Codes for Distributed Storage
Other titles in Foundations and Trends® in Communications and Information Theory

*Rank-Metric Codes and Their Applications*
Hannes Bartz, Lukas Holzbaur, Hedongliang Liu, Sven Puchinger, Julian Renner and Antonia Wachter-Zeh
ISBN: 978-1-63828-000-2

*Common Information, Noise Stability, and Their Extensions*
Lei Yu and Vincent Y. F. Tan
ISBN: 978-1-63828-014-9

*Information-Theoretic Foundations of DNA Data Storage*
Ilan Shomorony and Reinhard Heckel

*Asymptotic Frame Theory for Analog Coding*
Marina Haikin, Matan Gavish, Dustin G. Mixon and Ram Zamir
ISBN: 978-1-68083-908-1

*Modeling and Optimization of Latency in Erasure-coded Storage Systems*
Vaneet Aggarwal and Tian Lan
ISBN: 978-1-68083-842-8

*An Algebraic and Probabilistic Framework for Network Information Theory*
S. Sandeep Pradhan, Arun Padakandla and Farhad Shirani
ISBN: 978-1-68083-766-7
Codes for Distributed Storage

Vinayak Ramkumar
Indian Institute of Science, Bengaluru
vinram93@gmail.com

S. B. Balaji
Qualcomm, Bengaluru
balaji.profess@gmail.com

Birenjith Sasidharan
Govt. Engineering College, Barton Hill, Trivandrum
birenjith@gmail.com

Myna Vajha
Qualcomm, Bengaluru
mynaramana@gmail.com

M. Nikhil Krishnan
International Institute of Information Technology Bangalore
nikhilkrishnan.m@gmail.com

P. Vijay Kumar
Indian Institute of Science, Bengaluru
pvk1729@gmail.com

Full text available at: http://dx.doi.org/10.1561/0100000115
Editorial Scope

Topics

Foundations and Trends® in Communications and Information Theory publishes survey and tutorial articles in the following topics:

- Coded modulation
- Coding theory and practice
- Communication complexity
- Communication system design
- Cryptology and data security
- Data compression
- Data networks
- Demodulation and Equalization
- Denoising
- Detection and estimation
- Information theory and statistics
- Information theory and computer science
- Joint source/channel coding
- Modulation and signal design
- Multiuser detection
- Multiuser information theory
- Optical communication channels
- Pattern recognition and learning
- Quantization
- Quantum information processing
- Rate-distortion theory
- Shannon theory
- Signal processing for communications
- Source coding
- Storage and recording codes
- Speech and Image Compression
- Wireless Communications

Information for Librarians

Foundations and Trends® in Communications and Information Theory, 2022, Volume 19, 4 issues. ISSN paper version 1567-2190. ISSN online version 1567-2328. Also available as a combined paper and online subscription.
# Contents

1 Introduction 3
   1.1 Conventional Repair of an MDS Code .................. 3
   1.2 Regenerating Codes and Locally Recoverable Codes ...... 5
   1.3 Overview of the Monograph .......................... 7

2 Maximum Distance Separable Codes 11
   2.1 Reed-Solomon Codes ..................................... 11
   2.2 Singleton Bound ........................................ 13
   2.3 Generalized Reed-Solomon Codes ........................ 14
   2.4 Systematic Encoding .................................... 16
   2.5 Cauchy MDS Codes ....................................... 16

3 Regenerating Codes 21
   3.1 Definition and Terminology .............................. 21
   3.2 Bound on File Size ...................................... 26
   3.3 Storage-Repair-Bandwidth Tradeoff ..................... 30
   3.4 Network Coding Approach to the File-Size Bound ...... 33
   3.5 Overview of RGC-Related Topics in the Monograph ...... 36

4 MBR Codes 38
   4.1 Polygonal MBR Code ...................................... 39
   4.2 Product-Matrix MBR Code .................................. 41
10.7  Tamo-Barg LRC .............................................. 135
10.8  Bounds on $d_{\text{min}}$ and Rate for Nonlinear LRCs .... 140
10.9  Extended Notions of Locality ............................... 142

11 Codes with Availability .................................. 149
  11.1  Linear Availability Codes ................................. 153
  11.2  Constructions of Linear Availability Codes .......... 153
  11.3  Upper Bounds on $d_{\text{min}}$ of Linear Availability Codes ... 157
  11.4  Strict Availability ....................................... 162

12 LRCs with Sequential Recovery .......................... 168
  12.1  Recovery from Two or Three Erasures ................. 169
  12.2  The General Case ....................................... 174

13 Hierarchical Locality ................................... 183
  13.1  An Upper Bound on $d_{\text{min}}$ .......................... 186
  13.2  Optimal Constructions .................................. 189

14 Maximally Recoverable Codes .......................... 193
  14.1  Recoverable Erasure Patterns ........................... 194
  14.2  Defining Maximally Recoverable Codes ............... 197
  14.3  Existence of MRCs ...................................... 199
  14.4  MRCs Constructed using Linearized Polynomials ....... 201
  14.5  Reduced Field-Size Construction for the Disjoint Locality Case ........................................... 203

15 Codes with Combined Locality and Regeneration .... 210
  15.1  Locality of a Code with Vector Alphabet .............. 210
  15.2  Codes with MSR/MBR Locality .......................... 212

16 Repair of Reed-Solomon Codes .......................... 219
  16.1  Vectorization Approach .................................. 219
  16.2  Tools Employed ......................................... 221
  16.3  Guruswami-Wootters Repair Scheme .................... 224
  16.4  Dau-Milenkovic Repair Scheme .......................... 225
  16.5  Bounds on Repair-Bandwidth ............................. 227
Codes for Distributed Storage

Vinayak Ramkumar¹, S. B. Balaji², Birenjith Sasidharan³, Myna Vajha⁴, M. Nikhil Krishnan⁵ and P. Vijay Kumar⁶

¹ Indian Institute of Science, Bengaluru, India; vinram93@gmail.com
² Qualcomm, Bengaluru, India; balaji.profess@gmail.com
³ Govt. Engineering College, Barton Hill, Trivandrum, India; birenjith@gmail.com
⁴ Qualcomm, Bengaluru, India; mynaramana@gmail.com
⁵ International Institute of Information Technology Bangalore, India; nikhilkrishnan.m@gmail.com
⁶ Indian Institute of Science, Bengaluru, India; pvk1729@gmail.com

ABSTRACT

In distributed data storage, information pertaining to a given data file is stored across multiple storage units or nodes in redundant fashion to protect against the principal concern, namely, the possibility of data loss arising from the failure of individual nodes. The simplest form of such protection is replication. The explosive growth in the amount of data generated on a daily basis brought up a second major concern, namely minimization of the overhead associated with such redundant storage. This concern led to the adoption by the storage industry of erasure-recovery codes such as Reed-Solomon (RS) codes and more generally, maximum distance separable codes, as these codes offer the lowest-possible storage overhead for a given level of reliability.

In the setting of a large data center, where the amount of stored data can run into several exabytes, a third concern
arises, namely the need for efficient recovery from a commonplace occurrence, the failure of a single storage unit. One measure of efficiency in node repair is how small one can make the amount of data download needed to repair a failed unit, termed the repair bandwidth. This was the subject of the seminal paper by Dimakis et al. [50] in which an entirely new class of codes called regenerating codes was introduced, that within a certain repair framework, had the minimum-possible repair bandwidth. A second measure relates to the number of helper nodes contacted for node repair, termed the repair degree. A low repair degree is desirable as this means that a smaller number of nodes are impacted by the failure of a given node. The landmark paper by Gopalan et al. [72] focuses on this second measure, leading to the development of the theory of locally recoverable codes. The two events also led to the creation of a third class of codes known as locally regenerating codes, where the aim is to simultaneously achieve reduced repair bandwidth and low repair degree. Research in a different direction led researchers to take a fresh look at the challenge of efficient RS-code repair, and led to the identification of improved repair schemes for RS codes that have significantly reduced repair bandwidth.

This monograph introduces the reader to these different approaches towards efficient node repair and presents many of the fundamental bounds and code constructions that have since emerged. Several open problems are identified, and many of the sections have a notes subsection at the end that provides additional background.
Given the failure-prone nature of a storage device, reliability against data loss has always been of paramount importance in the storage industry. In the early days, this was achieved through simple replication of data, for example, triple replication was a commonplace selection within the Hadoop distributed file system (HDFS). However, the explosive growth in the amount of data stored over the past couple of decades encouraged the industry to look for other means of ensuring reliability and having less storage overhead. Here, the class of maximum distance separable (MDS) codes are a natural choice as they incur the least amount of storage overhead for a given level of protection, measured in terms of the maximum number of node failures that can be tolerated.

1.1 Conventional Repair of an MDS Code

Many of the schemes employed in redundant array of independent disks (RAID) technology make use of MDS codes. An \([n, k]\) MDS code is a block code of length \(n\) and dimension \(k\) over a suitably-defined finite field. To store data using an \([n, k]\) MDS code, the data file is first partitioned into \(k\) equal-sized fragments, that are then stored on \(k\) distinct storage units. An additional set of \(r = (n - k)\) fragments
of redundant data are then created and stored on a further set of \( r \) storage units in such a manner that the contents of any \( k \) out of the \( n \) storage units suffice to recover the data. In this way, the contents of a file are efficiently stored in redundant fashion, across a set of \( n \) storage units. For example, RAID 6 makes use of a \([5, 3]\) MDS code. Other examples of MDS codes that appear in the erasure coded-version HDFS-EC of HDFS are a \([9, 6]\) MDS code as well as a \([14, 10]\) code, the latter employed by Facebook. Throughout the monograph, we will alternately refer to a storage unit as a node.

Today’s data centers store massive amounts of information, amounts that can run into several exabytes, i.e., \( 10^{18} \) bytes. While protection against data loss and maintaining low values of storage overhead continue to be of primary importance, a third concern has recently surfaced. This has to do with the efficiency with which a failed storage unit can be repaired. We will view the repair process as one in which a new storage unit, which we will term as the replacement node, is brought in as a substitute for a failed storage unit. The replacement node then draws from the partial or entire contents of all or a subset of the remaining \((n - 1)\) nodes, and uses the data so received to replicate the contents of the original failed node.

As is well known, an \([n, k, d_{\text{min}}]\) code \( C \) is protected against data loss if the number of node failures does not exceed \((d_{\text{min}} - 1)\). For a given value of \((d_{\text{min}} - 1)\), MDS codes in general, and Reed-Solomon (RS) codes in particular, have the least possible value of storage overhead given by \( \frac{n}{k} = \frac{n}{n-d_{\text{min}}+1} \). This follows as the minimum distance \( d_{\text{min}} \) of an MDS code satisfies \( d_{\text{min}} = (n-k+1) \), which by the Singleton bound [156] is the largest value possible. In coding-theoretic terms, the problem of node repair is equivalent to recovery from erasure of a single code symbol. The most obvious approach is to invoke a parity-check (p-c) equation involving the erased code symbol. Let

\[
C = (c_1 \ c_2 \ \ldots \ c_n)
\]

be a code word and let us assume without loss of generality, erasure of the first code symbol \( c_1 \). Any p-c equation involving \( c_1 \) of the form

\[
\sum_{i=1}^{n} h_i c_i = 0, \quad h_1 \neq 0,
\]
is associated to a codeword

$$h = (h_1, h_2, \ldots, h_n)$$

belonging to the \([n, n-k]\) dual code \(C^\perp\). In the case of an MDS code \(C\), its dual \(C^\perp\) is also an MDS code and hence has parameters \([n, n-k, k+1]\). Thus any codeword \(h\) in \(C^\perp\) has Hamming weight \(w_H(h) \geq k + 1\). Thus if a p-c equation

$$\sum_{i=1}^{n} h_i c_i = 0,$$

is used to recover the code symbol \(c_1\), then we have

$$c_1 = \sum_{i=2}^{n} \left( -\frac{h_i}{h_1} \right) c_i,$$

(1.1)

with at least \(k\) terms of the form \(-\frac{h_i}{h_1}\) on the right side being nonzero.

1.2 Regenerating Codes and Locally Recoverable Codes

For the operation of a data center, equation (1.1) has two implications. Firstly, that the replacement of the failed node must necessarily contact \(k\) “helper nodes”, i.e., nodes that store the code symbols \(\{c_i \mid \frac{h_i}{h_1} \neq 0\}\). Secondly, equation (1.1) suggests that each helper node must transfer its entire contents (represented by \(c_i\)) for repair of the failed node. The number of helper nodes contacted (at least \(k\) in the case of an MDS code) is called the repair degree of the code. The total amount of data downloaded for repair of the failed node is termed the repair bandwidth. In the case of an MDS code, it is clear that the repair bandwidth is at least \(k\) times the amount of data stored in the failed node.

This is illustrated below in the case of an \([14, 10]\) MDS code. Assume a data file of size equal to 1 GB. The data file is partitioned into 10 fragments, each of size 100 MB and each data fragment is stored in a different node. Four parity nodes are then created, corresponding to the four parity symbols of the MDS code. The contents of the 14 nodes can be regarded as the layering of \(10^8\) codewords, each belonging to the \([14, 10]\) MDS code over \(\mathbb{F}_{2^8}\). Fig. 1.1 shows repair of a failed node. As
can be seen, there are $k = 10$ helper nodes corresponding to nodes 2 through 11 and each helper node passes on the 100 MB of data or parity stored in the respective node, to the repair center. Thus in this case the repair degree equals 10 and the repair bandwidth equals $10 \times 100 \text{ MB} = 1 \text{ GB}$.

Seminal papers by Dimakis et al. [50] and Gopalan et al. [72] heralded the theory of two entirely new classes of erasure-recovery codes, termed as regenerating codes (RGCs) and locally recoverable codes (LRCs), that were designed with the express aim of lowering the repair bandwidth and repair degree respectively. The development of the theory of RGCs and LRCs also led to the creation of a class of codes termed as locally regenerating codes by Kamath et al. [117] and Rawat et al. [189], where the aim is to simultaneously achieve reduced repair bandwidth and low repair degree. Research in a slightly different direction, pioneered by Shanmugam et al. [215] and Guruswami and Wootters [85], led to a re-examination of the repair bandwidth of RS codes and the design of more efficient repair schemes that permitted node repair with reduced repair bandwidth.

As an indication of the kind of impact that research on the topics of RGCs and LRCs has had on the development of coding theory, we note that papers reporting research in this area have received many best paper awards over the years. The list includes [50], [62], [72], [103], [137], [185], [228], [229], [237], [255].
1.3 Overview of the Monograph

This monograph presents an overview of how research on the topic of codes for distributed storage has evolved in a certain direction (see Fig. 1.2 for an overview of topics covered here). There have been several excellent prior surveys on the topic, including those found in [46], [51], [136], [145]. Additionally, concise surveys by the authors of the present monograph can be found in [10], [178].

Given the vast nature of the literature on the topic of codes for distributed storage, we have undoubtedly missed many papers that have made a strong contribution. We apologize in advance to the authors of these papers for the inadvertent omission. Furthermore, as can be seen from the listing of topics in Fig. 1.2, our focus here is only on certain specific approaches to coding for distributed storage.

![Diagram](http://dx.doi.org/10.1561/0100000115)

**Figure 1.2:** An overview of the coverage of codes for distributed storage in this monograph.

**MDS Codes** Section 2 provides background on MDS, RS codes and a generalization of RS codes known as generalized RS (GRS) codes.

**Regenerating Codes** The next seven sections deal with RGCs. The definition of an RGC along with a fundamental upper bound on file size is presented in Section 3. The bound reveals that there is a tradeoff between the storage overhead and the repair bandwidth. Sections 4 and 5 present constructions for the two main classes of RGCs, namely minimum bandwidth regenerating (MBR) codes and minimum storage regenerating (MSR) codes, that lie at the two ends of the storage-repair
Introduction

bandwidth tradeoff. The tradeoff itself is explored in the following section, Section 6. Constructions for RGCs that lie on interior points of the tradeoff are presented in Section 7. The sub-packetization level of an RGC may be regarded as denoting the number of symbols stored per node. An alternate viewpoint is to regard a regenerating code as a code over a vector symbol alphabet of the form $\mathbb{F}_q^\alpha$, with $\alpha$ denoting the sub-packetization level. Lower values of sub-packetization are desirable in practice, as a large sub-packetization level, apart from increasing the complexity of implementation, also limits the smallest size of a file that can be stored. Section 8 presents lower bounds on the sub-packetization level of an MSR code.

Several variants of RGCs have been explored in the literature. Piggyback codes, $\epsilon$-MSR codes and the codes of Li-Liu-Tang, are MDS codes that have reduced repair bandwidth and much smaller sub-packetization level. Cooperative RGCs explore the cooperative repair of a set of $t > 1$ failed nodes. Secure RGCs are designed to provide security in the presence of an eavesdropper or an active adversary. Rack-aware RGCs are designed to minimize the amount of cross-rack repair data that is transferred. An erasure-recovery code is said to possess the repair-by-transfer (RBT) property, if it enables repair of a failed node without need for computation at either helper or replacement node. Fractional repetition codes form a class of erasure-recovery codes that possesses the repair-by-transfer property and can be viewed as a generalization of a class of RBT MBR codes. The former codes potentially offer reduced storage overhead at the cost of reduced freedom in the selection of helper nodes. All these variants of RGCs can be found discussed in Section 9.

Locally Recoverable Codes  As noted above, the need for repair of a failed node with low degree prompted the creation of LRCs. Section 10 introduces LRCs and presents an upper bound on the rate and minimum distance of an LRC as well as optimal code constructions.

One means of handling the simultaneous failure of several nodes with low repair degree is to make the local codes that are at the core of an LRC more powerful. There are other approaches however, each with its own advantages and disadvantages. The three sections that follow present these other approaches. Availability codes, discussed in
Section 11, represent one such example. This class of codes has the additional feature that in the case of a single erased node, there are multiple, node-disjoint means of recovering from the node failure. This can be a very useful feature to have in practice, particularly as a means of handling cases when there are multiple simultaneous demands for the data contained within a particular node.

Sequential-recovery LRCs place the least stringent conditions on an LRC for the local recovery from multiple erasures, and consequently, have smallest possible storage overhead. These are discussed in Section 12. If an LRC has large block length and small value of repair degree \( r \), and a particular local code is overwhelmed by erasures, the only option is to fall back on the properties of the full-length block code to recover from the erasure pattern, leading to a sharp increase in the repair degree. Codes with hierarchical locality, discussed in Section 13, are designed to address this situation, provide layers of local codes having increasing block length as well as erasure-recovery capability, and permit a more graceful degradation in repair degree with an increasing number of erasures.

Maximally recoverable codes (MRCs), discussed in Section 14, may be regarded as the subclass of LRCs that are as MDS as possible in the sense that every set of \( k \) columns of the generator matrix of an MRC is a linearly independent set, unless the locality constraints imposed make it impossible for this to happen. An MRC is maximal in the sense that if an MRC is not able to recover from an erasure pattern, then no other code satisfying the same locality constraints can possibly recover from the same erasure pattern.

**Locally Regenerating Codes** Section 15 introduces a class of codes in which the local codes are themselves regenerating codes. As a result, these codes simultaneously offer both low repair degree as well as low repair bandwidth.

**Improved Repair Schemes for RS Codes** The evolution of RGCs and LRCs spurred researchers to take a fresh look at the challenge of efficient RS-code repair and led to the identification of improved repair
schemes for RS codes having significantly reduced repair bandwidth. These developments are described in Section 16.

**Codes in Practice**  The final section, Section 17, discusses the impact that the theoretical developments discussed in this monograph have had in practice.
References


References


[177] V. A. Rameshwar and N. Kashyap, “Achieving secrecy capacity of minimum storage regenerating codes for all feasible (n, k, d) parameter values,” in National Conference on Communications, Bangalore, India, 2019, pp. 1–6, 2019.


References


Index

\[(n, M, d_{\text{min}})\text{ code, 12}\]
\[[n, k, d_{\text{min}}]\text{ code, 12}\]
\[\epsilon\text{-MSR code, 111, 113, 114}\]
active limited-knowledge adversary model, 120
active omniscient adversary model, 120
algebraic geometry codes, 147
all-symbol locality, 135, 184
annihilator polynomial, 138, 190
anti-code, 166
Azure, 134, 183, 184, 234
balanced incomplete block design, 165, 166
Beehive code, 233
binary MBR codes, 44
bipartite graph, 105
Butterfly code, 233
Cascade code, 82, 87
Cauchy matrix, 16, 17, 41
Cauchy MDS codes, 16
centralized repair, 57, 119
Ceph, 233, 235
Chinese remainder theorem, 190, 191
Clay code, 233
codes with availability, 149
codes with MBR locality, 215
codes with MSR locality, 215, 216
Combinatorial Nullstellensatz, 35, 67, 68, 115, 116, 200, 201
constant-repair-matrix property, 66
cooperative locality, 143
regenerating code, 119
repair, 57, 119
corner points, 70
coset, 138, 140, 190
coupled-layer MSR code, 58, 233
cowedge multiplication, 91
cross-rack repair bandwidth, 123
cut-set bound, 33, 230, 231
data collection, 22, 24, 26, 27
data cube, 59
Determinant code, 82, 87
Diagonal MSR code, 52, 114
disjoint locality, 203, 208, 212
exact repair, 22, 23, 32
exact repair tradeoff, 31, 73, 80
excluded erasure patterns, 195
exterior product, 90
file-size bound, 26, 27, 30, 33
fractional repetition codes, 43, 116
full Reed-Solomon code, 221
full-rank condition, 98
functional repair, 22, 24, 30–32, 34, 36
functional repair tradeoff, 31, 70
generalized Hamming weight, 158
generalized Reed-Solomon codes, 7, 14, 221
girth, 176, 177, 179, 180
global parity symbols, 134
good polynomial, 138–140
grid-like topology, 209
Hamming distance, 12
Hamming weight, 12
Hashtag code, 234
HDFS, 3, 230, 233, 234
help-by-transfer, 38
helper nodes, 5, 6, 23, 117, 124
helper-set-independence property, 66, 82
Hitchhiker, 234
Hoffman-Singleton graph, 177
inclusion-exclusion, 151
information-symbol locality, 126, 184
interference alignment, 97, 98
interior point, 36, 74, 80
intersection score, 63
Lagrange interpolation, 14, 18
lazy repair, 37
lexicographic ordering, 154
linear availability codes, 153
linear LRC, 125
linear repair scheme, 227, 229, 230
linearized polynomials, 94, 201–203, 225, 226
liquid storage, 37
local parity symbols, 134
locally recoverable codes, 124
MBR codes, 32, 38
MDS array codes, 20, 32
MDS codes, 3, 13
MDS conjecture, 20
middle codes, 185
Index

minimum bandwidth
  cooperative
  regenerating point, 120
minimum bandwidth
  rack-aware
  regenerating point, 123
minimum distance, 12
minimum storage cooperative
  regenerating point, 120
minimum storage rack-aware
  regenerating point, 123
minimum weight, 12
monic polynomial, 11
Moore graph, 176, 177, 179
Moulin code, 88
MSR codes, 32, 45
multilinear algebra, 68, 88
multiple erasures, 168, 230

NCCloud, 232
near-optimal repair bandwidth, 111
network coding, 31, 33, 35
node, 4
node repair, 5, 22, 24, 117
nonlinear LRC, 125
normalized repair bandwidth, 30

optimal regenerating code, 30
optimal-access MSR code, 45, 58, 68, 102, 233
optimal-access repair, 230
optimal-update MSR code, 57
outer bounds, 76

p-c equation, 4

pairwise
  forward transform, 61
  reverse transform, 61
parity nodes, 5
passive eavesdropper model, 120, 122
pentagon MBR code, 39, 216, 233
Permuted-Diagonal MSR code, 69
piecewise linear, 70
piggybacking framework, 112, 234
polygonal MBR code, 39
product code, 153, 174
product-matrix
  framework, 41
  MBR, 41
  MSR, 46, 233
pyramid code, 132, 134
rack-aware regenerating code, 122
RAID, 3, 234
rank profile, 212
rate of
  PM-MSR code, 52
  MBR code, 33
  RGC, 23
recoverable erasure patterns, 194
reduced field-size constructions
  of MRC, 208
Reed-Solomon codes, 4, 11
regenerating codes, 21
repair bandwidth, 5, 6, 21, 23,
Index

38, 45, 210, 219, 220, 224, 227, 229, 230
repair degree, 5, 6, 124, 183, 210
repair matrix, 97, 103
repair polynomials, 222–224, 226
repair subspace, 97, 99, 103
repair-by-transfer, 38, 39, 116, 117
replacement node, 4, 22
resilient regenerating code, 121
secure
  MBR code, 122
  MSR code, 122
  regenerating code, 120
sequential recovery, 168
shortening, 50, 51, 65, 66
Signed Determinant code, 81
simplex code, 166
Singleton bound, 13, 126
Steiner system, 117
Steiner triple system, 166
storage overhead, 23, 30, 45
storage-repair-bandwidth tradeoff, 30
strict availability, 162
sub-packetization level, 8, 23, 220, 229, 231
subgroup, 138, 140
systematic code, 16, 17, 50
systematic MSR codes, 68
table-based repair, 43, 117
Tamo-Barg LRC, 136, 216
tensor product, 89
trace function, 221, 225
trace-dual basis, 221, 222
uniform rank accumulation codes, 212, 214
Vandermonde matrix, 12, 14, 41, 47, 55
vector code, 210, 211
vector symbol alphabet, 219
Xorbas, 234
Zigzag code, 67

Full text available at: http://dx.doi.org/10.1561/0100000115