

# On Minimizing Data-Read and Download for Storage-Node Recovery

Nihar B. Shah

**Abstract**—We consider the problem of efficient recovery of the data stored in any individual node of a distributed storage system, from the rest of the nodes. Applications include handling failures and degraded reads. We measure efficiency in terms of the amount of data-read and the download required. To minimize the download, we focus on the minimum bandwidth setting of the ‘regenerating codes’ model for distributed storage. Under this model, the system has a total of  $n$  nodes, and the data stored in any node must be (efficiently) recoverable from any  $d$  of the other  $(n-1)$  nodes. Lower bounds on the two metrics under this model were derived previously; it has also been shown that these bounds are achievable for the amount of data-read and download when  $d = n - 1$ , and for the amount of download alone when  $d \neq n - 1$ .

In this paper, we complete this picture by proving the converse result, that when  $d \neq n - 1$ , these lower bounds are strictly loose with respect to the amount of read required. The proof is information-theoretic, and hence applies to non-linear codes as well. We also show that under two (practical) relaxations of the problem setting, these lower bounds can be met for both read and download simultaneously.

## I. INTRODUCTION

Consider a distributed storage system with  $n$  storage nodes, each of which has a storage capacity of  $\alpha$  bits. Data of size  $B$  bits is to be stored across these nodes in a manner that the entire data can be recovered from any  $k$  of the  $n$  nodes. A problem that has received considerable attention in the recent past is that of efficient recovery of the data stored in an individual node, from the data stored in the remaining nodes in the system. This arises during handling of failures in distributed storage systems: upon failure of a node, it is replaced by a new node that must (efficiently) recover the data stored previously in the failed node from the remaining nodes in the system. A second application is that of degraded reads: if a node is busy or temporarily unavailable, then any request for the data stored in that node must be served (quickly) by downloading data from the remaining nodes.

We measure the efficiency of this process in terms of two metrics: the amount of data that must be read at the other nodes, and the amount of data downloaded from them. To optimize the amount of download, we consider the minimum bandwidth (MBR) setting of the regenerating codes model [1] for distributed storage. Under this model, recovery of the data stored in any individual node must be accomplished by connecting to any  $d$  ( $k \leq d < n$ ) other nodes and downloading  $\beta_D$  bits of data from each of them. Furthermore, under this model, these parameters must satisfy the condition

$$d\beta_D = \alpha . \quad (1)$$

An intuitive explanation of (1) is that recovery of the data stored in a node should entail only as much download as the amount stored. We shall assume throughout this paper that (1) is satisfied.

Under the MBR setting described above, a lower bound on the amount of download was derived in [1] as  $\beta_D \geq \frac{B}{kd - \binom{k}{2}}$ . It is easy to see that the amount of data that is read at a node is at least as much as the amount of data downloaded from that node.<sup>1</sup> It follows that the total amount of read  $\beta_R$  at any of the  $d$  nodes helping in the recovery must obey  $\beta_R \geq \beta_D$ , and hence is also lower bounded as  $\beta_R \geq \frac{B}{kd - \binom{k}{2}}$ .

In this paper, we investigate the existence of codes that satisfy the aforementioned lower bounds with equality, i.e., satisfy

$$\beta_D = \frac{B}{kd - \binom{k}{2}} , \quad (2)$$

$$\beta_R = \frac{B}{kd - \binom{k}{2}} \quad (3)$$

for the recovery of the data of any of the  $n$  nodes from any  $d$  other nodes in the system. It was shown previously in [2], [3] that when  $d = n - 1$ , the amount of download and read can simultaneously achieve (2) and (3) respectively for the recovery of the data of any individual node. Also, explicit codes with a download equalling (2) for all values of the parameters were constructed previously in [4]. However, it remained unknown whether or not the lower bound on the read (3) can also be matched along with that on the download (2) when  $d \neq n - 1$ .

We complete this picture by showing that under the MBR setting described above, when  $d \neq n - 1$ , it is *impossible* to construct codes that simultaneously satisfy (2) and (3) for the download and read respectively. The proof is information-theoretic, and allows us to conclude that these bounds cannot be met even with non-linear codes.

We also consider two (practical) relaxations of the problem setting, under which we provide explicit codes that can simultaneously achieve both (2) and (3) for all values of the system parameters. Under the setting described above, the data of any individual node must be recoverable from any  $d$  other nodes, with the download and read satisfying (2) and (3) respectively. The two relaxations respectively weaken the two “any” criteria with respect to the read. Under the first relaxation, we require the read to achieve (3) for only the recovery of the data stored in the systematic nodes (recovery of the data of the remaining nodes are allowed to have a larger read). This relaxed setting

The author is with the department of Electrical Engineering and Computer Sciences, University of California, Berkeley. E-mail: nihar@eecs.berkeley.edu

<sup>1</sup>The download may be smaller than the amount of read, since the data passed may be a (non-injective) function of the data that is read.

is relevant to the problem of degraded reads, where typically, the data stored in (only) the systematic nodes is of interest. Under the second relaxation, for the recovery of the data of any node, we require that (3) be achieved for the read from at least one set of  $d$  other nodes. The codes presented for both these relaxations are obtained by modifying the ‘product-matrix’ codes of [4].

We now take a brief digression to discuss a related notion, that of ‘repair-by-transfer’, which shall be called upon frequently in the paper. Observe that when (2) and (3) are satisfied, the amount of download  $\beta_D$  is equal to the amount of read  $\beta_R$ . As a result, whenever (2) and (3) are met, each of the  $d$  nodes helping in the recovery must simply pass a part of the data that it stores, without performing any computations. This is termed *repair-by-transfer* [3]. It follows that a repair-by-transfer code that satisfies (2) for the amount of download automatically achieves (3) for the read as well. Thus the problem considered in this paper can equivalently be stated as follows: *for the MBR setting described above, under what conditions is it possible to design a code that can perform repair-by-transfer with a download satisfying (2)?*

The rest of the paper is organized as follows. Section II describes related literature. Section III presents an information-theoretic proof showing the impossibility of achieving the previously derived lower bounds. Section IV considers (practical) relaxations of this setting, and provides explicit codes operating under these relaxations. Section V presents conclusions.

## II. RELATED LITERATURE

As described previously, explicit codes meeting (2) and (3) for recovery of the data of any node are presented in [2], [3] for the MBR setting when  $d = n - 1$ . The notion of ‘repair-by-transfer’ is also introduced therein. The repair-by-transfer codes of [2], [3] were subsequently extended to a more general but relaxed setting in [5]. In [5], the condition of efficiently recovering the data of an individual node from *any*  $d$  nodes is relaxed to doing so from specific subsets  $d$  nodes (termed ‘table-based’ repair), with respect to both the amount of read and the amount of download. In contrast, the relaxations presented subsequently in this paper make such relaxations only for the amount of read, and the amount of download continues to achieve (2) for every set of  $d$  nodes.

In addition to the MBR setting discussed above, the regenerating codes model of [1] has another setting associated to it: the minimum storage regeneration (MSR) setting. Under the MSR setting, the storage is required to be at an absolute minimum, and for this value of storage, the amount of download is optimized. The problem of minimizing read in the MSR setting is studied in [6]–[9]. In particular, MSR codes performing repair-by-transfer with a minimum download for the systematic nodes are constructed in [6]–[8]. A somewhat different setting called ‘functional’ repair is considered in [9] for the application of repair of failed nodes. In this setting, the node replacing a failed node may recover data that is different from what was stored in the failed node, but which retains certain desired properties. MSR codes performing functional repair-by-transfer with minimum download for all nodes are constructed in [9].

## III. IMPOSSIBILITY OF REPAIR-BY-TRANSFER IN MBR WHEN $d \neq n - 1$

It was shown previously in [2], [3] that when  $d = n - 1$ , both (2) and (3) can be achieved simultaneously. In this section, we present the converse to this result: we show that when  $d \neq n - 1$ , there cannot exist any code under which the data stored in any node can be recovered from any  $d$  other nodes while satisfying (2) and (3). This result encompasses both linear and non-linear codes. The proof may be skipped without any loss in continuity.

*Theorem 1:* Under the MBR setting, when  $d \neq n - 1$ , there cannot exist any code that performs repair-by-transfer of any node from any  $d$  other nodes with a download satisfying (2).

*Proof:* The proof proceeds via a contradiction. Let us suppose there exists such a code for some system parameters with  $d \neq n - 1$ . The proof is divided into three parts. First, it is shown that there exist (at least) three nodes that store (at least) one bit of data in common. Next, it is shown that for recovery of the data of any one of these nodes, the other two nodes must pass this bit. Finally, we show that under this condition, such an attempt of recovery must necessarily fail.

For  $i \in \{1, \dots, n\}$ , let  $W_i$  be a random variable corresponding to the data stored in node  $i$ . For recovering the data of node  $i$  from a set  $\mathcal{D}$  of  $d$  nodes, let  $\mathcal{D}S_j^i$  denote the random variable corresponding to the data passed by node  $j \in \mathcal{D}$  to node  $i$ . Let  $H(\cdot)$  denote Shannon entropy and  $I(\cdot; \cdot)$  the mutual information. Let  $\beta_D = \beta_R = \beta$ . In the proof, we shall employ the following four properties, established in [3], that any such code must satisfy.

- [3, Property 1]  $H(W_i) = \alpha$
- [3, Property 2]  $I(W_i; W_j) = \beta$
- [3, Property 3]  $H(\mathcal{D}S_j^i) = \beta$
- [3, Lemma 3]  $H(W_i | \mathcal{D}S_j^i) \leq (d - 1)\beta$ ,  
 $H(W_i | \mathcal{D}S_j^i, \mathcal{D}S_l^i) \leq (d - 2)\beta$ .

Consider recovery of the data stored in nodes  $\{1, \dots, d + 1\}$  (one at a time), from node  $n$  and  $(d - 1)$  other arbitrary nodes. In each case, node  $n$  passes a subset of  $\beta$  bits out of the  $\alpha (= d\beta)$  bits that it stores. We emphasize that due to the requirement of repair-by-transfer, the bits passed are simply *subsets* of those it stores (and do not arise from any computations on the stored bits). Node  $n$  thus passes a total of  $(d + 1)\beta$  bits. It follows from the pigeonhole principle that there exists at least one bit that occurs at least twice in this set of  $(d + 1)\beta$  bits. Moreover, [3, Property 3] implies that the  $\beta$  bits passed by a node, for recovery of the data of any other node, must all be distinct. Thus there must exist at least two nodes out of  $\{1, \dots, d + 1\}$  for which node  $n$  passes the same bit. Let us assume that these two nodes are nodes 1 and 2, and let  $b$  denote this common bit.

Since the data of any node  $i$  must be completely recovered, the quantity  $W_i$  is deterministic given the data passed by the  $d$  nodes in the recovery process. It follows from [3, Property 1] that the entropy of the  $d\beta (= \alpha)$  bits passed by the  $d$  nodes is  $\alpha$ . As a special case, it follows that  $H(b) = 1$ . From the description above, one can also see that that  $H(b|W_1) = 0$  and  $H(b|W_2) = 0$ . Moreover, since bit  $b$  was originally stored in node  $n$ ,  $H(b|W_n) = 0$ . Thus, the bit  $b$  is stored in nodes 1, 2 and  $n$ , and  $H(b) = 1$ .

Now consider recovering the data of node  $n$  from nodes  $\{1, \dots, d\}$ . We shall now show that nodes 1 and 2 must both pass bit  $b$ . Abbreviating our earlier notation, we let  $S_1$  and  $S_2$  be random variables corresponding to data passed by nodes 1 and 2 respectively. Thus, by definition, we have  $H(S_1|W_1) = H(S_2|W_2) = 0$ . From the properties discussed above, we get

$$\begin{aligned}
2\beta &= I(W_n; W_1) + I(W_n; W_2) \\
&\geq I(W_n; b, S_1) + I(W_n; b, S_2) \\
&= I(W_n; S_1) + H(b|S_1) - H(b|W_n, S_1) \\
&\quad + I(W_n; S_2) + H(b|S_2) - H(b|W_n, S_2) \\
&= I(W_n; S_1) + H(b|S_1) + I(W_n; S_2) + H(b|S_2) \\
&= 2H(W_n) - H(W_n|S_1) + H(b|S_1) - H(W_n|S_2) + H(b|S_2) \\
&= 2d\beta - H(W_n|S_1) + H(b|S_1) - H(W_n|S_2) + H(b|S_2) \\
&\geq 2d\beta - (d-1)\beta + H(b|S_1) - (d-1)\beta + H(b|S_2) \\
&= 2\beta + H(b|S_1) + H(b|S_2) .
\end{aligned}$$

Thus,  $H(b|S_1) = H(b|S_2) = 0$ . It follows that

$$\begin{aligned}
2\beta &= H(W_n) - (d-2)\beta \\
&\leq H(W_n) - H(W_n|S_1, S_2) \\
&= I(W_n; S_1, S_2) \\
&\leq H(S_1, S_2) \\
&\leq H(S_1, S_2, b) \\
&= H(S_2) + H(b|S_2) + H(S_1|S_2, b) \\
&\leq \beta + 0 + H(S_1|b) \\
&= \beta + H(b|S_1) + H(S_1) - H(b) \\
&= 2\beta - 1 .
\end{aligned} \tag{4}$$

Clearly, (4) yields a contradiction. ■

#### IV. EXPLICIT CODES FOR TWO RELAXATIONS

Repair-by-transfer under the regenerating codes setting described above amounts to (efficiently) recovering the contents of *any* failed node from *any* of the  $d$  nodes. We saw in the previous section that the bounds of (2) and (3) *cannot* be achieved simultaneously when  $d \neq n-1$ . Thus in this section, we consider two relaxations to this setup, which shall allow us to achieve these bounds. The two relaxations are obtained by slackening the two instances of the quantifier “any” for the amount of read. Note that under both relaxations, we shall continue to impose the requirements of the MBR setting, i.e., of recovering the entire data from any  $k$  nodes, and satisfying (1) and (2) on the amount of download for recovery of the data of any node from any  $d$  nodes.

##### A. Optimal recovery for systematic nodes

A systematic code is defined as one under which some  $k$  out of the  $n$  nodes store data in a raw (uncoded) form. These  $k$  nodes are called the systematic nodes, while the other  $(n-k)$  nodes are termed parity nodes. For many applications such as degraded reads, efficient recovery of the data in a systematic node is of greater importance than that of a parity node. Keeping this in mind, we relax the setting described above to the following requirements:

- one should be able to recover the data stored in any node from any  $d$  other nodes with a download equal to (2)
- one should be able to recover the data stored in any systematic node, from any  $d$  other nodes, with the read and download equal to (3) and (2) respectively.

In other words, the requirement of repair-by-transfer is relaxed to hold only when recovering the data of a systematic node.

We now present an explicit code that achieves the conditions listed above. This code is a modification of the ‘product-matrix’ MBR code of [4].<sup>2</sup> The code is linear, and operates over any finite field  $\mathbb{F}_q$  of size  $q (\geq n)$ . As in [4], we present constructions for the case when  $\beta_D = 1$  symbol over  $\mathbb{F}_q$ ; codes for a general  $\beta_D$  can be obtained via multiple concatenations of this code (see [4, Section I-C]). When  $\beta_D = 1$  symbol over  $\mathbb{F}_q$ , (1) reduces to having  $\alpha = d$  symbols over  $\mathbb{F}_q$ .

We first present a brief overview of the construction of a product-matrix MBR code as in [4]. Denote this code as  $\mathcal{C}$ . The product-matrix MBR code is designed to satisfy (2), i.e., when  $\beta_D = 1$  symbol over  $\mathbb{F}_q$ , it operates on a data of size

$$B = kd - \binom{k}{2} \tag{5}$$

symbols over  $\mathbb{F}_q$ . Under the encoding mechanism of [4], this data is arranged as the entries of a  $(d \times d)$  *symmetric* matrix  $M$  of the form

$$M = \begin{bmatrix} S & R \\ R^T & 0 \end{bmatrix}$$

where  $S$  is a  $(k \times k)$  *symmetric* matrix and  $R$  is a  $(k \times (d-k))$  matrix.  $R^T$  denotes the transpose of  $R$ , and  $0$  is a  $((d-k) \times (d-k))$  zero matrix. Observe that the total number of independent entries in  $S$  is  $\frac{k(k+1)}{2}$  and that in  $R$  is  $k(d-k)$ , and these two quantities add up to (5).

Each node  $i \in \{1, \dots, n\}$  in the product-matrix MBR code is associated to a  $d$ -length vector  $\psi_i$ . The vectors  $\{\psi_i\}_{i=1}^n$  are chosen to satisfy two conditions: (a) any  $d$  of these  $n$  vectors are linearly independent, and (b) when restricted to the first  $k$  components, any  $k$  of these  $n$  vectors are linearly independent.

Every node  $i \in \{1, \dots, n\}$  stores the  $\alpha (= d)$  symbols

$$\psi_i^T M .$$

In this section, we shall assume that the code is systematic [4, Theorem 1], [4, Section IV-B] with nodes  $\{1, \dots, k\}$  being the systematic nodes.

It is shown in [4, Theorem 3], by means of an explicit decoding algorithm, that the entire data can be recovered from the data of any  $k$  of the  $n$  nodes. This exploits the property of linear independence of the first  $k$  components of  $\{\psi_i\}_{i=1}^n$ .

Let us now look at recovering the data stored in an individual node  $i \in \{1, \dots, n\}$  from some  $d$  nodes  $\{j_1, \dots, j_d\}$ . Under the protocol proposed in [4], each of these  $d$  nodes computes the inner product of the  $d$  symbols stored in it with the  $d$ -length vector  $\psi_i$ , and passes the result. Thus, the aggregate data obtained is  $\{\psi_{j_1}^T M \psi_i, \dots, \psi_{j_d}^T M \psi_i\}$ . The linear independence of the  $d$  vectors  $\{\psi_{j_1}, \dots, \psi_{j_d}\}$  and the symmetry of matrix  $M$  allows for recovery of the desired data

<sup>2</sup>While we discuss only the MBR case here, the ideas presented are also applicable to the product-matrix MSR codes of [4].

$\psi_i^T M$ . Observe that the amount of download is equal to  $d$  symbols over  $\mathbb{F}_q$ , and hence the code achieves (2).

We shall now modify the code  $\mathcal{C}$  described above to obtain a new code  $\mathcal{C}_1$  that, in addition, also minimizes the read during recovery of the data stored in any systematic node. Define a  $(d \times d)$  matrix

$$\Psi_0 = [\psi_1 \ \psi_2 \ \dots \ \psi_d]. \quad (6)$$

Under  $\mathcal{C}_1$ , each node  $i \in \{1, \dots, n\}$  stores the  $\alpha (= d)$  symbols

$$\psi_i^T M \Psi_0$$

(as opposed to storing  $\psi_i^T M$  under  $\mathcal{C}$ ).

Let us now verify that code  $\mathcal{C}_1$  meets all the requirements. First, observe that the  $(d \times d)$  matrix  $\Psi_0$  is invertible. Thus, the data stored in any node under  $\mathcal{C}_1$  is equivalent [4, Appendix B] to that stored under  $\mathcal{C}$ . This results in the fulfilment of the conditions of recovery of the entire data from any  $k$  nodes, and recovery of the data stored in any node from any  $d$  nodes with a download equalling (2).

Now consider recovering the data stored in any systematic node  $i \in \{1, \dots, k\}$  from any  $d$  nodes  $\{j_1, \dots, j_d\}$ . Under  $\mathcal{C}_1$ , every node  $\ell \in \{j_1, \dots, j_d\}$  simply reads and passes the  $i^{\text{th}}$  symbol it stores, which from (6), equals  $\psi_\ell^T M \psi_i$ . The data thus obtained is identical to that obtained under  $\mathcal{C}$ , thereby ensuring successful recovery. The amount of read and download is exactly  $d$ , thus meeting (3) and (2).<sup>3</sup>

### B. Optimal recovery from $d$ specific nodes

In certain applications, the flexibility of minimizing the read from any set of  $d$  nodes may be an overkill. This motivates the next relaxation, that mandates the following requirements:

- one should be able to recover the data stored in any node from any  $d$  other nodes with a download equal to (2)
- for any node, there must exist at least one set of  $d$  other nodes such that recovery from these  $d$  nodes entails a read and download equal to (3) and (2) respectively.

In other words, for recovery of the data stored in any node, the requirement of repair-by-transfer is relaxed to hold only for any one subset of  $d$  nodes.

We now modify the product-matrix MBR code  $\mathcal{C}$  described above to obtain a code  $\mathcal{C}_2$  that satisfies these conditions. To simplify notation, define an operator  $\oplus : \{1, \dots, n\} \times \{1, \dots, n\} \rightarrow \{1, \dots, n\}$  that computes a sum that cycles in the set  $\{1, \dots, n\}$ , i.e., for any  $x, y \in \{1, \dots, n\}$ ,  $x \oplus y := 1 + ((x - 1 + y) \bmod n)$ . Let  $\ominus$  be an analogous subtraction operator, with  $x \ominus y := 1 + ((x - 1 - y) \bmod n)$ . Under  $\mathcal{C}_2$ , each node  $i \in \{1, \dots, n\}$  stores the  $\alpha$  symbols

$$\psi_i^T M [\psi_{i \oplus 1} \ \psi_{i \oplus 2} \ \dots \ \psi_{i \oplus d}].$$

Let us now verify that code  $\mathcal{C}_2$  meets all the requirements. Since any  $d$  vectors from the set  $\{\psi_1, \dots, \psi_n\}$  are linearly independent, the matrix  $[\psi_{i \oplus 1} \ \psi_{i \oplus 2} \ \dots \ \psi_{i \oplus d}]$  is invertible for every  $i$ . Thus the data stored by a node under  $\mathcal{C}_2$  is equivalent [4, Appendix B] to that stored under  $\mathcal{C}$ . This results

<sup>3</sup>We note that this property, in fact, is applicable to the repair of any of the first  $d$  ( $\geq k$ ) nodes.

in the fulfilment of the properties of recovery of the entire data from any  $k$  nodes and recovery of data of any individual node from any  $d$  nodes with a minimum download.

Under  $\mathcal{C}_2$ , in order to recover the data stored in any node  $i$  with a read and download equal to (3) and (2), the  $d$  nodes  $(i \ominus d), \dots, (i \ominus 1)$  are queried. Each node  $\ell \in \{(i \ominus d), \dots, (i \ominus 1)\}$  simply reads and transfers the symbol  $\psi_\ell^T M \psi_i$  that it has stored. The data thus obtained is identical to that obtained under  $\mathcal{C}$ , allowing for successful recovery of the desired data. This meets the bounds (3) and (2) on the read and download.

## V. CONCLUSIONS

We consider the problem of constructing codes for distributed storage under which the data stored in any individual node can be efficiently recovered from the remaining nodes. In particular, we wish to achieve the previously derived [1] lower bounds on the amount of download and read. Achieving these bounds is equivalent to performing *repair-by-transfer* [2], [3] while meeting the bound on the download. Explicit codes with these properties were constructed previously [2], [3] for  $d = n - 1$ , and in this paper, we complete the picture by providing the converse to this result. In particular, we provide an information-theoretic impossibility result to show that the bound is not achievable when  $d \neq n - 1$  (even with non-linear codes). Obtaining tighter lower and upper bounds on the read under this setting are interesting directions for future research.

We also construct explicit codes for the two following (practical) relaxations, meeting the aforementioned bounds: (a) the read and download is simultaneously optimized for recovery of data of only systematic nodes, only the download is optimized for other nodes, and (b) for recovery of the data of any node, there is at least one set of  $d$  nodes from which the read and download are simultaneously optimized, only the download is optimized for recovery from any other set of  $d$  nodes. These codes are obtained by modifying the product-matrix codes of [4].

## REFERENCES

- [1] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4539–4551, 2010.
- [2] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit construction of optimal exact regenerating codes for distributed storage," in *Proc. Allerton Conf.*, Urbana-Champaign, Sep. 2009.
- [3] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Distributed storage codes with repair-by-transfer and non-achievability of interior points on the storage-bandwidth tradeoff," *IEEE Trans. Inf. Theory*, vol. 58, no. 3, pp. 1837–1852, Mar. 2012.
- [4] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Optimal exact-regenerating codes for the MSR and MBR points via a product-matrix construction," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5227–5239, Aug. 2011.
- [5] S. El Rouayheb and K. Ramchandran, "Fractional repetition codes for repair in distributed storage systems," in *Proc. Allerton Conf.*, Sep. 2010.
- [6] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Interference alignment in regenerating codes for distributed storage: Necessity and code constructions," *IEEE Trans. Inf. Th.*, vol. 58, no. 4, 2012.
- [7] V. Cadambe, C. Huang, and J. Li, "Permutation code: optimal exact-repair of a single failed node in MDS code based distributed storage systems," in *IEEE ISIT*, 2011.
- [8] I. Tamo, Z. Wang, and J. Bruck, "MDS array codes with optimal rebuilding," in *Proc. IEEE ISIT*, St. Petersburg, Jul. 2011.
- [9] K. Shum and Y. Hu, "Functional-repair-by-transfer regenerating codes," in *IEEE ISIT*, Jul. 2012.