# Distributed Storage Codes With Repair-by-Transfer and Nonachievability of Interior Points on the Storage-Bandwidth Tradeoff

Nihar B. Shah, K. V. Rashmi, P. Vijay Kumar, *Fellow, IEEE*, and Kannan Ramchandran, *Fellow, IEEE*

***Abstract*—Regenerating codes are a class of recently developed codes for distributed storage that, like Reed-Solomon codes, permit data recovery from any subset of $k$ nodes within the $n$-node network. However, regenerating codes possess in addition, the ability to repair a failed node by connecting to an arbitrary subset of $d$ nodes. It has been shown that for the case of functional repair, there is a tradeoff between the amount of data stored per node and the bandwidth required to repair a failed node. A special case of functional repair is exact repair where the replacement node is required to store data identical to that in the failed node. Exact repair is of interest as it greatly simplifies system implementation. The first result of this paper is an explicit, exact-repair code for the point on the storage-bandwidth tradeoff corresponding to the minimum possible repair bandwidth, for the case when $d = n-1$. This code has a particularly simple graphical description, and most interestingly has the ability to carry out exact repair without any need to perform arithmetic operations. We term this ability of the code to perform repair through mere transfer of data as *repair by transfer*. The second result of this paper shows that the interior points on the storage-bandwidth tradeoff cannot be achieved under exact repair, thus pointing to the existence of a separate tradeoff under exact repair. Specifically, we identify a set of scenarios which we term as "helper node pooling," and show that it is the necessity to satisfy such scenarios that overconstrains the system.**

***Index Terms*—Distributed storage, minimum bandwidth, node repair, regenerating codes, storage versus repair-bandwidth tradeoff.**

## I. Introduction

In A distributed storage system, the source data (*message*) is encoded and dispersed across nodes in a network in such a manner that an end user (termed as the data collector) can retrieve the data stored by tapping into a subset of nodes in the network. A popular option that reduces network congestion and leads to increased resiliency in the face of node failures is to employ erasure coding, for example, by calling upon maximum-distance-separable (MDS) codes such as Reed–Solomon (RS) codes.

Let the source data to be stored in the network be represented by a collection of $B$ message symbols, with each message symbol drawn from a finite field $\mathbb{F}_q$ of size $q$. An $[n, k]$ RS code encodes the message of size $k = B$ to obtain $n$ symbols over $\mathbb{F}_q$, and stores one distinct coded symbol in each of the $n$ nodes in the network. Under this encoding operation, the entire data can be recovered by a data collector by connecting to any arbitrary $k$ nodes, a process of data recovery that we will refer to as *reconstruction*. Several distributed storage systems such as RAID-6 [1], OceanStore [2], and Total Recall [3] employ such an erasure-coding option.

Upon failure of an individual node, a self-sustaining data storage network must necessarily possess the ability to *regenerate* (i.e., repair) the failed node. An obvious means of accomplishing this task is by first permitting the replacement node to download the entire data stored in any $k$ nodes and then proceeding to extract the data that was stored in the failed node. Such a procedure is indeed mandated when RS codes are employed to distribute the data and nodes are restricted to carry out linear operations on the data stored within them.

RS codes treat the data stored in each node as a single symbol belonging to the finite field $\mathbb{F}_q$. When this is coupled with the restriction that individual nodes perform linear operations over $\mathbb{F}_q$, it follows that the smallest unit of data that can be downloaded from a node assisting in the repair of a failed node equals the amount of information stored in the node itself (namely, the equivalent of an $\mathbb{F}_q$ symbol). As a consequence of the MDS property of an RS code, when carrying out repair of a failed node, the replacement node must necessarily collect data from at least $k$ other nodes. It follows that the total amount of data download needed to repair a failed node can be no smaller than $B$, the size of the entire message.

However, downloading the entire message of size $B$ in order to repair a single node that stores only a fraction $\frac{1}{k}$ of the entire data is wasteful, and raises the question as to whether there is a better alternative. Such an alternative is provided by the concept of a *regenerating code* introduced in the pioneering paper by Dimakis *et al.* [4].

Manuscript received November 16, 2010; revised June 13, 2011; accepted September 21, 2011. Date of publication October 27, 2011; date of current version February 29, 2012. The work of P. Vijay Kumar was supported in part by the National Science Foundation under Grant 0964507 and in part by a grant from the Indian Defence Research and Development Organization. The material in this paper was presented in part at the Allerton Conference on Communication, Control, and Computing, Monticello, IL, Sep. 2009, in part at the Information Theory and Applications Workshop, San Diego, CA, Feb. 2010, and in part at the 2010 IEEE International Symposium on Information Theory.

N. B. Shah, K. V. Rashmi, and K. Ramchandran are with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720 USA (e-mail: nihar@eecs.berkeley.edu; rashmikv@eecs.berkeley.edu; kannanr@eecs.berkeley.edu).

P. V. Kumar is with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India, and also with the Department of Electrical Engineering Systems, University of Southern California, Los Angeles, CA 90089-2565 USA (e-mail: vijay@ece.iisc.ernet.in).

Communicated by C. Fragouli, Associate Editor for Communication Networks.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2011.2173792

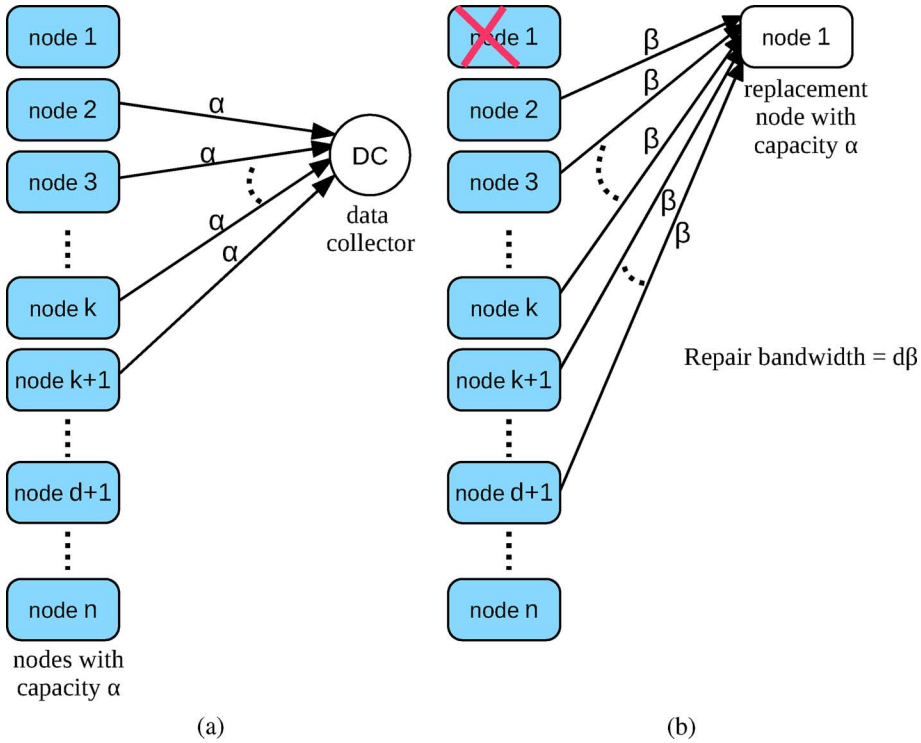


Fig. 1. Regenerating codes setup. (a) Data reconstruction. (b) Repair of a failed node.

### A. Regenerating Codes

In the regeneration framework introduced in [4], codes whose symbol alphabet is a vector over $\mathbb{F}_q$, i.e., an element of $\mathbb{F}_q^\alpha$ for some integer parameter $\alpha > 1$ are employed. As with RS codes, each node still continues to store a single code symbol. However, given the vector nature of the code-symbol alphabet, we may equivalently regard, each node as storing a collection of $\alpha$ symbols, each symbol drawn from $\mathbb{F}_q$. Under this setup, it is clear that while maintaining linearity over $\mathbb{F}_q$, it is possible for an individual node to transfer a fraction of the data stored within the node.

Apart from this new parameter $\alpha$, two other parameters $d$ and $\beta$ are associated with the regenerating framework introduced in [4]. A replacement node is permitted to connect to an arbitrary subset of $d$ nodes out of the remaining $(n-1)$ nodes while downloading $\beta \leq \alpha$ symbols from each node. The $d$ nodes helping in the repair of a failed node are termed as *helper nodes*. The total amount $d\beta$ of data downloaded for repair purposes is termed the *repair bandwidth*. Thus, under this framework, apart from the field size $q$, we have

$$\{\,[n,\ k,\ d],\ (\beta,\ \alpha,\ B)\,\}$$

as the parameter set. The corresponding codes are called *regenerating codes*. Typically, with a regenerating code, the average repair bandwidth $d\beta$ is small compared to the size of the file $B$. Fig. 1(a) and (b) illustrates reconstruction and node repair, respectively, while also depicting the relevant parameters.

An important feature of regenerating codes is that they allow each storage node to store more than the minimum required in order to reduce the repair bandwidth. While an MDS code would require the message size $B = k\alpha$, we shall soon see that regenerating codes permit the system to store $B \leq k\alpha$, which reduces the repair bandwidth.

Note that the parameters $k$ and $d$ are the minimum values under which reconstruction and repair can always be guaranteed. This restricts the range of $d$ to

$$k \leq d \leq n-1 \tag{1}$$

for, if the repair parameter $d$ were less than the reconstruction parameter $k$, this would imply that one could in fact reconstruct the data by connecting to $d$ nodes, thereby contradicting the minimality of $k$.

### B. Exact Versus Functional Repair

Under the notion of *functional repair* introduced in [4], a failed node is replaced by a node that is functionally equivalent, i.e., following replacement, the resulting network of $n$ nodes must continue to possess the reconstruction and repair properties. With this being the sole constraint, the data (and the code) stored at the replacement node may be (arbitrarily) different from that stored in the corresponding failed node. This change in code coefficients at the replacement node may necessitate additional communication to the remaining entities in the network informing them of the change. Moreover, it may also require changes in the decoding algorithms at the data collectors and replacement nodes to accommodate the modified code coefficients.

In contrast, under *exact repair*, introduced subsequently in [5], [6], a replacement node is required to store exactly the same data as was stored in the failed node. Hence, there is no change in the coefficients of a replaced node under exact repair. This

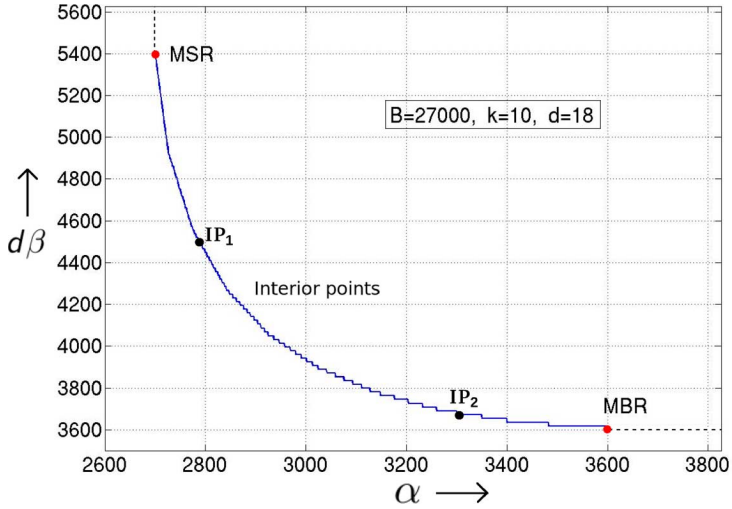


Fig. 2. Storage-bandwidth tradeoff curve. Storage space $\alpha$ versus repair bandwidth $d\beta$. Also depicted are the two end points MBR and MSR, and two interior points $\mathrm{IP}_1$ and $\mathrm{IP}_2$.

obviates additional communication overheads during the repair operation, and also avoids retuning of the reconstruction and repair algorithms.

An additional advantage of exact repair over its functional counterpart is the ability to maintain the code in systematic form. In a systematic code, there exist a set of $k$ nodes (which we shall refer to as "systematic nodes" in the sequel) which together contain the entire message in an uncoded form. The ability to store data in systematic form is practically useful since a data collector connecting to the $k$ systematic nodes can obtain the entire message without having to perform any decoding operations. Under functional repair, it may not be possible to maintain a code in systematic form since the replacement of a failed systematic node may contain data not in the systematic form. On the other hand, under exact repair, the replacement of a failed systematic node will continue to be systematic.

Thus, exact repair greatly simplifies system implementation and is of considerable practical interest. We use the term *exact-repair code* to denote a regenerating code that is capable of performing exact repair of any failed node.

### C. Storage-Repair Bandwidth Tradeoff

A major result in the field of regenerating codes is the proof in [7] that uses the cut-set bound of network coding to establish that the parameters of a regenerating code must necessarily satisfy[1]

$$B \leq \sum_{i=0}^{k-1} \min\{\alpha,\ (d-i)\beta\} \tag{2}$$

Since both storage and bandwidth come at a cost, it is naturally desirable to minimize both $\alpha$ as well as $\beta$. However, it can be deduced (see [7]) that achieving equality in (2), for fixed values of $B$ and $[n,\ k,\ d]$ leads to a tradeoff between the storage space $\alpha$ and the repair bandwidth $d\beta$. This tradeoff is termed as the storage-repair bandwidth tradeoff, or more simply as the storage-bandwidth tradeoff. In Fig. 2, the tradeoff is plotted for $B = 27000$ symbols for a system with $k = 10,\ d = 18$, and some $n > 18$.

For fixed values of $B$ and $[n,\ k,\ d]$, a regenerating code is said to be *optimal*, if the parameters $(\beta,\ \alpha,\ B)$ are such that

1) equality holds in (2), i.e.,

$$B = \sum_{i=0}^{k-1} \min\{\alpha,\ (d-i)\beta\} \tag{3}$$

and

2) if either $\alpha$ or $\beta$ is decreased, (3) fails to hold.

The parameters $(\beta,\ \alpha,\ B)$ of any optimal regenerating code are said to lie on the storage-bandwidth tradeoff.

Observe that when $\alpha < (d-(k-1))\beta$, the parameter $\beta$ can be decreased without violating (3). Hence, the parameters of an optimal regenerating code must necessarily satisfy

$$\alpha \geq (d-k+1)\beta \tag{4}$$

The case when $\alpha$ takes the minimum value

$$\alpha = (d-k+1)\beta, \tag{5}$$

is one of the extreme points of the tradeoff called the minimum storage regenerating (MSR) point. From (3) and (5), we see that the parameters at the MSR point satisfy

$$\left(\alpha = \frac{B}{k},\ \beta = \frac{B}{k(d-k+1)}\right) \tag{6}$$

On the other hand, if $\beta < \frac{\alpha}{d}$ (i.e., if $\alpha > d\beta$), the parameter $\alpha$ can be decreased without violating (3). Hence, the parameters of an optimal regenerating code must necessarily satisfy

$$\beta \geq \frac{\alpha}{d} \tag{7}$$

The case when $\beta$ takes this minimum value, i.e.

$$\alpha = d\beta \tag{8}$$

[1]This bound on the message size $B$ is originally derived in [7] using the principles of network coding. An information-theoretic derivation is presented in Section IV-B in this paper.

is the other extreme point of the tradeoff called the minimum bandwidth regenerating (MBR) point. From (3) and (8) we see that the parameters at the MBR point satisfy

$$\left(\alpha = \frac{2dB}{k(2d-k+1)},\ \beta = \frac{2B}{k(2d-k+1)}\right) \tag{9}$$

It can be inferred from (4) and (7) that an optimal regenerating code must have the parameter $\alpha$ lying in the range

$$(d-k+1)\beta \leq \alpha \leq d\beta \tag{10}$$

Points on the tradeoff other than the two extreme (MSR and MBR) points have the parameter $\alpha$ lying strictly within this range: $(d-k+1)\beta < \alpha < d\beta$. These points are, hence, referred to as the *interior points* on the tradeoff curve.

Note that the scenario when $k = 1$ results in $B = \alpha = d\beta$ which can be satisfied trivially by a repetition code. Thus, we assume $k > 1$ throughout this paper.

The storage-bandwidth tradeoff was derived in [7] for the case of functional repair and was shown to be tight in [7] and [8]. Clearly, the bound continues to hold even under the exact-repair setting, since exact repair is an instance of functional repair. However, the achievability of this bound under an exact-repair requirement has remained an open problem, and will be addressed in this paper.

### D. Summary of the Results in This Paper

The first main result of this paper is an explicit construction of exact MBR codes for the case $d = n-1$ that has a simple graphical description. An interesting and potentially useful aspect of this construction is its *repair-by-transfer* property where a failed node is repaired by simple transfer of data without the need for any computation at either the helper nodes or the replacement node. The property of repair-by-transfer leads to several advantages which gives the code practical appeal: 1) reduced complexity of repair; 2) minimum disk reads at helper nodes; 3) no intelligence required at storage disks; and 4) efficient handling of situations where a user requires access to only the data stored in a single node, even while that node is under repair. These properties will be revisited during the description of the code construction in Section III. An additional advantage, when specialized to the parameter set $[n,\ k = n-2,\ d = n-1]$, is that all operations in the system can be accomplished using XOR operations alone.[2]

The second main result of this paper answers an open problem regarding the achievability of the storage-bandwidth tradeoff under exact repair at the interior points. First, a set of interesting properties required to be satisfied by an exact-repair code are derived, which may also be of independent interest. Subsequently, the nonachievability of the interior points on the storage-bandwidth tradeoff under exact repair is established, with the possible exception of points within the immediate vicinity of the MSR point.

### E. Organization

This paper is organized as follows. A brief overview of the related literature is provided in Section II. Section III contains the exact MBR code construction. A set of properties that any exact-repair code must necessarily satisfy are provided in Section IV, which are then used to establish the nonachievability of the storage-bandwidth tradeoff for exact repair at essentially all interior points. Section V presents conclusions.

[2]An animated video of an example of this code is available in [9].

## II. Related Work

The concept of regenerating codes, introduced in [4] and [7], permits storage nodes to store more than the minimal $B/k$ units of data in order to reduce the repair bandwidth. Several distributed systems are analyzed, and estimates of the mean node availability in such systems are obtained. Using these values, the substantial performance gains offered by regenerating codes in terms of bandwidth savings are demonstrated. The problem of minimizing repair bandwidth for *functional* repair of nodes is formulated as a multicast network-coding problem in a network having an infinite number of nodes. A cut-set lower bound on the repair bandwidth is derived. Coding schemes achieving this bound are presented in [7] and [8] which, however, are nonexplicit. These schemes require large field size and the repair and reconstruction algorithms are also of high complexity.

The notion of exact repair is introduced independently in [5] and [6]. In [5], the MSR point is shown to be achievable under exact repair for the parameters $[n,\ k = 2,\ d = n-1]$. The proposed coding scheme uses the concept of interference alignment. Even here, the constructions are not explicit, and have large complexity and field-size requirement.

The first explicit construction of regenerating codes appears in [6]. An explicit MSR code is constructed here, for $[n,\ k,\ d = k+1]$ (see also the journal version [10]). A computer search for exact-repair MSR codes for the parameter set $[n = 5,\ k = 3,\ d = 4]$ is carried out in [11], and for this set of parameters, codes for several values of field size are obtained.

A slightly different setting from the exact-repair situation is considered in [12], where optimal MDS codes are given for the parameters $[n > 2k,\ k,\ d = k+1]$. Again, the schemes given here are nonexplicit, and have high complexity and large field-size requirement.

An explicit code structure at the MSR point that guarantees reconstruction and exact repair of the systematic nodes is provided in [14], for parameters $[n \geq 2k,\ k,\ d = n-1]$. This code makes use of interference alignment, and is termed as the "MISER" code in the journal-submission version [10] of [14]. Following the initial submission of [14], it is shown in [15] that the code introduced in [14] can perform exact repair of parity nodes as well, and explicit mechanisms for the same are also provided. The impossibility of constructing linear, scalar (i.e., $\beta = 1$) exact MSR codes when $d < 2k-3$ is shown in [10] and [14]. On the other hand, in the limiting case of $B$ (and hence $\alpha$ and $\beta$) approaching infinity, the MSR point is shown to be achievable under exact repair for all $[n,\ k,\ d]$ in [16] and [17].

A general framework, termed the *product-matrix* framework, that enables code construction for a wide range of parameters is introduced in [13]. Explicit codes at the MBR point for all values of the parameters $[n,\ k,\ d]$ and at the MSR point for the parameter set $[n,\ k,\ d \geq 2k-2]$ are constructed in this framework. Also contained in this paper is a simpler description of the MISER code in the product-matrix framework.

There has also been some work in the literature that considers slightly different models for distributed storage systems. Codes for more relaxed settings with respect to the reconstruction/repair requirements are presented in [18] and [19]. The papers [20]–[23] provide alternative frameworks for regenerating codes that introduce additional parameters for the system; tradeoffs between storage and repair are derived in each of the papers for the functional-repair scenario. A method to construct distributed storage systems that use existing erasure codes and also enjoy the benefits of efficient node repair is presented in [24].

The preliminary version of the nonachievability results provided in this paper appeared in [25] and used subspace arguments to show nonachievability of the interior points with linear codes. This paper employs (stronger) information-theoretic arguments to show the nonachievability with any (not necessarily linear) code. In [25], we also present exact MBR codes for arbitrary values of $[n,\ k,\ d]$ (that, however, lack the repair-by-transfer property). The codes in [25] are subsumed by the product-matrix codes in [13]. A description of how the MBR codes in this paper relate to the product-matrix exact MBR codes in [13] is provided in Section III-F of this paper.

Following the initial presentation of the exact MBR codes performing repair by transfer in [6] (described in Section III in this paper), El Rouayheb and Ramchandran [26] use the graph-based approach for code construction presented here to extend these codes to a larger set of parameters.[3] They, however, consider a somewhat relaxed setting wherein a replacement node can connect to only certain fixed subsets of $d$ nodes for repair; they also provide upper bounds on the storage capacity of such systems. In [27], the authors present a distributed file system on which they implement and analyze the exact MBR codes described in this paper. In [28] and [29], authors show that the exact MBR codes described in this paper can be used to provide information-theoretic security in the presence of eavesdroppers and adversarial node attacks.

## III. Explicit Exact MBR Code for $d = n - 1$ With Repair by Transfer

In this section, we provide an explicit construction of exact MBR codes wherein the parameter $d$ takes the largest permissible value of $n - 1$.[4] These codes are capable of performing exact repair of any failed node in such a way that the repair process is accomplished with mere transfer of data and without need for any arithmetic operations either at the helper nodes or at the replacement node. This property makes the code practically appealing.

First, we present a brief overview of the MBR point on the storage-bandwidth tradeoff and the concept of *striping* of data which will be used in the code construction.

[3]In this paper, the authors refer to the repair-by-transfer property of codes in [6] as *uncoded repair*.

[4]It can be inferred from the storage-bandwidth tradeoff (3) that for fixed values of the parameters $n,\ k,\ \alpha$ and $B$, the repair bandwidth $d\beta$ decreases with increase in $d$.

### A. MBR Point Parameters

The MBR point is an extreme point on the storage-bandwidth tradeoff that corresponds to the least possible repair bandwidth. As previously discussed in Section I-C, the parameters $\alpha$ and $\beta$ for the MBR point satisfy

$$\begin{aligned} \alpha &= d\beta \\ B &= \left(kd - \frac{k(k-1)}{2}\right)\beta \end{aligned} \tag{11}$$

Thus, at the MBR point, a replacement node downloads exactly the number of symbols it eventually stores.

At this point, we briefly digress to analyze a particular relation between the parameters $(\beta,\ \alpha,\ B)$ of a regenerating code that will aid in code construction as well as in simplifying the system implementation of the code.

### B. Striping of Data

Given a set of parameters $(\beta,\ \alpha,\ B)$ of an optimal regenerating code [i.e., satisfying (3)], the parameters $(\beta' = \delta\beta,\ \alpha' = \delta\alpha,\ B' = \delta B)$ for any positive integer $\delta$ also satisfy (3). Thus, for some $[n,\ k,\ d]$, an optimal regenerating code for $(\beta',\ \alpha',\ B')$ can be obtained easily by dividing the $B' = \delta B$ message symbols into $\delta$ groups of $B$ symbols each, and applying the optimal $(\beta,\ \alpha,\ B)$ code to each group independently. In particular, if one can construct an $[n,\ k,\ d]$ MBR code with $\beta = 1$, then one can construct an $[n,\ k,\ d]$ MBR code for any larger integer value of $\beta$ as well. Moreover, from a practical standpoint, a code with smaller $\beta$ will involve manipulating a smaller number of message symbols and, hence, may lead to algorithms of lesser complexity. For these reasons, in this paper we design codes for the case of $\beta = 1$. In this scenario, the values of $\alpha$ and $B$ at the MBR point are given by

$$\begin{aligned} \alpha &= d \\ B &= kd - \binom{k}{2} \end{aligned} \tag{12}$$

Next, we present an example construction, before moving on to the general case.

### C. Example Code

The example deals with the parameter set $[n = 5,\ k = 3,\ d = 4]$, $\beta = 1$ which from (12) gives $\alpha = 4$ and $B = 9$. Let the nine message symbols be denoted by $\{m_i | 1 \le i \le 9\}$.

*Encoding:* Our code construction can be visualized in terms of a fully connected graph on five nodes, each representing a distinct node in the network (see Fig. 3). We encode the nine message symbols using a $[10,\ 9]$ MDS code $\mathcal{C}$, and to each of the ten edges in the graph, we assign a distinct symbol from this set $\{c_i | 1 \le i \le 10\}$ of code symbols. The code $\mathcal{C}$ can be chosen as any $[10,\ 9]$ MDS code, for example, $\mathcal{C}$ could be a single parity check code of length 10. Under our construction, each storage node stores the four symbols assigned to the four edges incident on the node, as shown in Fig. 3.

*Data Reconstruction:* Suppose a data collector connects to nodes 2, 3, and 4. The data collector, then, recovers $3 \times 4 = 12$ coded symbols of which $12 - \binom{3}{2} = 9$ are distinct. Since the code $\mathcal{C}$ is an $[10,\ 9]$-MDS code by construction, the data collector

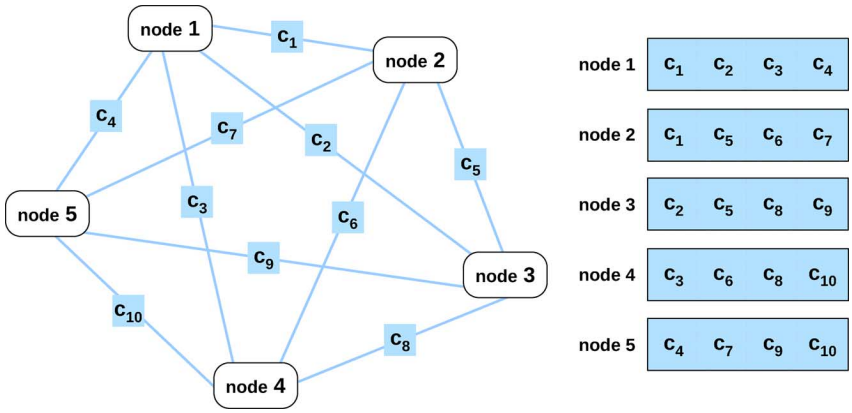


Fig. 3. Graphical representation of the repair-by-transfer code for the MBR point with $[n = 5,\ k = 3,\ d = 4]$.

can recover the nine message symbols from these nine coded symbols.

*Repair by Transfer:* Suppose node 3 fails, and is replaced by an empty replacement node. Under our construction, each of the four remaining nodes pass the symbol assigned to the edge that it has in common with node 3, i.e., nodes 1, 2, 4, and 5 pass on the symbols $c_2$, $c_5$, $c_8$, and $c_9$, respectively, to the replacement node (from Fig. 3). But these are precisely the four symbols that were stored in node 3 prior to failure, and hence, node 3 is exactly repaired. Note that the repair is accomplished by mere transfer of data and no arithmetic operations are required either at the helper nodes or at the replacement node.

This property, which we term as *repair by transfer*, has considerable practical appeal. The ability of a code to perform repair by transfer, while obviating the need to perform any computations during repair, permits the use of "dumb" storage disks (without extra intelligence). Such a code also minimizes the number of disk reads required at the helper nodes during repair. Moreover, this property enables an additional feature wherein a user who requires access to only the data stored in a single node can do so even while that node is under repair. This is due to the specific code structure wherein a copy of the data stored in a node is also stored across the other nodes in a distributed manner.

### D. Code Construction for the General Set of Parameters $[n,\ k,\ d = n - 1]$

As discussed previously, the code is constructed for the case of $\beta = 1$, and codes for any higher value of $\beta$ can be obtained via concatenation. The construction follows along the lines similar to the example provided in the previous section.

Let the $B$ message symbols be denoted by $\{m_i | 1 \leq i \leq B\}$.

*Encoding:* Let $\mathcal{C}$ be an $[\binom{n}{2},\ B]$-MDS code. Encode the $B$ message symbols using the code $\mathcal{C}$, and let the $\binom{n}{2}$ coded symbols be denoted by $\{c_i | 1 \leq i \leq \binom{n}{2}\}$. As in the example, we can visualize the code construction via a fully connected graph on $n$ nodes, each representing a distinct node in the network. To each of the $\binom{n}{2}$ edges in the graph, we assign a distinct code symbol from the set $\{c_i | 1 \leq i \leq \binom{n}{2}\}$ (in any arbitrary order).

Under our construction, each storage node stores the $\alpha$ $(= n - 1)$ symbols assigned to the $(n - 1)$ edges incident on the node. Thus, each symbol in $\{c_i | 1 \leq i \leq \binom{n}{2}\}$ is stored in precisely two nodes.

The following theorems establish the properties of data reconstruction and repair by transfer.

*Theorem 1 (Data Reconstruction):* A data collector can recover all the $B$ message symbols by connecting to any subset of $k$ nodes.

*Proof:* The data collector connects to a subset of $k$ nodes in the network, and recovers the $k\alpha$ $(= kd)$ code symbols stored in these nodes. Since every pair of nodes has exactly one symbol in common, there are $\binom{k}{2}$ redundant symbols among these $kd$ code symbols. Thus, the data collector has access to $kd - \binom{k}{2} = B$ distinct code symbols from the set $\{c_i | 1 \leq i \leq \binom{n}{2}\}$. Since the code $\mathcal{C}$ is an $[\binom{n}{2},\ B]$-MDS code by construction, the data collector can recover the $B$ message symbols $\{m_i | 1 \leq i \leq B\}$ from these $B$ code symbols. ■

Note that in our construction, the data reconstruction (decoding) procedure is identical to that of decoding an MDS code over an erasure channel.

*Theorem 2 (Repair by Transfer):* Exact repair of any failed node can be achieved by connecting to the remaining $(n - 1)$ nodes and downloading the minimum possible data. Furthermore, the process involves mere transfer of data and does not require any arithmetic operations.

*Proof:* On failure of a storage node, the replacement node connects to the $(n-1)$ remaining nodes. Each of the remaining nodes passes to the replacement node, the symbol assigned to the edge that it has in common with the failed node. By construction, these symbols are precisely the $(n-1) = \alpha$ symbols that were stored in the node prior to failure. Thus, the replacement node simply stores these symbols, completing the process of exact repair. Clearly, the repair process does not require any arithmetic operation either at the helper nodes or at the replacement node. ■

### E. Size of the Finite Field

The sole constraint on the field size required in the code construction arises from the need for the existence of an $[\binom{n}{2},\ B]$ MDS code. The existence of doubly extended RS codes tells us that a field size of $(\binom{n}{2} - 1)$ will suffice.

*Remark 1:* The example employed a single parity check code as its MDS code. This has practical appeal since all operations can be carried out in binary field using only XORs. We note

that a single parity check code will suffice whenever $k = n-2$, which from (12) gives $\binom{n}{2} = B + 1$.

### F. Relation to the Product-Matrix MBR Codes in [13]

Subsequent to the initial presentation of the repair-by-transfer codes in [6], a new product-matrix framework was introduced in [13]. This framework was used in [13] to construct explicit MSR codes for all $[n,\ k,\ d \geq 2k-2]$ and explicit MBR codes for all values of the parameters $[n,\ k,\ d]$. While the codes presented in [13] do not possess the repair-by-transfer property, in hindsight, one can provide an alternative construction to repair-by-transfer codes using the product-matrix MBR code for $d = n-1$.

In the product-matrix MBR code in [13] with $d = n-1$, let $\Psi$ denote the $(n \times d)$ encoding matrix and let $M$ denote the $(d \times d)$ message matrix. The message matrix $M$ contains as its elements, the $B$ message symbols arranged in a particular redundant fashion. The encoding matrix $\Psi$ is chosen such that any $d$ of its rows are linearly independent. The $\alpha$ symbols stored in node $i$ $(1 \leq i \leq n)$ are given by $\underline{\psi}_i^t M$, where $\underline{\psi}_i^t$ denotes the $i$th row of $\Psi$. For the repair of a failed node $f$, each remaining node passes an inner product of the $\alpha$ symbols stored in it with the encoding vector $\underline{\psi}_f^t$ of node $f$, i.e., the helper node $i$ passes the symbol $\underline{\psi}_i^t M \underline{\psi}_f$.

Taking a cue from the repair-by-transfer code constructed in this paper, we consider an equivalent code where node $i$ stores the set of $\alpha$ $(= n-1)$ symbols $\{\underline{\psi}_i^t M \underline{\psi}_j \mid 1 \leq j \leq n,\ j \neq i\}$.[5] Clearly, this code can perform repair by transfer. Furthermore, since the transformation of the $\alpha$ symbols stored in each node is nonsingular, the code retains the data-reconstruction property.

## IV. Nonexistence of Exact-Repair Codes Achieving the Interior Points on the Storage-Bandwidth Tradeoff

We now move on to the second main result of this paper which proves the nonachievability of the interior points on the storage-bandwidth tradeoff under exact repair. Originally, the storage-bandwidth tradeoff was derived for the case of functional repair. However, given the clear advantages of exact repair, much of the work in the field of regenerating codes has been dedicated to exact repair. In particular, the two extreme points of the tradeoff (MBR and MSR) have been much studied, and the achievability of the tradeoff for exact repair at the extreme points has been characterized to a large extent [10], [13]–[17] (see Section II for more details). On the other hand, the tightness of the storage-bandwidth tradeoff under exact repair has remained open for the interior points. In this section, we address this issue by proving the impossibility of constructing codes, performing exact repair at essentially all interior points on the storage-bandwidth tradeoff.

We begin by providing an information-theoretic perspective on regenerating codes. We then reparameterize the storage-bandwidth tradeoff in terms of two new parameters $p$ and $\theta$, which makes the proofs easier to understand. Following this, we derive a set of interesting properties that the amount of information stored and passed by the nodes must necessarily satisfy.

In our proof, we exploit the requirement that a regenerating code must be able to repair a failed node by connecting to *any* set of $d$ nodes. Showing that, under some scenario, there exists at least one set of $d$ nodes that are incapable of supporting repair proves the impossibility result. We identify such a scenario of repair wherein a common pool of nodes help in the repair of multiple nodes, and show that under this scenario, all the properties cannot be satisfied simultaneously. This establishes the impossibility of constructing exact repair codes operating at the interior points on the tradeoff.

### A. Notation

While in earlier sections we worked with individual symbols from a certain alphabet, in keeping with the information-theoretic approach of this section, we treat the message symbols as well as the data stored and passed by the nodes as random variables.

Under this information-theoretic perspective, the nodes in the network store data pertaining to a source (message) $M$, whose entropy is $B$, i.e.

$$H(M) = B \tag{13}$$

Next, we introduce the random variables, pertaining to the data stored in the nodes and the data passed by nodes for data recovery and repair purposes.

Let $W_\ell$ denote the random variable corresponding to the data stored in node $\ell$ $(1 \leq \ell \leq n)$. We will assume that each storage node has a storage capacity of $\alpha$ and is, hence, incapable of storing variables whose entropy are greater than $\alpha$, thus

$$H(W_\ell) \leq \alpha \tag{14}$$

Consider exact repair of node $\ell$ using a set $\mathcal{D}$ of $d$ helper nodes, and let node $m \in \mathcal{D}$. In this scenario, we denote the random variable corresponding to the data passed by the helper node $m$ to aid in the repair of node $\ell$ by ${}_{\mathcal{D}}S_m^\ell$. We also assume that the data links used for repair have capacity $\beta$ and, hence, are incapable of carrying variables whose entropy are greater than $\beta$, i.e.

$$H({}_{\mathcal{D}}S_m^\ell) \leq \beta \tag{15}$$

Both (14) and (15) are in keeping with the original setting where each node had the capacity to store $\alpha$ symbols and each data link used in repair had the capacity to carry $\beta$ symbols.

Note that since repair is exact, the random variables $W_\ell$ and ${}_{\mathcal{D}}S_m^\ell$ are invariant with time, i.e., they remain constant irrespective of the sequence of failures and repairs that occur in the system.[6]

The reconstruction property requires that the message $M$ be determined completely from the data stored in any $k$ nodes, and the exact-repair property requires that the data stored in the failed node be determined completely from the data passed by

[5]Two (linear) regenerating codes are defined to be equivalent if one can be obtained from the other by 1) a nonsingular transformation of the $B$ message symbols, and 2) a nonsingular transformation of the $\alpha$ symbols stored at each storage node. The reader is referred to [10] for a formal definition of equivalence of regenerating codes.

[6]In contrast, functional repair permits a replacement node to store data different from that stored in the failed node, thus leaving open the possibility that these variables are dependent on time.

the $d$ helper nodes. These requirements of reconstruction and exact repair can be stated information theoretically as follows.

1) From the reconstruction property required of a regenerating code, it must be that, for every subset of $k$ storage nodes: $\{\ell_i \mid 1 \leq i \leq k\}$, we need

$$H\left(M \left| \{W_{\ell_i}\}_{i=1}^{k} \right.\right) = 0 \tag{16}$$

2) Similarly, the exact-repair property requirement leads to the condition that for every node $\ell$ $(1 \leq \ell \leq n)$

$$H\left(W_\ell \left| \left\{{}_{\mathcal{D}}S_m^\ell\right\}_{m\in\mathcal{D}} \right.\right) = 0, \tag{17}$$

where $\mathcal{D}$ represents the set of $d$ helper nodes participating in the exact repair of node $\ell$.

In the sequel, for simplicity, we will drop the left subscript $\mathcal{D}$ and the set of $d$ nodes participating in the repair process will be clear from the context. Furthermore, since a node can pass only a function of what is stored in the node, it follows that for any node $m$

$$H\left(S_m^\ell \middle| W_m\right) = 0 \tag{18}$$

*Remark 2:* Throughout this section, we will assume that all the random variables are functions of the message $M$. This is without loss of generality since one can always assume a genie that reveals all the extraneous sources of randomness to every entity in the system, and this would still retain the necessity of the properties proved here. However, for convenience, we do not indicate this dependence in the notation. Thus, we have

$$H(W_\ell|M) = 0 \text{ and } H(S_m^\ell|M) = 0 \tag{19}$$

Now, from (13) and (19), one can rewrite the reconstruction property in (16) as

$$H\left(\{W_{\ell_i}\}_{i=1}^{k}\right) = B \tag{20}$$

Next, we set up notation to denote certain sets of random variables which will be used frequently. Let $\mathcal{A}$ denote a collection of storage nodes. Then, the set of random variables corresponding to the data stored in the nodes in $\mathcal{A}$ is denoted by

$$W_{\mathcal{A}} \triangleq \{W_i\}_{i\in\mathcal{A}} \tag{21}$$

Further, define $[m]$ as the set of numbers $\{1,\ldots,m\}$ for some positive integer $m$, and denote

$$W_{[m]} \triangleq \{W_i\}_{i=1}^{m} \tag{22}$$

Note that the notation $[0]$ will correspond to an empty set.

The random variables corresponding to the data passed for repair may be grouped in two ways. Denote the collection of random variables passed by nodes in set $\mathcal{A}$ to assist in the repair of a particular failed node $\ell$ by

$$S_{\mathcal{A}}^\ell \triangleq \left\{S_m^\ell\right\}_{m\in\mathcal{A}} \tag{23}$$

On the other hand, the collection of random variables passed by node $m$ to assist in the repairs of nodes in the set $\mathcal{A}$ is denoted by

$$S_m^{\mathcal{A}} \triangleq \left\{S_m^\ell\right\}_{\ell\in\mathcal{A}} \tag{24}$$

Note that in both the aforementioned cases, the other helper nodes participating in the repair process will be clear from the context.

### B. Information-Theoretic Derivation of the Tradeoff

We now present an information-theoretic derivation of the storage-bandwidth tradeoff (2), since it is convenient to remain in the information-theoretic domain throughout this section. The results established in this section are derived only for the case of exact repair. The extensions to the case of functional repair are straightforward and are explained subsequently.

The following lemma establishes a relation between the information stored in various nodes.

*Lemma 3:* For an arbitrary node $\ell$, an arbitrary subset $\mathcal{A}$ consisting of $a$ $(0 \leq a \leq d)$ nodes such that $\ell \notin \mathcal{A}$, and a scenario wherein the set of $d$ nodes helping in the repair of failed node $\ell$ includes the $a$ nodes in $\mathcal{A}$, it must be that

$$H\left(W_\ell \left| S_{\mathcal{A}}^\ell \right.\right) \leq \min(\alpha,\ (d-a)\beta) \tag{25}$$

Hence

$$H\left(W_\ell | W_{\mathcal{A}}\right) \leq \min(\alpha,\ (d-a)\beta) \tag{26}$$

*Proof:* Consider exact repair of node $\ell$ by connecting to the $a$ nodes in set $\mathcal{A}$ and $(d-a)$ other arbitrary nodes. Denote the set of these $(d-a)$ helper nodes by $\mathcal{B}$. Then, exact repair of node $\ell$ requires (recall (17))

$$\begin{aligned}
0 &= H\left(W_\ell \left| S_{\mathcal{A}}^\ell, S_{\mathcal{B}}^\ell \right.\right) && (27)\\
&= H\left(W_\ell \left| S_{\mathcal{A}}^\ell \right.\right) - I\left(W_\ell; S_{\mathcal{B}}^\ell \left| S_{\mathcal{A}}^\ell \right.\right) && (28)\\
&\geq H\left(W_\ell \left| S_{\mathcal{A}}^\ell \right.\right) - H\left(S_{\mathcal{B}}^\ell | S_{\mathcal{A}}^\ell\right) && (29)\\
&\geq H\left(W_\ell \left| S_{\mathcal{A}}^\ell \right.\right) - H\left(S_{\mathcal{B}}^\ell\right) && (30)\\
&\geq H\left(W_\ell \left| S_{\mathcal{A}}^\ell \right.\right) - (d-a)\beta && (31)\\
&\geq H\left(W_\ell | W_{\mathcal{A}}\right) - (d-a)\beta && (32)
\end{aligned}$$

where (31) follows since each of the $(d-a)$ helper nodes in $\mathcal{B}$ can pass at most $\beta$ units of information to the replacement node, and (32) is a result of the fact that a node can only pass a function of what it stores, i.e., $H(S_m^\ell|W_m) = 0$ for all nodes $m \in \mathcal{A}$. The inequality in (32), coupled with the constraint on the storage capacity of the nodes (i.e., $H(W_\ell) \leq \alpha$), leads to the desired result. ■

*Remark 3 (The Case of Functional Repair):* In the case of functional repair, the data stored in a node after repair need not be identical to that stored in it prior to failure. This makes the corresponding random variable a function of time. In this scenario, the aforementioned lemma applies when $W_\ell$ is the random variable corresponding to the data stored in node $\ell$ after being repaired with the help of $a$ nodes in $\mathcal{A}$ and $(d-a)$ other arbitrary nodes.

The next theorem gives a simple derivation of the storage-bandwidth tradeoff (2) for exact repair from an information-theoretic perspective.

*Theorem 4:* Any $[n,\ k,\ d]$, $(\beta,\ \alpha,\ B)$ regenerating code must necessarily satisfy

$$B \leq \sum_{\ell=0}^{k-1} \min\{\alpha,\ (d-\ell)\beta\} \tag{33}$$

*Proof:* The reconstruction property [recall (20)] requires

$$B = H\left(W_{[k]}\right) \tag{34}$$

$$= \sum_{\ell=0}^{k-1} H\left(W_{\ell+1} \left| W_{[\ell]}\right.\right) \tag{35}$$

$$\leq \sum_{\ell=0}^{k-1} \min\{\alpha,\ (d-\ell)\beta\} \tag{36}$$

where (36) follows from Lemma 3. ■

*Remark 4 (The Case of Functional Repair):* The aforementioned theorem holds for the case of functional repair as well. Here, a sequence of failures and repairs is to be considered, starting from node 2 through node $k$ (in this order). The $d$ nodes assisting node $\ell+1$ $(1 \leq \ell \leq k-1)$ in its repair include the $\ell$ nodes in set $[\ell]$, and $W_{\ell+1}$ is the random variable corresponding to the data stored in node $\ell+1$ after its repair.

Next, we present a convenient way to represent all points on the storage-bandwidth tradeoff in terms of $\alpha$ and $\beta$.

### C. Representation for the Points on the Tradeoff

As previously discussed in Section I-C, for any $[n,\ k,\ d]$ optimal regenerating code, the parameter $\alpha$ lies in the range

$$(d-(k-1))\beta \ \leq\ \alpha \ \leq\ d\beta \tag{37}$$

Moreover, given the values of $\alpha$ and $\beta$, the point of operation depends on the value of $\alpha$ in comparison to $\beta$. More specifically, the evaluation of the minimum in the terms in the summation of (3) depends on the value of the integer $p$ such that $\alpha \in ((d-p-1)\beta, (d-p)\beta]$. In this case, the value of $B$ evaluates to

$$B = (p+1)\alpha + \sum_{i=p+1}^{k-1} (d-i)\beta$$

Based on the aforementioned discussion, in order to get a good handle on the point of operation on the tradeoff, we reparameterize $(\beta,\ \alpha)$ in terms of two new parameters $(p,\ \theta)$ as

$$\alpha = (d-p)\beta - \theta \tag{38}$$

for some $p$ and $\theta$ with $p \in \{0,\ldots,k-1\}$ and $\theta \in [0,\beta)$. Thus, the parameter $\theta$ serves the purpose of identifying the precise value of $\alpha$ in the range $((d-p-1)\beta, (d-p)\beta]$. Note that the range of $\alpha$ in (37) implies that for $p = k-1$, it must be that $\theta = 0$.

The storage-bandwidth tradeoff can, thus, be partitioned into the two end points and a middle region.

1) The MSR point: $p = k-1$ (which implies $\theta = 0$).
2) The MBR point: $p = 0,\ \theta = 0$.
3) The interior points: $p \in \{0,\ldots,k-2\}, \theta \in [0,\beta)$ except $\{p=0, \theta=0\}$.

For instance, the values of $(\alpha,\ \beta,\ p,\ \theta)$ at the four points depicted on the storage-bandwidth tradeoff in Fig. 2 are

MSR : (2700, 300, 9, 0)
$\text{I\!P}_1$ : (2786, 250, 6, 214)
$\text{I\!P}_2$ : (3300, 204, 1, 168)
MBR : (3600, 200, 0, 0).

### D. Properties of Exact-Repair Codes

We now present a set of properties that any exact-repair code with parameters satisfying the storage-bandwidth tradeoff with equality (3) must necessarily possess. These properties pertain to the random variables stored in the nodes and those passed for exact repair. The proofs of these properties are relegated to Appendix A.

The first two properties provide insights pertaining to the data stored in the nodes, and the subsequent properties provide insights about the data passed for repair.

*Property 1 (Entropy of the Data Stored):* For an arbitrary storage node $\ell$ $(1 \leq \ell \leq n)$

$$H\left(W_\ell\right) = \alpha \tag{39}$$

*Property 2 (Mutual Information Among the Nodes):* For a set $\mathcal{A}$ comprising an arbitrary collection of $a$ nodes, and an arbitrary node $\ell \notin \mathcal{A}$

$$I\left(W_\ell; W_{\mathcal{A}}\right) = \begin{cases} 0 & a \leq p \\ (a-p)\beta - \theta & p < a < k \\ \alpha = (d-p)\beta - \theta & a \geq k \end{cases} \tag{40}$$

Note that in the aforementioned Property 2, a threshold effect manifests itself twice in the mutual information, the first threshold occurring at $a = p+1$ and the second at $a = k$. This is illustrated in Fig. 4(a). This is a phenomenon similar (albeit more complex) to the single threshold effect in MDS codes where a code symbol has zero mutual information with up to $(k-1)$ other code symbols and has mutual information equal to its entropy with any $k$ or more other code symbols. Also plotted alongside in Fig. 4(b) and (c) are the behaviors of the two extreme points—the MBR $(p = 0,\ \theta = 0)$ and the MSR $(p = k-1,\ \theta = 0)$ points, respectively. Note that any MSR code is necessarily MDS.

An intuitive interpretation of Property 2 is as follows. The property attempts to evaluate the mutual information between the data stored in a node, and that stored in a set of nodes. In such a situation, there are two opposing forces in play: the reconstruction requirement which tends to limit this mutual information in order to allow any $k$ nodes to accumulate all information about the source data, and the repair requirement which requires this quantity to be large enough to allow exact repair to take place. The property shows that in a regenerating code satisfying (3), these two forces are in equilibrium. In order to attain the maximum possible value of $B$, the mutual information is zero when the cardinality of the set is small, and increases by the smallest possible increments that would allow for repair.

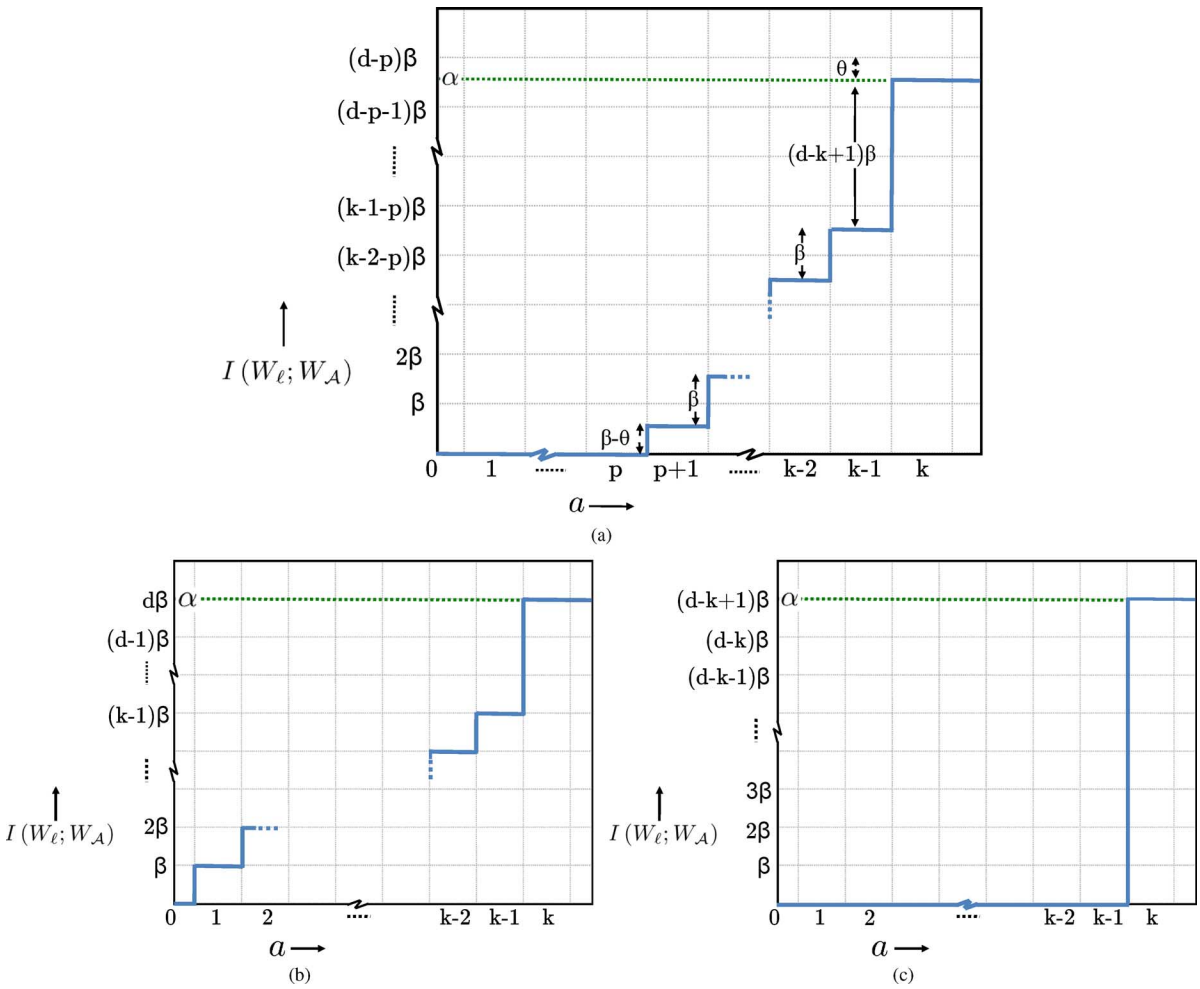


Fig. 4. Amount of information that a node $\ell$ has in common with a set $\mathcal{A}$ of $a$ other arbitrary nodes (Property 2). For (a) a general point on the tradeoff, (b) the MBR point, and (c) the MSR point.

*Corollary 5:* Consider a set $\mathcal{A}$ comprising an arbitrary collection of $a < k$ nodes. In the scenario where the set of $d$ helper nodes assisting in the repair of an arbitrary node $\ell \notin \mathcal{A}$ includes the $a$ nodes in $\mathcal{A}$, it must be that

$$H(W_\ell|S^\ell_{\mathcal{A}}) = \min\{\alpha,\ (d-a)\beta\} \tag{41}$$

*Property 3 (Entropy of the Data Passed):* In the scenario where node $m$ is an arbitrary node helping in the repair of a second arbitrary node $\ell$, it must be that

$$H\left(S^\ell_m\right) = \beta \tag{42}$$

irrespective of the identity of the $(d-1)$ other helper nodes.

*Helper Node Pooling:* Regenerating codes permit a failed node to choose an arbitrary set of $d$ remaining nodes to aid in its repair. In particular, this includes situations where nodes may form a pool and help each other in the repair process. More formally, consider a set $\mathcal{F}$ consisting of $f \leq (d+1)$ nodes, and a subset $\mathcal{R}$ of the set $\mathcal{F}$ consisting of $r$ nodes. We refer to "helper node pooling" as a scenario where on failure of any node $\ell \in \mathcal{R}$, the $d$ helper nodes assisting in its repair include the $(f-1)$ remaining nodes in $\mathcal{F}$. We denote the remaining $(d-(f-1))$ arbitrary helper nodes assisting in the repair of node $\ell$ by $\mathcal{V}(\ell)$.[7] The helper-node-pooling scenario is illustrated in Fig. 5.

Regenerating codes must necessarily satisfy helper-node-pooling scenarios. This leads to surprising (and as we shall see, implausible) upper bounds on the amount of information passed by a single helper node in the pool to multiple replacement nodes. In the following two properties, $S^\ell_m$ is used to denote the random variable corresponding to the data passed by node $m \in \mathcal{F}\backslash\mathcal{R}$ to assist in the repair of node $\ell \in \mathcal{R}$ in the scenario where the $d$ helper nodes are $\{(\mathcal{F}\backslash\{\ell\}) \cup \mathcal{V}(\ell)\}$.[8] Furthermore, in agreement with our earlier notation, we define

$$S^{\mathcal{R}}_m \triangleq \left\{S^\ell_m\right\}_{\ell\in\mathcal{R}}$$

[7]This notation is in anticipation of the usage of these sets in Properties 4 and 5, which consider the repair of each of the nodes in $\mathcal{R}$. In the scenario considered, every *replacement* node in $\mathcal{R}$ connects to the remaining $(f-1)$ nodes in $\mathcal{F}$, and hence, $\mathcal{F}$ forms a *fixed* set of helper nodes. On the other hand, the set $\mathcal{V}(\ell)$, comprising the $(d-(f-1))$ helper nodes of node $\ell \in \mathcal{R}$, is specific to node $\ell$ and is allowed to *vary* with $\ell$.

[8]The set $\mathcal{V}(\ell)$ representing the $(d-(f-1))$ arbitrary helper nodes assisting in the repair of node $\ell$ plays no role in the properties or the proofs. Hence, for ease of understanding, the reader may choose to assume this set also to be fixed, i.e., $\mathcal{V}(\ell) = \mathcal{V}$.

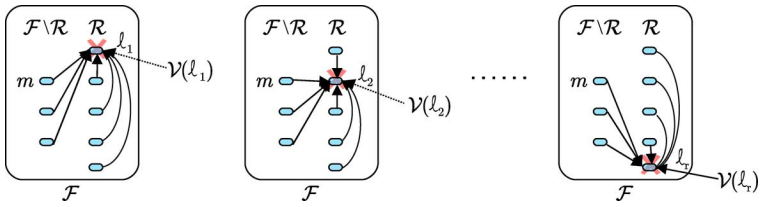


Fig. 5. Helper node pooling, and the setting of Properties 4 and 5. The nodes in the set $\mathcal{F}$ are partitioned into two: a subset $\mathcal{R}$ and the remainder $\mathcal{F}\setminus\mathcal{R}$. The repair of each of the nodes in $\mathcal{R}$, as in the helper-node-pooling scenario, is depicted.

*Property 4:* In the helper-node-pooling scenario where

$$\min\{k, f\} > p + 2 \geq r$$

for an arbitrary node $m \in \mathcal{F}\backslash\mathcal{R}$ it must be that

$$H\left(S_m^{\mathcal{R}}\right) \leq 2\beta - \theta \tag{43}$$

*Property 5:* In the helper-node-pooling scenario where

$$\min\{k, f\} > p + 1 \geq r \geq 2$$

for an arbitrary node $m \in \mathcal{F}\backslash\mathcal{R}$ and an arbitrary pair of nodes $\{\ell_1,\ \ell_2\} \in \mathcal{R}$, it must be that

$$H(S_m^{\ell_1} \mid S_m^{\ell_2}) \leq \theta \tag{44}$$

and hence

$$H\left(S_m^{\mathcal{R}}\right) \leq \beta + (r-1)\theta \tag{45}$$

The intuition behind these properties is as follows. In Property 5, the set $\mathcal{R}$ contains at most $(p+1)$ nodes. Hence, during repair of any node in $\mathcal{R}$, the other nodes in $\mathcal{R}$ cannot contribute "directly" as the mutual information between the data stored in these nodes, and that stored in the failed node is zero (from Property 2). It follows that for the repair of any node in $\mathcal{R}$, the mutual information between what a node $m \notin \mathcal{R}$ passes for repair, and what is stored in the failed node is required to be large. This is true for each of the $r$ nodes in $\mathcal{R}$. On the other hand, the data $S_m^{\ell}$ that node $m$ passes are a function of the data $W_m$ stored in it. In addition, from Property 2, the mutual information between $W_m$ and $W_{\mathcal{R}}$ is at most $(\beta - \theta)$. This forces node $m$ to pass highly correlated information for the repair of the $r$ nodes in $\mathcal{R}$. The intuition behind Property 4 closely follows the aforementioned argument, differing only in the cardinality of the set $\mathcal{R}$.

Properties 4 and 5 do not impose any constraints on systems operating at the MSR or the MBR point. At the MSR point, we have $p = k - 1$, and hence, the cardinality $r$ of the set $\mathcal{R}$ considered in these two properties evaluate to $k+1$ and $k$, respectively. However, since any $k$ nodes suffice to recover the entire data, the upper bound on the mutual information between $W_m$ and $W_{\mathcal{R}}$ evaluates simply to the trivial bound $\alpha$. At the MBR point, we have $p = 0$ and $\theta = 0$, and in the two properties, the cardinality of set $\mathcal{R}$ evaluates to 2 and 1, respectively. For these sizes, the upper bounds of $2\beta$ and $\beta$ on the cumulative data passed by node $m$ for the repair of the nodes in $\mathcal{R}$ turn out to be trivial.

*Remark 5:* The cardinality of the set $\mathcal{R}$ considered in Properties 4 and 5 are carefully chosen, and attempted analogous proofs for other cardinalities of $\mathcal{R}$ fail to yield tighter bounds.

### *E. Nonexistence Proof*

We now show that the properties derived in the preceding section overconstrain the system, causing a majority of the points on the storage-bandwidth tradeoff to be nonachievable under exact repair. Recall that the parameters $(\beta,\ \alpha)$ at any point on the tradeoff are written as

$$\alpha = (d-p)\beta - \theta$$

Here, the interior points correspond to $p \in \{0, \ldots, k-2\}$ and $\theta \in [0, \beta)$, except for the point $(p = 0, \theta = 0)$.

We first consider the case when $\alpha$ is a multiple of $\beta$, in Theorem 6. A majority of regenerating schemes and code constructions in the literature [5], [6], [10]–[15], [18] are designed for this case of $\alpha$ being a multiple of $\beta$. Thus, for this case, at the interior points it must be that

$$\alpha = (d-p)\beta, \quad \theta = 0$$

with $p$ lying in the range

$$1 \ \leq \ p \ \leq \ k-2 \tag{46}$$

*Theorem 6:* For any given values of $B$ and $[n,\ k,\ d]$, exact-repair codes do not exist for the parameters $(\beta,\ \alpha,\ B)$ lying at any interior point on the storage-bandwidth tradeoff with $\theta = 0$.

*Proof:* The proof is by contradiction: for any given values of the parameters $[n,\ k,\ d]$ and $(\beta,\ \alpha,\ B)$ such that $\theta = 0$, we assume the existence of an exact-repair code with these parameters and show that the code fails to satisfy some of the properties of exact-repair codes.

Let $\mathcal{G}$ denote the distributed storage network under consideration, and further, let $\mathcal{F}$ denote an arbitrary sub-network of $\mathcal{G}$ consisting of $(d+1)$ or greater nodes. As the notation suggests, in this proof, $\mathcal{F}$ plays the role of the fixed set in the helper-node-pooling scenario. Given an optimal exact-repair code for the network $\mathcal{G}$, it is clear that the code is also an optimal exact-repair code for the subnetwork $\mathcal{F}$, with the same parameter values $k,\ d,\ \beta,\ \alpha$, and $B$. In the present proof, we restrict our attention to a subnetwork $\mathcal{F}$ consisting of precisely $(d+1)$ nodes.

A brief outline of the proof is as follows. Clearly, the nodes in $\mathcal{F}$ form a helper node pool, and hence, Property 5 can be used to upper bound the total amount of information that a node can pass to aid in the repair of a set of nodes. This limits the cumulative information received by the nodes in $\mathcal{F}$ during their respective repair operations, which in turn limits the total data $B$ stored in the network to $(d+1)\beta$. Finally, the value of $B$ determined by the storage-bandwidth tradeoff (3) is found to be strictly larger.

Since the subnetwork $\mathcal{F}$ consists of $(d+1)$ nodes, a failed node $\ell \in \mathcal{F}$ is repaired with the assistance of the $d$ remaining

nodes in $\mathcal{F}$. Thus, for any node $\ell \in \mathcal{F}$, the exact-repair property requires

$$H(W_\ell \mid S^\ell_{\mathcal{F}\setminus\{\ell\}}) = 0 \tag{47}$$

Also, for any three distinct nodes $\{m,\ \ell_1,\ \ell_2\} \in \mathcal{F}$, Property 5 [see(44)] when $\theta = 0$ implies that

$$H(S^{\ell_1}_m \mid S^{\ell_2}_m) = 0 \tag{48}$$

Now, for a data collector to be able to recover the entire data by connecting to a set of $k$ nodes $\mathcal{K} \subset \mathcal{F}$ it must be that

$$\begin{aligned}
B &= H(W_\mathcal{K}) && (49)\\
&\le H(W_\mathcal{F}) && (50)\\
&= I\left(W_\mathcal{F}; \left\{S^\ell_{\mathcal{F}\setminus\{\ell\}}\right\}_{\ell\in\mathcal{F}}\right) && (51)\\
&\le H\left(\left\{S^\ell_{\mathcal{F}\setminus\{\ell\}}\right\}_{\ell\in\mathcal{F}}\right) && (52)\\
&= H\left(\left\{S^{\mathcal{F}\setminus\{m\}}_m\right\}_{m\in\mathcal{F}}\right) && (53)\\
&\le \sum_{m\in\mathcal{F}} H\left(S^{\mathcal{F}\setminus\{m\}}_m\right) && (54)\\
&= \sum_{m\in\mathcal{F}} \beta && (55)\\
&= (d+1)\beta && (56)
\end{aligned}$$

where (51) follows from the exact-repair requirement stated in (47), (53) is a re-writing of (52), and (55) employs (48) and Property 3.

On the other hand, since any optimal regenerating code must satisfy the storage-bandwidth tradeoff (3), it must be that

$$\begin{aligned}
B &= \sum_{i=0}^{k-1} \min\left(\alpha,\ (d-i)\beta\right) && (57)\\
&= \sum_{i=0}^{k-1} \min\left((d-p)\beta, (d-i)\beta\right) && (58)\\
&= 2(d-p)\beta + \sum_{i=2}^{k-1} \min\left((d-p)\beta, (d-i)\beta\right) && (59)\\
&\ge 2(d-p)\beta + (k-2)\beta && (60)\\
&\ge (d+2)\beta && (61)
\end{aligned}$$

where (58) holds since $\alpha = (d-p)\beta$, (59) holds since $p \ge 1$ [see (46)], (60) follows since each term in the summation in (59) is at least $\beta$, and (61) is derived using $d \ge k \ge p+2$. This is in contradiction to (56). ■

*Theorem 7:* For any given values of $B$ and $[n,\ k,\ d]$, exact-repair codes do not exist for the parameters $(\beta,\ \alpha,\ B)$ lying at any interior point on the storage-bandwidth tradeoff with $\theta \ne 0$, except possibly for the case

$$p = k-2 \text{ with } \left(\text{either } \theta \ge \frac{d-p-1}{d-p}\beta \text{ or } \ k = 2\right)$$

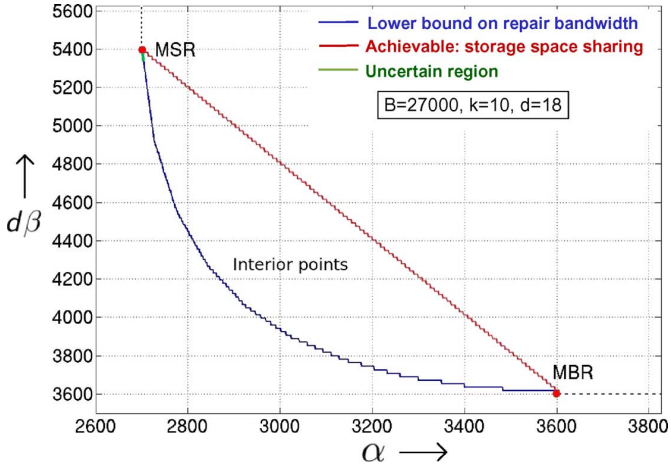


Fig. 6. Achievable value of repair bandwidth $d\beta$ for exact repair of all nodes plotted alongside the storage repair-bandwidth tradeoff curve, which is a lower bound on the repair bandwidth.

*Proof:* The proof of this theorem also exploits the existence of helper node pools in the system. Refer to Appendix B for the proof. ■

*Remark 6:* It can be verified that the properties derived in Section IV-D, and the nonachievability results in this section continue to hold even if optimal exact repair of only $k$ of the nodes is desired, and the remaining $(n-k)$ nodes are permitted to repair functionally with no restriction on the repair bandwidth.

### F. Achievable Curve Via Storage Space Sharing

We have seen that for a majority of the points in the interior of the tradeoff curve, the cut-set bound cannot be achieved under exact repair. On the other hand, from the *product-matrix* codes provided in [13], the cut-set bound can be achieved at the extreme points of the tradeoff curve: for all $[n,\ k,\ d]$ at the MBR point and for all $[n,\ k,\ d \ge 2k-2]$ at the MSR point. A linear storage-space-sharing scheme between these codes can be used to establish an achievable region between these two extreme points for exact repair. Given the system parameters $[n,\ k,\ d]$ and $(\alpha,\ B)$, with $d \ge 2k-2$, the net repair bandwidth $d\beta$ required under this scheme can be computed as

$$d\beta = \frac{2B - k\alpha}{k(d-k+1)} \tag{62}$$

Fig. 6 depicts the curve achieved via storage-space-sharing alongside the storage-bandwidth tradeoff curve for the parameters $[n > 18,\ k = 10,\ d = 18, \text{and } B = 27000]$.

## V. Conclusion

In this paper, an explicit exact MBR code for the parameters $[n,\ k,\ d = n-1]$ is presented. This code has very low repair complexity; repair of a failed node can be achieved by mere transfer of data and does not require any computation at either the helper nodes or the repair nodes. The ability of the code to perform repair by transfer minimizes the number of disk reads required at the helper nodes, and also permits the use of storage

disks with no extra intelligence. Moreover, this code, when specialized to the parameter set $[n,\ k=n-2,\ d=n-1]$, can be constructed over the binary field: repair does not require any computation, and the encoding and reconstruction processes require only XOR operations.

A set of properties that any exact-repair code must necessarily satisfy are derived. Specific scenarios termed helper node pooling are identified, which lead to upper bounds (that are surprisingly small) on the amount of information that a node can pass to assist in the repair of a set of nodes. These upper bounds are, then, used to show the nonachievability of almost all interior points on the storage-bandwidth tradeoff under exact repair.

## Appendix A
## Proofs of the Properties of Exact-Repair Codes

*Proof of Property 1:* Without loss of generality, assume $\ell=1$. Now, for reconstruction by a data collector connecting to the first $k$ nodes, it must be that

$$\begin{aligned} B &= H\left(W_{[k]}\right) && (63)\\ &= H(W_1)+\sum_{j=1}^{k-1} H\left(W_{j+1}\left|W_{[j]}\right.\right) && (64)\\ &\le \alpha+\sum_{j=1}^{k-1} H\left(W_{j+1}\left|W_{[j]}\right.\right) && (65)\\ &\le \alpha+\sum_{j=1}^{k-1}\min\{\alpha,\ (d-j)\beta\} && (66)\\ &= \sum_{j=0}^{k-1}\min\{\alpha,\ (d-j)\beta\} && (67)\\ &= B && (68)\end{aligned}$$

where (65) follows from (14), (66) results from Lemma 3, (67) uses the fact that $\alpha \le d\beta$ (from (37)), and (68) follows since we need to satisfy the storage-bandwidth tradeoff with equality (3). Thus, (65) must be satisfied with equality, which forces $H(W_1)=\alpha$. ∎

*Proof of Property 2:* The result clearly holds when $a \ge k$ since 1) data contained in any $k$ nodes suffice to recover the entire data, and 2) $H(W_\ell)=\alpha$ (from Property 1).

Now for the case when $a<k$, without loss of generality, we assume that the set $\mathcal{A}$ comprises of the first $a$ nodes in the system and node $\ell$ is the $(a+1)$th node, i.e., $\mathcal{A}=[a]$ and $\ell=a+1$. For reconstruction by a data collector connecting to the first $k$ nodes, we need

$$\begin{aligned} B &= H\left(W_{[k]}\right) && (69)\\ &= \sum_{j=0}^{k-1} H\left(W_{j+1}\left|W_{[j]}\right.\right) && (70)\\ &\le \sum_{j=0}^{k-1}\min\{\alpha,\ (d-j)\beta\} && (71)\\ &= B && (72)\end{aligned}$$

where (71) follows from Lemma 3 and (72) is a result of satisfying the storage-bandwidth tradeoff with equality (3). Thus, (71) must be satisfied with equality. This, coupled with the upper bound on each term $H(W_{j+1}|W_{[j]})$ from Lemma 3 gives (for the choice of $j$ as $a$)

$$H\left(W_{a+1}\left|W_{[a]}\right.\right)=\min\{\alpha,\ (d-a)\beta\} \tag{73}$$

Defining $(x-y)^+ \triangleq \max\{x-y,\ 0\}$ for any two numbers $x$ and $y$, and noting that $x-\min\{x,\ y\}=(x-y)^+$, it follows that

$$\begin{aligned} I\left(W_{a+1};W_{[a]}\right) &= H(W_{a+1})-H(W_{a+1}|W_{[a]}) && (74)\\ &= \alpha-\min\{\alpha,\ (d-a)\beta\} && (75)\\ &= (\alpha-(d-a)\beta)^+ && (76)\\ &= ((a-p)\beta-\theta)^+ && (77)\end{aligned}$$

where (75) follows from Property 1 and (73), and (77) follows since $\alpha=(d-p)\beta-\theta$. ∎

*Proof of Corollary 5:* From Lemma 3, it must be that

$$H(W_\ell|S_{\mathcal{A}}^\ell)\le\min\{\alpha,\ (d-a)\beta\} \tag{78}$$

On the other hand, since $H(S_m^\ell|W_m)=0$ for every node $m\in\mathcal{A}$

$$\begin{aligned} H(W_\ell|S_{\mathcal{A}}^\ell) &\ge H(W_\ell|W_{\mathcal{A}}) && (79)\\ &= H(W_\ell)-I(W_\ell;W_{\mathcal{A}}) && (80)\\ &= \alpha-(\alpha-(d-a)\beta)^+ && (81)\\ &= \min\{\alpha,\ (d-a)\beta\} && (82)\end{aligned}$$

where (81) employs Property 1 and Property 2 with $a<k$. ∎

*Proof of Property 3:* Partition the set of $d$ helper nodes assisting in the repair of node $\ell$ into a set $\mathcal{A}$ consisting of $(k-1)$ nodes and a second set $\mathcal{B}$ consisting of the remaining $(d-k+1)$ nodes, such that node $m\in\mathcal{B}$. Then, Corollary 5 mandates that

$$H(W_\ell|S_{\mathcal{A}}^\ell)=(d-k+1)\beta \tag{83}$$

However, exact repair of node $\ell$ requires [recall (17)]

$$H(W_\ell|S_{\mathcal{A}}^\ell,S_{\mathcal{B}}^\ell)=0 \tag{84}$$

From (83) and (84), it follows that

$$H\left(S_{\mathcal{B}}^\ell\right)\ge(d-k+1)\beta \tag{85}$$

Noting that each of the $(d-k+1)$ helper nodes in set $\mathcal{B}$ can pass no more than $\beta$ units of information, it must be that

$$H\left(S_{\mathcal{B}}^\ell\right)=(d-k+1)\beta \tag{86}$$

from which it follows that

$$H\left(S_m^\ell\right)=\beta \tag{87}$$

∎

*Proof of Property 4:* Clearly, if the statement holds for some values of $f$ and $r$, it continues to hold for all $f'$, $r'$ when $f'\ge f$ and $r'\le r$. Hence, throughout the proof, the set $\mathcal{R}$ is

assumed to be comprised of $r = p + 2$ nodes, and the set $\mathcal{F}$ is such that $\mathcal{F} = \mathcal{R} \cup \{m\}$. Thus, $f = p + 3$.

Consider repair of an arbitrary node $\ell \in \mathcal{R}$ where the set of $d$ helper nodes includes node $m$ and the $(p+1)$ remaining nodes in $\mathcal{R}$. As an intermediate step, we wish to prove $H\left(S_m^\ell \middle| W_{\mathcal{R}}\right) = 0$. For this, consider

$$\begin{aligned}
&I\left(S_m^\ell; W_{\mathcal{R}}\right) \\
&\quad = I\left(S_m^\ell; W_\ell, W_{\mathcal{R}\setminus\{\ell\}}\right) && (88)\\
&\quad = I\left(S_m^\ell; W_{\mathcal{R}\setminus\{\ell\}}\right) + I\left(S_m^\ell; W_\ell \middle| W_{\mathcal{R}\setminus\{\ell\}}\right) && (89)\\
&\quad \geq I\left(S_m^\ell; W_\ell \middle| W_{\mathcal{R}\setminus\{\ell\}}\right) && (90)\\
&\quad = H\left(W_\ell \middle| W_{\mathcal{R}\setminus\{\ell\}}\right) - H\left(W_\ell \middle| W_{\mathcal{R}\setminus\{\ell\}}, S_m^\ell\right) && (91)\\
&\quad \geq H\left(W_\ell \middle| W_{\mathcal{R}\setminus\{\ell\}}\right) - H\left(W_\ell \middle| S^\ell_{\mathcal{R}\setminus\{\ell\}}, S_m^\ell\right) && (92)\\
&\quad = (d-p-1)\beta - (d-p-2)\beta && (93)\\
&\quad = \beta && (94)
\end{aligned}$$

where (92) follows since $S^\ell_{\mathcal{R}\setminus\{\ell\}}$ is a function of $W_{\mathcal{R}\setminus\{\ell\}}$, and (93) follows from Property 2 and Corollary 5 with $r = p + 2 < k$. Then, it must be that

$$\begin{aligned}
H\left(S_m^\ell | W_{\mathcal{R}}\right) &= H\left(S_m^\ell\right) - I\left(S_m^\ell; W_{\mathcal{R}}\right) && (95)\\
&\leq \beta - \beta && (96)\\
&= 0 && (97)
\end{aligned}$$

and hence

$$H\left(S_m^\ell \middle| W_{\mathcal{R}}\right) = 0 \tag{98}$$

Now, since the choice of node $\ell$ from the set $\mathcal{R}$ was arbitrary, (98) holds for all $\ell \in \mathcal{R}$ and hence

$$H\left(S_m^{\mathcal{R}} \middle| W_{\mathcal{R}}\right) = 0 \tag{99}$$

It follows that

$$\begin{aligned}
H(S_m^{\mathcal{R}}) &= I(W_{\mathcal{R}}; S_m^{\mathcal{R}}) && (100)\\
&\leq I(W_{\mathcal{R}}; W_m) && (101)\\
&= 2\beta - \theta && (102)
\end{aligned}$$

where (102) follows from Property 2. ■

*Proof of Property 5:* The steps followed in this proof are similar to those in the proof of Property 4. Clearly, if the statement holds for some values of $f$ and $r$, it continues to hold for all $f', r'$ when $f' \geq f$ and $r' \leq r$. Hence, throughout the proof, the set $\mathcal{R}$ is assumed to be comprised of $r = p + 1$ nodes, and the set $\mathcal{F}$ is such that $\mathcal{F} = \mathcal{R} \cup \{m\}$. Thus, $f = p + 2$.

Consider repair of an arbitrary node $\ell \in \mathcal{R}$ where the set of $d$ helper nodes includes node $m$ and the $p$ remaining nodes in $\mathcal{R}$. As an intermediate step, we wish to prove $H\left(S_m^\ell \middle| W_{\mathcal{R}}\right) \leq \theta$. For this, consider

$$\begin{aligned}
&I\left(S_m^\ell; W_{\mathcal{R}}\right) \\
&\quad = I\left(S_m^\ell; W_\ell, W_{\mathcal{R}\setminus\{\ell\}}\right) && (103)\\
&\quad = I\left(S_m^\ell; W_{\mathcal{R}\setminus\{\ell\}}\right) + I\left(S_m^\ell; W_\ell \middle| W_{\mathcal{R}\setminus\{\ell\}}\right) && (104)\\
&\quad \geq I\left(S_m^\ell; W_\ell \middle| W_{\mathcal{R}\setminus\{\ell\}}\right) && (105)\\
&\quad = H\left(W_\ell \middle| W_{\mathcal{R}\setminus\{\ell\}}\right) - H\left(W_\ell \middle| W_{\mathcal{R}\setminus\{\ell\}}, S_m^\ell\right) && (106)\\
&\quad \geq H\left(W_\ell \middle| W_{\mathcal{R}\setminus\{\ell\}}\right) - H\left(W_\ell \middle| S^\ell_{\mathcal{R}\setminus\{\ell\}}, S_m^\ell\right) && (107)\\
&\quad = (d-p)\beta - \theta - (d-p-1)\beta && (108)\\
&\quad = \beta - \theta && (109)
\end{aligned}$$

where (107) follows since $S^\ell_{\mathcal{R}\setminus\{\ell\}}$ is a function of $W_{\mathcal{R}\setminus\{\ell\}}$ and (108) follows from Property 2 and Corollary 5 with $r = p+1 < k$. Then, it must be that

$$\begin{aligned}
H\left(S_m^\ell | W_{\mathcal{R}}\right) &= H\left(S_m^\ell\right) - I\left(S_m^\ell; W_{\mathcal{R}}\right) && (110)\\
&\leq \beta - (\beta - \theta) && (111)\\
&= \theta && (112)
\end{aligned}$$

Since the choice of node $\ell$ from the set $\mathcal{R}$ was arbitrary, (112) holds for all $\ell \in \mathcal{R}$.

Next, we prove $H(S_m^{\ell_1} | S_m^{\ell_2}) \leq \theta$ for an arbitrary pair of nodes $\{\ell_1,\ \ell_2\} \in \mathcal{R}$. For this, consider

$$\begin{aligned}
&H(S_m^{\ell_1}, S_m^{\ell_2}) \\
&\quad = I(W_{\mathcal{R}}; S_m^{\ell_1}, S_m^{\ell_2}) + H(S_m^{\ell_1}, S_m^{\ell_2} | W_{\mathcal{R}}) && (113)\\
&\quad \leq I(W_{\mathcal{R}}; W_m) + H(S_m^{\ell_1}, S_m^{\ell_2} | W_{\mathcal{R}}) && (114)\\
&\quad = I(W_{\mathcal{R}}; W_m) + H(S_m^{\ell_1} | W_{\mathcal{R}}) \\
&\qquad + H(S_m^{\ell_2} \big| W_{\mathcal{R}}, S_m^{\ell_1}) && (115)\\
&\quad \leq (\beta - \theta) + \theta + \theta && (116)\\
&\quad = \beta + \theta && (117)
\end{aligned}$$

where (114) follows since $S_m^{\ell_1}$ and $S_m^{\ell_2}$ are functions of $W_m$, and (116) follows from Property 2 and (112). Then, it must be that

$$\begin{aligned}
H(S_m^{\ell_1} | S_m^{\ell_2}) &= H(S_m^{\ell_1}, S_m^{\ell_2}) - H(S_m^{\ell_2}) && (118)\\
&\leq (\beta + \theta) - \beta && (119)\\
&= \theta && (120)
\end{aligned}$$

Finally, ordering the nodes in $\mathcal{R}$ in an arbitrary manner as $\{\ell_i \mid 1 \leq i \leq r\}$ and noting that (120) holds for every pair of nodes in $\mathcal{R}$, we have

$$\begin{aligned}
H(S_m^{\mathcal{R}}) &\leq H(S_m^{\ell_1}) + \sum_{i=2}^{r} H(S_m^{\ell_i} | S_m^{\ell_{i-1}}) && (121)\\
&\leq \beta + (r-1)\theta && (122)
\end{aligned}$$

■

## Appendix B
## Proof of Theorem 7

*Proof:* The proof is by contradiction: for any given values of the parameters $[n,\ k,\ d]$ and $(\beta,\ \alpha,\ B)$ such that $\theta \neq 0$, we assume the existence of an exact-repair code with these parameters and show that the properties of exact-repair codes lead to a contradiction.

As in the proof in Theorem 6, we restrict our attention to a subnetwork $\mathcal{F}$ of the distributed storage network $\mathcal{G}$ consisting of $(d + 1)$ nodes, and ignore the remaining nodes in $\mathcal{G}$. Thus, on failure of a node in this subnetwork $\mathcal{F}$, the $d$ helper nodes comprise of the $d$ remaining nodes in $\mathcal{F}$.

We consider the processes of the exact repair of two nodes, nodes $\ell$ and $m$. We partition the $(d - 1)$ remaining nodes in

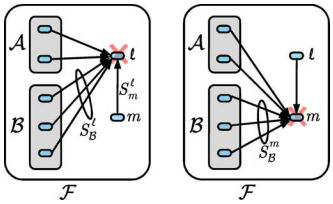


Fig. 7. Setting and notation for the proof of Theorem 7: repair of nodes $\ell$ and $m$ using the remaining $d$ nodes in $\mathcal{F}$.

$\mathcal{F}$ into two sets: set $\mathcal{A}$ of cardinality $p$ and set $\mathcal{B}$ of cardinality $(d-p-1)$. The proof, in a nutshell, uses the properties established earlier in this paper to obtain bounds on the quantity $H\left(S_{\mathcal{B}}^{\ell}, S_{\mathcal{B}}^{m}, S_{m}^{\ell}\right)$ (Fig. 7 depicts the setting and relevant parameters). First, the restriction that Property 2 imposes on the mutual information of nodes $m$ and $\ell$ with the nodes in $\mathcal{A}$ is used to obtain a lower bound on the amount of information passed by the $(d-p-1)$ nodes in $\mathcal{B}$ for the repair of nodes $\ell$ and $m$. Next, Properties 3 and 4 are invoked to obtain an upper bound on the information passed by any single node (in $\mathcal{B}$) to repair nodes $\ell$ and $m$. From this, it turns out that the total amount of information that can be passed by the nodes in $\mathcal{B}$ falls short of the amount required for the repair. Details follow.

Restricting ourselves to the subnetwork $\mathcal{F}$, exact repair of nodes $\ell$ and $m$ requires [from (17)]

$$H\left(W_{\ell}\left|S_{\mathcal{A}}^{\ell}, S_{\mathcal{B}}^{\ell}, S_{m}^{\ell}\right.\right)=0 \tag{123}$$
$$H\left(W_{m}\left|S_{\mathcal{A}}^{m}, S_{\mathcal{B}}^{m}, S_{\ell}^{m}\right.\right)=0 \tag{124}$$

respectively. However, since $S_i^m$ is a function of $W_i$ for every helper node $i$, it follows from the two aforementioned equations that

$$\begin{aligned} &H\left(W_{\ell}, W_{m}\left|W_{\mathcal{A}}, S_{\mathcal{B}}^{\ell}, S_{\mathcal{B}}^{m}, S_{m}^{\ell}\right.\right) \\ &\quad=H\left(W_{\ell}\left|W_{\mathcal{A}}, S_{\mathcal{B}}^{\ell}, S_{\mathcal{B}}^{m}, S_{m}^{\ell}\right.\right) \\ &\qquad+H\left(W_{m}\left|W_{\ell}, W_{\mathcal{A}}, S_{\mathcal{B}}^{\ell}, S_{\mathcal{B}}^{m}, S_{m}^{\ell}\right.\right) &(125)\\ &\quad=0 &(126)\end{aligned}$$

We first derive a lower bound on the quantity $H(S_{\mathcal{B}}^{\ell}, S_{\mathcal{B}}^{m}, S_{m}^{\ell})$, as follows:

$$\begin{aligned} &H(S_{\mathcal{B}}^{\ell}, S_{\mathcal{B}}^{m}, S_{m}^{\ell}) \\ &\quad\geq H(S_{\mathcal{B}}^{\ell}, S_{\mathcal{B}}^{m}, S_{m}^{\ell}|W_{\mathcal{A}}) &(127)\\ &\quad\geq I(S_{\mathcal{B}}^{\ell}, S_{\mathcal{B}}^{m}, S_{m}^{\ell}; W_{\ell}, W_{m}|W_{\mathcal{A}}) &(128)\\ &\quad=H(W_{\ell}, W_{m}|W_{\mathcal{A}}) \\ &\qquad-H(W_{\ell}, W_{m}|W_{\mathcal{A}}, S_{\mathcal{B}}^{\ell}, S_{\mathcal{B}}^{m}, S_{m}^{\ell}) &(129)\\ &\quad=H\left(W_{\ell}, W_{m}\left|W_{\mathcal{A}}\right.\right) &(130)\\ &\quad=H\left(W_{\ell}\left|W_{\mathcal{A}}\right.\right)+H\left(W_{m}\left|W_{\mathcal{A}}, W_{\ell}\right.\right) &(131)\\ &\quad=H(W_{\ell})-I\left(W_{\ell}; W_{\mathcal{A}}\right)+H(W_{m}) \\ &\qquad-I\left(W_{m}; W_{\mathcal{A}}, W_{\ell}\right) &(132)\\ &\quad=\alpha-0+\alpha-(\beta-\theta) &(133)\\ &\quad=(2d-2p-1)\beta-\theta &(134)\end{aligned}$$

where (130) follows from (126), and (133) follows from Properties 1 and 2.

Next, we obtain an upper bound on the quantity $H\left(S_{\mathcal{B}}^{\ell}, S_{\mathcal{B}}^{m}, S_{m}^{\ell}\right)$. We consider the case of $p+2<k$ and the case of $p+2=k$ separately.

*Case 1: $p+2<k$:* In this case

$$\begin{aligned} H\left(S_{\mathcal{B}}^{\ell}, S_{\mathcal{B}}^{m}, S_{m}^{\ell}\right) &\leq \sum_{i\in\mathcal{B}} H\left(S_{i}^{\ell}, S_{i}^{m}\right)+H\left(S_{m}^{\ell}\right) &(135)\\ &\leq \sum_{i\in\mathcal{B}}(2\beta-\theta)+\beta &(136)\\ &=(2d-2p-1)\beta-(d-p-1)\theta &(137)\end{aligned}$$

where (136) follows from Properties 3 and 4. Since $\theta\neq 0$ and $d\geq k>p+2$, the inequalities in (134) and (137) are in contradiction.

*Case 2: $p+2=k$:* In this case, Property 5 is used to obtain an upper bound on $H\left(S_{\mathcal{B}}^{\ell}, S_{\mathcal{B}}^{m}, S_{m}^{\ell}\right)$. Note that this property does not hold when $k=2$, and hence, we consider the case when $k>2$

$$\begin{aligned} H\left(S_{\mathcal{B}}^{\ell}, S_{\mathcal{B}}^{m}, S_{m}^{\ell}\right) &\leq \sum_{i\in\mathcal{B}} H\left(S_{i}^{\ell}, S_{i}^{m}\right)+H\left(S_{m}^{\ell}\right) &(138)\\ &\leq \sum_{i\in\mathcal{B}}(\beta+\theta)+\beta &(139)\\ &=(d-p)\beta+(d-p-1)\theta &(140)\end{aligned}$$

where (139) follows from Properties 3 and 5. Clearly, the inequalities in (140) and (134) contradict when

$$\theta<\frac{d-p-1}{d-p}\beta \tag{141}$$

■

## ACKNOWLEDGMENT

This study was done while N. B. Shah and K. V. Rashmi were with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore, India.

## REFERENCES

[1] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, Chicago, IL, Jun. 1988, pp. 109–116.

[2] S. Rhea, P. Eaton, D. Geels, H. Weatherspoon, B. Zhao, and J. Kubiatowicz, "Pond: The OceanStore prototype," in *Proc. 2nd USENIX Conf. File Storage Technol.*, 2003, pp. 1–14.

[3] R. Bhagwan, K. Tati, Y. C. Cheng, S. Savage, and G. M. Voelker, "Total recall: System support for automated availability management," in *Proc. 1st Conf. Symp. Network. Syst. Design Implement. (NSDI)*, 2004, pp. 337–350.

[4] A. G. Dimakis, P. B. Godfrey, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," in *Proc. 26th IEEE Int. Conf. Comput. Commun.*, Anchorage, AK, May 2007, pp. 2000–2008.

[5] Y. Wu and A. Dimakis, "Reducing repair traffic for erasure coding-based storage via interference alignment," in *Proc. IEEE Int. Symp. Inf. Theory*, Seoul, Korea, Jul. 2009, pp. 2276–2280.

[6] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit construction of optimal exact regenerating codes for distributed storage," in *Proc. 47th Annu. Allerton Conf. Commun., Control, Comput.*, Urbana-Champaign, IL, Sep. 2009, pp. 1243–1249.

[7] Y. Wu, A. G. Dimakis, and K. Ramchandran, "Deterministic regenerating codes for distributed storage," presented at the presented at the 45th Annu. Allerton Conf. Control, Comput., Commun., Urbana-Champaign, IL, Sep. 2007.
[8] Y. Wu, "Existence and construction of capacity-achieving network codes for distributed storage," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 277–288, Feb. 2010.
[9] [Online]. Available: http://www.eecs.berkeley.edu/~nihar/storagevideo.html,2010
[10] 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. Theory*, to be published.
[11] D. Cullina, A. G. Dimakis, and T. Ho, "Searching for minimum storage regenerating codes," presented at the presented at the 47th Annu. Allerton Conf. Commun., Control, Comput., Urbana-Champaign, IL, Sep. 2009.
[12] Y. Wu, "A construction of systematic MDS codes with minimum repair bandwidth," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3738–3741, Jun. 2011.
[13] 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.
[14] N. B. Shah, K. V. Rashmi, P. V. Kumar, and K. Ramchandran, "Explicit codes minimizing repair bandwidth for distributed storage," in *Proc. IEEE Inf. Theory Workshop*, Cairo, Egypt, Jan. 2010, pp. 1–5.
[15] C. Suh and K. Ramchandran, "Exact-repair MDS code construction using interference alignment," *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1425–1442, Mar. 2011.
[16] V. R. Cadambe, S. A. Jafar, and H. Maleki, "Distributed data storage with minimum storage regenerating codes—Exact and functional repair are asymptotically equally efficient," 2010 [Online]. Available: arXiv:1004.4299 [cs.IT]
[17] C. Suh and K. Ramchandran, "On the existence of optimal exact-repair MDS codes for distributed storage," 2010 [Online]. Available: arXiv:1004.4663 [cs.IT]
[18] B. Gastón and J. Pujol, "Double circulant minimum storage regenerating codes," 2010 [Online]. Available: arXiv:1007.2401 [cs.IT]
[19] F. Oggier and A. Datta, "Self-repairing homomorphic codes for distributed storage systems," 2010 [Online]. Available: arXiv:1008.0064 [cs.IT]
[20] N. B. Shah, K. V. Rashmi, and P. V. Kumar, "A flexible class of regenerating codes for distributed storage," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, Jun. 2010, pp. 1943–1947.
[21] A. M. Kermarrec, G. Straub, and N. L. Scouarnec, "Repairing multiple failures with coordinated and adaptive regenerating codes," 2011 [Online]. Available: arXiv:1102.0204 [cs.IT]
[22] Y. Hu, Y. Xu, X. Wang, C. Zhan, and P. Li, "Cooperative recovery of distributed storage systems from multiple losses with network coding," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 2, pp. 268–276, Feb. 2010.
[23] S. Akhlaghi, A. Kiani, and M. R. Ghanavati, "A fundamental trade-off between the download cost and repair bandwidth in distributed storage systems," in *Proc. IEEE Int. Symp. Netw. Coding*, Toronto, ON, Jun. 2010, pp. 1–6.
[24] K. V. Rashmi, N. B. Shah, and P. V. Kumar, "Enabling node repair in any erasure code for distributed storage," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2011, pp. 1235–1239.
[25] K. V. Rashmi, N. B. Shah, P. V. Kumar, and K. Ramchandran, "Explicit and optimal exact-regenerating codes for the minimum-bandwidth point in distributed storage," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, Jun. 2010, pp. 1938–1942.
[26] S. El Rouayheb and K. Ramchandran, "Fractional repetition codes for repair in distributed storage systems," presented at the 48th Annu. Allerton Conf. Control, Comput., Commun., Urbana-Champaign, IL, Sep. 2010.
[27] Y. Hu, C. Yu, Y. Li, P. Lee, and J. Lui, "NCFS: On the practicality and extensibility of a network-coding-based distributed file system," presented at the Int. Symp. Netw. Coding, Beijing, China, Jul. 2011.
[28] S. Pawar, S. El Rouayheb, and K. Ramchandran, "On secure distributed data storage under repair dynamics," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, Jun. 2010, pp. 2543–2547.
[29] S. Pawar, S. El Rouayheb, and K. Ramchandran, "Securing dynamic distributed storage systems from malicious nodes," in *Proc. IEEE Int. Symp. Inf. Theory*, St. Petersburg, Russia, Jul. 2011, pp. 1452–1456.

**Nihar B. Shah** is a Ph.D. student at the Department of Electrical Engineering and Computer Science at the University of California at Berkeley. He received his M.E. degree from the Indian Institute of Science (IISc), Bangalore in 2010. He is a recipient of the Prof. S.V.C. Aiya medal for the best master-of-engineering student in the ECE department at IISc. His research interests include coding and information theory, algorithms, and statistical inference.

**K. V. Rashmi** is pursuing her Ph.D. in the Department of Electrical Engineering and Computer Science at the University of California at Berkeley. She received her M.E. degree from the Indian Institute of Science (IISc), Bangalore in 2010. Her research interests include coding theory, information theory, networks, communications and signal processing, with a current focus on coding for data storage networks and network coding.

**P. Vijay Kumar** (S'80–M'82–SM'01–F'02) received the B.Tech. and M.Tech. degrees from the Indian Institutes of Technology (Kharagpur and Kanpur), and the Ph.D. Degree from the University of Southern California (USC) in 1983, all in Electrical Engineering. From 1983–2003 he was on the faculty of the EE-Systems Department at USC. Since 2003 he has been on the faculty of the Indian Institute of Science, Bangalore and also holds the position of adjunct research professor at USC. His current research interests include codes for distributed storage, distributed function computation, sensor networks and space-time codes for MIMO and cooperative communication networks. He is a fellow of the IEEE and an ISI highly-cited author. He is co-recipient of the 1995 IEEE Information Theory Society prize paper award as well as of a best paper award at the DCOSS 2008 conference on sensor networks.

**Kannan Ramchandran** (Ph.D.'93, Columbia University) is a Professor of Electrical Engineering and Computer Science at the University of California at Berkeley, where he has been since 1999. Prior to that, he was with the University of Illinois at Urbana-Champaign from 1993 to 1999, and was at AT&T Bell Laboratories from 1984 to 1990. He is a Fellow of the IEEE and has won numerous awards including the Eli Jury thesis award at Columbia, a couple of Best Paper awards from the IEEE Signal Processing Society, a Hank Magnusky Scholar award at Illinois, an Okawa Foundation Research Prize at Berkeley, an Outstanding Teaching Award from the EECS Department at UC Berkeley, and has co-authored several best student paper awards at conferences and workshops. His current research interests include distributed signal processing and coding for wireless systems, coding for distributed storage, peer-to-peer networking and video content delivery, security, and multi-user information and communication theory.