \right) \\ \mu_p^{(l)}(j, k) &= 2 \tanh^{-1} \left(\left(\tanh \frac{v_i^{(l)}}{2} \right)^j \left(\tanh \frac{v_p^{(l)}}{2} \right)^{k-1} \right). \end{aligned} \quad (1)$$

Let $u_i^{(l)}$ be the average LLR from a parity nodes to an information bit at the l -round, and similarly let $u_p^{(l)}$ be that from a parity check node to a parity bit. Then, by averaging LLR from a check node to a information

and parity bit, we have

$$\begin{aligned} u_i^{(l)} &= \sum_{j=1}^{d_c} \rho_{j,k}^{(i)} \mu_i^{(l)}(j, k), \\ u_p^{(l)} &= \sum_{j=1}^{d_c} \rho_{j,k}^{(p)} \mu_p^{(l)}(j, k), \end{aligned} \quad (2)$$

where $\rho_{j,k}^{(i)}$ and $\rho_{j,k}^{(p)}$ are the same as previous sections.

Thus we obtain our main result of this subsection as below:

Theorem 7. For joint decoding of different content-replicated codes, the LLRs after the l -th round of sum-product decoding at the variable node are given by

$$\begin{aligned} v_i^{(l)} &= \mu_i^{(0)} + (2d_v - 1) \cdot u_i^{(l-1)}, \\ v_p^{(l)} &= \mu_p^{(0)} + (d_v - 1) \cdot u_p^{(l-1)}, \end{aligned}$$

where $\mu_i^{(0)}$ is the initial LLR for information bits of y_0^{2N-K-1} , and $\mu_p^{(0)}$ is that for parity bits.

Check nodes are updated as equation (2) and equation (1).

b) Approximate algorithm for density evolution:

For the calculation of density evolution of LDPC codes, there are several papers so far, such as [7], [9] and [4]. The method presented in [7] obtains thresholds with the Fourier transform, which is computationally intensive. The method presented in [9] obtains approximate thresholds for AWGN channels with sum-product decoding based on two assumptions of the LLR passed: one is their densities are approximately Gaussian when the channel is AWGN, and the other one is the so-called *symmetry condition* which requires a density function $f(x)$ to satisfy $f(x) = f(-x)e^x$ (by enforcing this condition for Gaussian with mean m and variance σ^2 , this condition reduces to $\sigma^2 = 2m$). Fu [4] pointed out that the Gaussian assumption does not always hold especially for LLR from check nodes.

For our analysis of density evolution, we turn to the method presented in [4] to obtain the approximate threshold for two reasons: one is the Gaussian assumption is invalid as pointed by Fu [4], and the other one is the *symmetry condition* property does not hold for our case, which is verified by intensive numerical calculations as shown in Fig. 7 (i.e., from this figure clearly the assumption that $\sigma^2 = 2m$ does not hold). Also the update rules stated by Theorem 7 and the initial samples of $\mu_i^{(0)}$ and $\mu_p^{(0)}$ are stationary (i.e., invariant with respect to the iteration number), thus those update rules preserve ergodicity. Therefore, based on the well-known property of ergodicity, i.e., any statistical parameter of the random process can be arbitrarily closely approximated by averaging over a sufficient number of samples, we have the following approximate algorithm for density evolution.

- 1) Step 0: choose a large number n , generate an initial

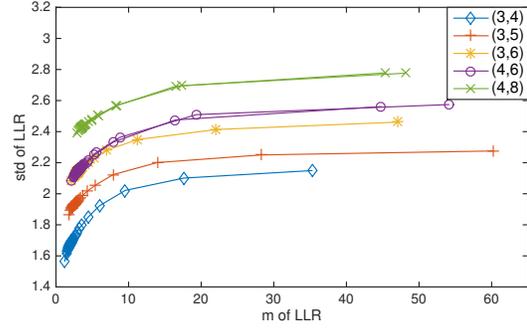


Fig. 7. m and σ^2 of LLR for joint decoding of different content-replicated codes.

n samples of $\mu_i^{(0)}$ according to $\mathcal{N}(2/\sigma^2, 2/\sigma^2)$, and similarly generate a n samples of $\mu_p^{(0)}$ according to $\mathcal{N}(2/\sigma^2, 4/\sigma^2)$.

- 2) Step 1 (for variable nodes): For iteration 0, copy $\mu_i^{(0)}$ to $v_i^{(l)}$ and copy $\mu_p^{(0)}$ to $v_p^{(l)}$ as shown by variable update formula of Theorem 7. For other iterations, take the n samples of $u_p^{(l-1)}$ and $u_i^{(l-1)}$ from the previous iteration, randomly interleave $(d_v - 1)$ samples $u_p^{(l)}$ and $(2d_v - 1)$ samples $u_i^{(l)}$, respectively. Then, update $v_i^{(l)}$ and $v_p^{(l)}$ by variable update formula Theorem 7.
- 3) Step 2 (for check nodes): For each iteration, take the n samples of $v_i^{(l)}$ and $v_p^{(l)}$ as calculated above. Randomly interleave $(d_c - 1)$ samples of them, and then compute the n samples of $u_i^{(l)}$ and $u_p^{(l)}$ based on equation (1) and check update formula Theorem 7.

c) Numerical results and analysis: Let $u(l)$ be the average of LLRs from a check node to a variable node at the l -th round of sum-product decoding, let $\lambda(x)$ and $\rho(x)$ be degree distribution functions for the LDPC code used, and define $\sigma_{diff}^{BP}(\lambda, \rho) = \sup\{\sigma : u(l) \rightarrow \infty \text{ as } l \rightarrow \infty\}$ be the threshold for our joint decoder. We calculate $\sigma_{diff}^{BP}(\lambda, \rho)$ based on the method presented above and compare it with $\sigma^{BP}(\lambda, \rho)$ and $\sigma_{iden}^{BP}(\lambda, \rho)$ in Table III. From the table we can see that it is possible that $\sigma_{diff}^{BP}(\lambda, \rho) > \sigma_{iden}^{BP}(\lambda, \rho)$.

C. Joint decoder for related content-replicated codes

1) *Joint decoder design:* Similar to the BEC case, an intermediate generator matrix \mathbf{G}_3 with rate 1/2 is used to connect two LDPC generator matrices \mathbf{G}_1 and \mathbf{G}_2 , and the encoding process is exactly the same as the BEC counterpart.

The decoding process is presented here: given $y_0^{N-1}(1)$ and $y_0^{N-1}(2)$, a *combined* codeword y_0^{2N-1} is constructed the same as before, i.e., $y_0^{2N-1} = (y_0^{N-1}(1)_{\mathcal{P}_1}, y_0^{N-1}(1)_{\mathcal{I}_1}, y_0^{N-1}(2)_{\mathcal{P}_2}, y_0^{N-1}(2)_{\mathcal{I}_2})$. The decoding result is obtained by applying sum-product decoding algorithm to y_0^{2N-1} with the parity check

matrix \mathbf{H} (constructed the same as Fig.5) and the initial LLR message $u_0 \sim \mathcal{N}(\frac{2}{\sigma^2}, \frac{4}{\sigma^2})$.

2) *Theoretical performance analysis by density evolution:*

a) *Density Evolution:* For density evolution, we assume that (d'_v, d'_c) regular LDPC code is used to connect two (d_v, d_c) regular LDPC codes.

For one (d_v, d_c) LDPC code, let $v_i^{(l)}$ be the average LLR from an information bit to its parity nodes (not the intermediate ones) at the l -round, and similarly let $v_p^{(l)}$ be that from a parity check bit. For a parity node of one (d_v, d_c) LDPC code, let $u_i^{(l)}$ and $u_p^{(l)}$ be its LLR sent to an information bit and a parity bit at l -round, respectively, and similarly their values can be expressed the same as equation (2).

For the intermediate (d'_v, d'_c) LDPC code, let $x^{(l)}$ be the average LLR sent to its parity nodes, and let $y^{(l)}$ be the average LLR sent to its variable nodes at the l -round of sum-product decoding. Thus we have

$$\begin{aligned} x^{(l)} &= \mu^{(0)} + d_v \cdot u_i^{(l-1)} + (d'_v - 1) \cdot y^{(l-1)}, \\ y^{(l)} &= 2 \tanh^{-1}(\tanh \frac{x^{(l)}}{2})^{d'_c-1}, \end{aligned}$$

where $\mu^{(0)}$ is the initial LLR for bits of y_0^{2N-1} .

We have the following result for the density evolution of our joint decoder:

Theorem 8. For joint decoding of related content-replicated codes (i.e., the two LDPC codes are both (d_v, d_c) LDPC codes and the intermediate LDPC code is an (d'_v, d'_c) LDPC code), the LLRs after the l -th round of sum-product decoding at the variable node are given by

$$\begin{aligned} v_i^{(l)} &= \mu^{(0)} + (d_v - 1) \cdot u_i^{(l-1)} + d'_v \cdot y_i^{(l-1)}, \\ v_p^{(l)} &= \mu^{(0)} + (d_v - 1) \cdot u_p^{(l-1)}, \end{aligned}$$

where $u_i^{(l)}$ and $u_p^{(l)}$ are updated as equation (2).

b) *Approximate algorithm for density evolution:*

We present the approximate algorithm for density evolution based on Theorem 8 below.

- 1) Step 0: choose a large number n , generate an initial n samples of $\mu^{(0)}$ according to $\mathcal{N}(2/\sigma^2, 4/\sigma^2)$.
- 2) Step 1 (for variable nodes): For iteration 0, copy $\mu^{(0)}$ to $v_i^{(0)}$, $x^{(0)}$ and $v_p^{(0)}$ as shown by variable update formula of Theorem 8 and equation (2). For other iterations, take the n samples of $u_p^{(l-1)}$, $u_i^{(l-1)}$ and $y^{(l-1)}$ from the previous iteration, randomly interleave $(d_v - 1)$ samples $u_p^{(l-1)}$, $u_i^{(l-1)}$ and $y^{(l-1)}$, respectively. Then, update $v_i^{(l)}$, $v_p^{(l)}$ and $x^{(l)}$ by variable update formula of Theorem 8 and equation (2).
- 3) Step 2 (for check nodes): For each iteration, take the n samples of $v_i^{(l)}$, $x^{(l)}$ and $v_p^{(l)}$ as calculated above. Randomly interleave the samples of them,

and then compute the n samples of $u_i^{(l)}$, $x^{(l)}$ and $u_p^{(l)}$ based on equation (2) and check update formula of Theorem 8 and equation (2).

c) *Numerical results and analysis:* Let $\sigma_{d'_v, d'_c}^{BP}(\lambda, \rho)$ be the threshold for our joint decoder with (λ, ρ) being degree distribution functions for our LDPC codes and (d'_v, d'_c) as the intermediate LDPC code, that is $\sigma_{d'_v, d'_c}^{BP}(\lambda, \rho) = \sup\{\sigma : u_i^{(l)} \rightarrow \infty \text{ as } l \rightarrow \infty\}$. We calculate $\sigma_{d'_v, d'_c}^{BP}(\lambda, \rho)$ based on the method presented above and compare it with $\sigma^{BP}(\lambda, \rho)$, $\sigma_{iden}^{BP}(\lambda, \rho)$ and $\sigma_{diff}^{BP}(\lambda, \rho)$ in the Table III. From the results, we can see that it is possible that $\sigma_{d'_v, d'_c}^{BP}(\lambda, \rho) > \sigma_{iden}^{BP}(\lambda, \rho)$ with appropriate (d'_v, d'_c) .

TABLE III
THRESHOLDS σ^* OF AWGN CHANNELS FOR JOINT DECODERS

(d_v, d_c)	σ^{BP}	σ_{iden}^{BP}	σ_{diff}^{BP}	$\sigma_{2,4}^{BP}$	$\sigma_{3,6}^{BP}$	$\sigma_{4,8}^{BP}$
(3,4)	1.261	1.555	1.69	1.655	1.5	1.45
(3,5)	1.004	1.264	1.379	1.462	1.267	1.201
(3,6)	0.880	1.116	1.19	1.38	1.161	1.085
(4,6)	1.002	1.242	1.3	1.386	1.207	1.145
(4,8)	0.838	1.044	1.065	1.3	1.091	1.007

REFERENCES

- [1] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Error patterns in MLC NAND flash memory: measurement, characterization, and analysis," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '12, Dresden, Germany, 2012, pp. 521–526.
- [2] T.-S. Chung, D.-J. Park, S. Park, D.-H. Lee, S.-W. Lee, and H.-J. Song, "A Survey of Flash Translation Layer," *J. Syst. Archit.*, vol. 55, no. 5-6, pp. 332–343, may 2009.
- [3] P. Desnoyers, "Analytic modeling of ssd write performance," in *International Systems and Storage conference (SYSTOR 2012)*, June 2012.
- [4] M. Fu, "On gaussian approximation for density evolution of low-density parity-check codes," in *Int. Conf. Communications*, Istanbul, May 2006.
- [5] Q. Li, A. Jiang, and E. F. Haratsch, "Noise Modeling and Capacity Analysis for NAND Flash Memories," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Honolulu, HI, June 2014, pp. 2262–2266.
- [6] D. J. C. MacKay, "Fountain codes," *IEE Proc.-Commun*, vol. 152, pp. 1062–1068, 2005.
- [7] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity check codes," *IEEE Transaction on Information Theory*, vol. 47, no. 2, pp. 619–637, February 2001.
- [8] T. J. Richardson and R. L. Urbanke, "The capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding," *IEEE Transaction on Information Theory*, vol. 47, no. 2, pp. 599–618, February 2001.
- [9] T. J. R. Sae-Yong Chung and R. L. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a gaussian approximation," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 657–670, February 2001.
- [10] A. M. Saleh, J. J. Serrano, and J. H. Patel, "Reliability of scrubbing recover-techniques for memory systems," *IEEE Transactions on reliability*, vol. 39, no. 3, pp. 114–122, April 1990.
- [11] D. Spleian and J. L. Wolf, "A Coding Theorem for Multiple Access Channels With Correlated Sources," in *The bell system technical journal*, September 1973, pp. 1037–1076.