Erasure codes for distributed storage and related problems, Part II

Alexander Barg

University of Maryland, College Park

NASIT, July 2019

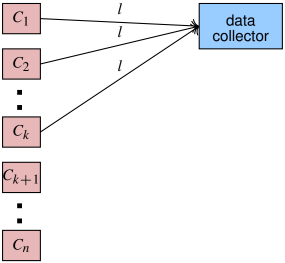
THE MAIN MESSAGE OF THIS TUTORIAL:

- The task of node repair in distributed storage gives rise to a range of new, previously unrecognized problems in coding theory and related areas of computer science and discrete mathematics.
- These problems have been actively studied for the past decade and led to the emergence of new methods and ideas in these areas.
- The goal of this tutorial is to introduce these methods and the associated results as well as to point out new research directions.

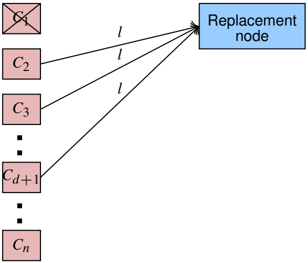
Repair Bandwidth: Motivation

- General problem: Correct a single erasure in the encoding
  - This is a new problem (2010) with unexpected answers
- Most codes correct one erasure; certainly, RS codes do.
- As mentioned before, we may need to “download” large volume of data
- What is the smallest amount of data send to decoder to correct one erasure?
- Do we gain in the repair bandwidth by downloading data from many nodes?

Coding tasks in storage

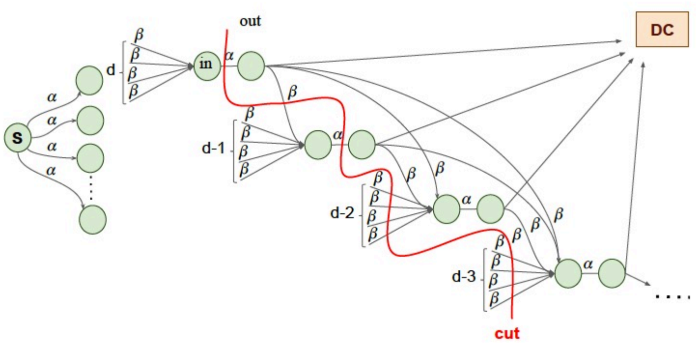


Data collection



Node repair

Information flow graph



A.G. DIMAKIS, P.B. GODDFREY, Y. WU, M.J. WAINWRIGHT, AND K. RAMCHANDRAN, *Network coding for distributed storage*, T-IT, 2010

Cutset bound

A file of size $\mathcal{B}$ is encoded into n nodes $C_i, i = 1, \dots, n$

- Each node has size (capacity) l
- k nodes suffice to recover the data
- d helper nodes are used to repair a failed node
- Helper node i contributes β_i symbols for node repair

Cutset bound

A file of size $\mathcal{B}$ is encoded into n nodes $C_i, i = 1, \dots, n$

- Each node has size (capacity) l
- k nodes suffice to recover the data
- d helper nodes are used to repair a failed node
- Helper node i contributes β_i symbols for node repair

General cutset bound (network coding):

$$\mathcal{B} \leqslant \sum_{i=1}^{k} \min\{l, (d-i)\beta_i\}$$

Cutset bound

A file of size $\mathcal{B}$ is encoded into n nodes $C_i, i = 1, \dots, n$

- Each node has size (capacity) l
- k nodes suffice to recover the data
- d helper nodes are used to repair a failed node
- Helper node i contributes β_i symbols for node repair

General cutset bound (network coding):

$$\mathcal{B} \leqslant \sum_{i=1}^{k} \min\{l, (d-i)\beta_i\}$$

Minimum storage (MSR) codes

$$l = \frac{\mathcal{B}}{k}$$

$$\beta_i = \frac{\mathcal{B}}{k(d-k+1)}$$

Minimum bandwidth (MBR) codes

$$l = d\beta_i$$

$$\beta_i = \frac{2\mathcal{B}}{k(2d-k+1)}$$

A.G. Dimakis, P.B. Goddfrey, Y. Wu, M.J. Wainwright, and K. Ramchandran, *Network coding for distributed storage*, T-IT, 2010

The repair problem

- We say that an (n, k, l) code over $F = \mathbb{F}_q$ has the optimal repair propery if the repair bandwidth meets the cutset bound

The repair problem

- We say that an (n, k, l) code over $F = \mathbb{F}_q$ has the optimal repair propery if the repair bandwidth meets the cutset bound
- In addition, optimize q and l

The repair problem

- We say that an (n, k, l) code over $F = \mathbb{F}_q$ has the optimal repair propery if the repair bandwidth meets the cutset bound
- In addition, optimize q and l
- The repair problem is essentially the first step in expanding coding to network environment
- How can information be stored and recovered in networks?
- Network coding was the first example, addressing a limited version of the question
- Multiple research directions arise

Regenerating codes

Regenerating codes

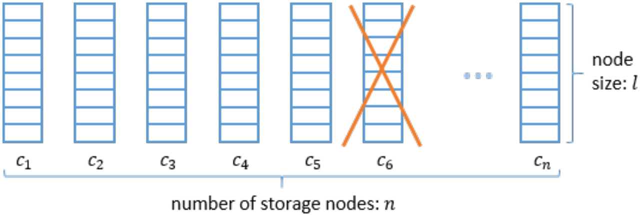
- C_i is a function of the information acquired from the coordinates $C_j, j \in \mathcal{R}$, where $\mathcal{R} \subset [n], |\mathcal{R}| = d \geqslant k$

Regenerating codes

- C_i is a function of the information acquired from the coordinates $C_j, j \in \mathcal{R}$, where $\mathcal{R} \subset [n], |\mathcal{R}| = d \geqslant k$
- In other words, there are functions $f_j : F^l \to F^a, j \in \mathcal{R}$ whose values jointly form the arguments for function $g_i : F^{da} \to F_l$ that recovers C_i

Regenerating codes

- C_i is a function of the information acquired from the coordinates $C_j, j \in \mathcal{R}$, where $\mathcal{R} \subset [n], |\mathcal{R}| = d \geqslant k$
- In other words, there are functions $f_j : F^l \to F^a, j \in \mathcal{R}$ whose values jointly form the arguments for function $g_i : F^{da} \to F_l$ that recovers C_i
- Our terminology is inspired by the application
  - C_i - failed node
  - $C_j, j \in \mathcal{R}$ the set of helper nodes; d - repair degree
  - $\{f_j(C_j), j \in \mathcal{R}\}$ downloaded information



MDS array codes for storage

[1] We use "coordinates" and "nodes" interchangeably

MDS array codes for storage

Each coordinate[1] of the codeword $(C_1, C_2, \ldots, C_n) \in F^n$ is an l-dimensional vector over F, so the codeword can be viewed as an $l \times n$ array over F

- (n, k, l) MDS array code:

[1] We use "coordinates" and "nodes" interchangeably

MDS array codes for storage

Each coordinate[1] of the codeword $(C_1, C_2, \ldots, C_n) \in F^n$ is an l-dimensional vector over F, so the codeword can be viewed as an $l \times n$ array over F

- (n, k, l) MDS array code:
  - code length n

[1] We use "coordinates" and "nodes" interchangeably

MDS array codes for storage

Each coordinate[1] of the codeword $(C_1, C_2, \ldots, C_n) \in F^n$ is an l-dimensional vector over F, so the codeword can be viewed as an $l \times n$ array over F

- (n, k, l) MDS array code:
  - code length n
  - k data nodes

[1] We use "coordinates" and "nodes" interchangeably

MDS array codes for storage

Each coordinate[1] of the codeword $(C_1, C_2, \ldots, C_n) \in F^n$ is an l-dimensional vector over F, so the codeword can be viewed as an $l \times n$ array over F

- (n, k, l) MDS array code:
  - code length n
  - k data nodes
  - $r = n - k$ parity nodes
  - MDS property: Contents of any r nodes can be determined by the other k nodes.

[1] We use "coordinates" and "nodes" interchangeably

MDS array codes for storage

Each coordinate[1] of the codeword $(C_1, C_2, \ldots, C_n) \in F^n$ is an l-dimensional vector over F, so the codeword can be viewed as an $l \times n$ array over F

- (n, k, l) MDS array code:
  - code length n
  - k data nodes
  - $r = n - k$ parity nodes
  - MDS property: Contents of any r nodes can be determined by the other k nodes.
  - The value of l is called sub-packetization of the code $\mathcal{C}$

[1] We use "coordinates" and "nodes" interchangeably

MDS array codes for storage

Each coordinate[1] of the codeword $(C_1, C_2, \ldots, C_n) \in F^n$ is an l-dimensional vector over F, so the codeword can be viewed as an $l \times n$ array over F

- (n, k, l) MDS array code:
  - code length n
  - k data nodes
  - $r = n - k$ parity nodes
  - MDS property: Contents of any r nodes can be determined by the other k nodes.
  - The value of l is called sub-packetization of the code $\mathcal{C}$
  - $\mathcal{C}$ is called a linear array code (or a vector code) if it is F-linear. It may not be F^l linear; if it is, it is also called a scalar code.

[1] We use "coordinates" and "nodes" interchangeably

MDS array codes for storage

Each coordinate[1] of the codeword $(C_1, C_2, \ldots, C_n) \in F^n$ is an l-dimensional vector over F, so the codeword can be viewed as an $l \times n$ array over F

- (n, k, l) MDS array code:
  - code length n
  - k data nodes
  - $r = n - k$ parity nodes
  - MDS property: Contents of any r nodes can be determined by the other k nodes.
  - The value of l is called sub-packetization of the code $\mathcal{C}$
  - $\mathcal{C}$ is called a linear array code (or a vector code) if it is F-linear. It may not be F^l linear; if it is, it is also called a scalar code.
  - MSR codes are necessarily MDS array codes.

[1] We use “coordinates” and “nodes” interchangeably

Cutset bound, the MSR case

- $\mathcal{C}$ a vector MDS code: every node is an l vector over F
- File of size kl
- Any k nodes suffice to decode

Cutset bound, the MSR case

- $\mathcal{C}$ a vector MDS code: every node is an l vector over F
- File of size kl
- Any k nodes suffice to decode

Lemma (A.G. Dimakis et al., 2010)

Suppose a node is repaired from d helper nodes, $k \leqslant d \leqslant n-1$. The repair bandwidth is at least

$$\beta = \frac{dl}{d-k+1}$$

Cutset bound, the MSR case

- $\mathcal{C}$ a vector MDS code: every node is an l vector over F
- File of size kl
- Any k nodes suffice to decode

Lemma (A.G. Dimakis et al., 2010)

Suppose a node is repaired from d helper nodes, $k \leqslant d \leqslant n-1$. The repair bandwidth is at least

$$\beta = \frac{dl}{d-k+1}$$

Proof:

- $\mathcal{C}$ is MDS $\Leftrightarrow$ no $k-1$ nodes carry any information about erased node

Cutset bound, the MSR case

Lemma (A.G. Dimakis et al., 2010)

Suppose a node is repaired from d helper nodes, $k \leqslant d \leqslant n-1$. The repair bandwidth is at least

$$\beta = \frac{dl}{d-k+1}$$

Proof:

- $\mathcal{C}$ is MDS $\Leftrightarrow$ no $k-1$ nodes carry any information about erased node
- $\Rightarrow$ From any $d-k+1$ nodes we should gain $\geqslant l$ symbols of F

Cutset bound, the MSR case

Lemma (A.G. Dimakis et al., 2010)

Suppose a node is repaired from d helper nodes, $k \leqslant d \leqslant n-1$. The repair bandwidth is at least

$$\beta = \frac{dl}{d-k+1}$$

Proof:

- $\mathcal{C}$ is MDS $\Leftrightarrow$ no $k-1$ nodes carry any information about erased node
- $\Rightarrow$ From any $d-k+1$ nodes we should gain $\geqslant l$ symbols of F
- Let $\mathcal{R} \subset [n], |\mathcal{R}| = d$ be the helper set, let $\mathcal{I} \subset \mathcal{R}$, $|\mathcal{I}| = k-1$

$$\beta(\mathcal{R} \backslash \mathcal{I}) := \sum_{i \in \mathcal{R} \backslash \mathcal{I}} \beta_i \geqslant l$$

Cutset bound, the MSR case

Lemma (A.G. Dimakis et al., 2010)

Suppose a node is repaired from d helper nodes, $k \leqslant d \leqslant n-1$. The repair bandwidth is at least

$$\beta = \frac{dl}{d-k+1}$$

Proof:

- $\mathcal{C}$ is MDS $\Leftrightarrow$ no $k-1$ nodes carry any information about erased node
- $\Rightarrow$ From any $d-k+1$ nodes we should gain $\geqslant l$ symbols of F
- Let $\mathcal{R} \subset [n]$, $|\mathcal{R}| = d$ be the helper set, let $\mathcal{I} \subset \mathcal{R}$, $|\mathcal{I}| = k-1$

$$\beta(\mathcal{R} \backslash \mathcal{I}) := \sum_{i \in \mathcal{R} \backslash \mathcal{I}} \beta_i \geqslant l$$

- $$\sum_{\substack{\mathcal{I} \subset \mathcal{R} \\ |\mathcal{I}|=k-1}} \sum_{i \in \mathcal{R} \backslash \mathcal{I}} \beta_i \geqslant \binom{d}{k-1} l$$

Lemma (A.G. Dimakis et al., 2010)

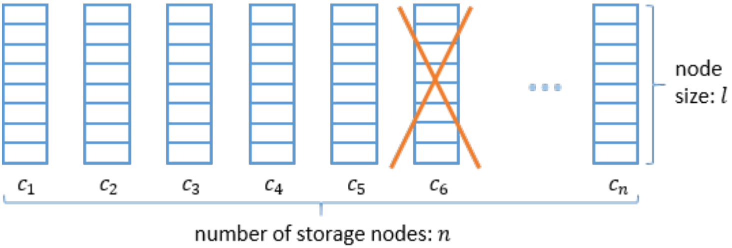
Suppose a node is repaired from d helper nodes, $k \leqslant d \leqslant n-1$. The repair bandwidth is at least

$$\beta = \frac{dl}{d-k+1}$$

Proof:

- $\mathcal{C}$ is MDS $\Leftrightarrow$ no $k-1$ nodes carry any information about erased node
- $$\sum_{i\in\mathcal{R}}\Bigg(\sum_{\substack{\mathcal{I}\subset\mathcal{T},i\in\mathcal{I}\\|\mathcal{I}|=k-1}}\beta_i\Bigg)=\binom{d-1}{k-1}\sum_{i\in\mathcal{R}}\beta_i\geqslant\binom{d}{k-1}l$$

The repair problem



Formal definition of the (single-node) repair problem

- Consider an (n,k,l) code $\mathcal{C}$ over B.

Formal definition of the (single-node) repair problem

- Consider an (n, k, l) code $\mathcal{C}$ over B.
- A codeword $C = (C_1, \dots, C_n)$, where $C_i = (c_{i,0}, c_{i,1}, \dots, c_{i,l-1})^T \in B^l, i = 1, \dots, n$.

Formal definition of the (single-node) repair problem

- Consider an (n, k, l) code $\mathcal{C}$ over B.
- A codeword $C = (C_1, \dots, C_n)$, where $C_i = (c_{i,0}, c_{i,1}, \dots, c_{i,l-1})^T \in B^l, i = 1, \dots, n$.
- A node $i \in [n]$ can be *repaired* from a subset of $d \geqslant k$ *helper nodes* $\mathcal{R}_i \subset [n] \backslash \{i\}$, by *downloading* $\beta_i(\mathcal{R}_i)$ symbols of B if there are
  - numbers $\beta_{i,j}, j \in \mathcal{R}_i$ and
  - d functions $f_{i,j} : B^l \to B^{\beta_{i,j}}, j \in \mathcal{R}_i$ and a function $g_i : B^{\sum_j \beta_{i,j}} \to B^l$

  such that

$$C_i = g_i(f_{i,j}(C_j), j \in \mathcal{R}_i)$$

  and

$$\sum_{j \in \mathcal{R}_i} \beta_{i,j} = \beta_i(\mathcal{R}_i).$$

Formal definition of the (single-node) repair problem

- Consider an (n,k,l) code $\mathcal{C}$ over B.
- A codeword $C=(C_1,\dots,C_n)$, where $C_i=(c_{i,0},c_{i,1},\dots,c_{i,l-1})^T\in B^l, i=1,\dots,n$.
- A node $i\in[n]$ can be *repaired* from a subset of $d\geqslant k$ *helper nodes* $\mathcal{R}_i\subset[n]\backslash\{i\}$, by *downloading* $\beta_i(\mathcal{R}_i)$ symbols of B if there are
  - numbers $\beta_{i,j}, j\in\mathcal{R}_i$ and
  - d functions $f_{i,j}:B^l\to B^{\beta_{i,j}}, j\in\mathcal{R}_i$ and a function $g_i:B^{\sum_j\beta_{i,j}}\to B^l$

  such that

$$C_i=g_i(f_{i,j}(C_j),j\in\mathcal{R}_i)$$

  and

$$\sum_{j\in\mathcal{R}_i}\beta_{i,j}=\beta_i(\mathcal{R}_i).$$

The repair bandwidth of i from $\mathcal{R}_i$:

$$\beta_i^*(\mathcal{R}_i)=\min_{f_{i,j},g_i}\beta_i(\mathcal{R}_i)$$

Constructions of Vector (Array) Codes

Low-rate regime $k \leqslant (n+1)/2$

Single-node repair: Product-matrix and other constructions of codes with d-optimal repair property (RASHMI-SHAH-KUMAR '11; RASHMI-SHAH-KUMAR '12; SUH-RAMCHANRDAN '11)

Constructions of Vector (Array) Codes

Low-rate regime $k \leqslant (n+1)/2$

Single-node repair: Product-matrix and other constructions of codes with d-optimal repair property (RASHMI-SHAH-KUMAR '11; RASHMI-SHAH-KUMAR '12; SUH-RAMCHANRDAN '11)

Existence proofs of codes for any k: (CADAMBE ET AL. '11, '12; PAPAILIOPOULOS ET AL. '13; TAMO-WANG-BRUCK '13; GOPARAJU-FAZELI-VARDY '16)

Constructions of Vector (Array) Codes

Low-rate regime $k \leqslant (n+1)/2$

Single-node repair: Product-matrix and other constructions of codes with d-optimal repair property (RASHMI-SHAH-KUMAR '11; RASHMI-SHAH-KUMAR '12; SUH-RAMCHANRDAN '11)

Existence proofs of codes for any k: (CADAMBE ET AL. '11, '12; PAPAILIOPOULOS ET AL. '13; TAMO-WANG-BRUCK '13; GOPARAJU-FAZELI-VARDY '16)

Any parameters including $k > (n+1)/2$

Constructions of Vector (Array) Codes

Low-rate regime $k \leqslant (n+1)/2$

Single-node repair: Product-matrix and other constructions of codes with d-optimal repair property (RASHMI-SHAH-KUMAR '11; RASHMI-SHAH-KUMAR '12; SUH-RAMCHANRDAN '11)

Existence proofs of codes for any k: (CADAMBE ET AL. '11, '12; PAPAILIOPOULOS ET AL. '13; TAMO-WANG-BRUCK '13; GOPARAJU-FAZELI-VARDY '16)

Any parameters including $k > (n+1)/2$

- (n, k) MDS codes with optimal repair and $l = r^n$, $d = n-1$;

Constructions of Vector (Array) Codes

Low-rate regime $k \leqslant (n+1)/2$

Single-node repair: Product-matrix and other constructions of codes with d-optimal repair property (RASHMI-SHAH-KUMAR '11; RASHMI-SHAH-KUMAR '12; SUH-RAMCHANRDAN '11)

Existence proofs of codes for any k: (CADAMBE ET AL. '11, '12; PAPAILIOPOULOS ET AL. '13; TAMO-WANG-BRUCK '13; GOPARAJU-FAZELI-VARDY '16)

Any parameters including $k > (n+1)/2$

- (n, k) MDS codes with optimal repair and $l = r^n$, $d = n - 1$;
- (n, k) universal MDS codes with d-optimal repair for any $k \leqslant d \leqslant n - 1$, $l = (d + 1 - k)^n$ over F, $|F| \geqslant (d + 1 - k)n$;

Constructions of Vector (Array) Codes

Low-rate regime $k \leqslant (n+1)/2$

Single-node repair: Product-matrix and other constructions of codes with d-optimal repair property (Rashmi-Shah-Kumar '11; Rashmi-Shah-Kumar '12; Suh-Ramchanrdan '11)

Existence proofs of codes for any k: (Cadambe et al. '11, '12; Papailiopoulos et al. '13; Tamo-Wang-Bruck '13; Goparaju-Fazeli-Vardy '16)

Any parameters including $k > (n+1)/2$

- (n,k) MDS codes with optimal repair and $l = r^n$, $d = n-1$;
- (n,k) universal MDS codes with d-optimal repair for any $k \leqslant d \leqslant n-1$, $l = (d+1-k)^n$ over F, $|F| \geqslant (d+1-k)n$;
- (n,k) universal MDS codes with (h,d)-optimal repair for any $h \leqslant r$, $k \leqslant d \leqslant n-h$, $l = s^n$, $s = \text{lcm}(1,2,\dots,r)$ over F, $|F| \geqslant sn$

(Min Ye and A.B., T-IT, no.4, 2017)

General encoding method [YE-B., 2017]

The code is formed of $l \times n$ matrices over F, each encoding kl data symbols.

General encoding method [Ye-B., 2017]

The code is formed of $l \times n$ matrices over F, each encoding kl data symbols.

- Parity-check equations:

$$\mathcal{C} = \{(C_1, C_2, \ldots, C_n) : \sum_{i=1}^{n} A_{t,i} C_i = 0, t = 1, \ldots, r\}$$

General encoding method [Ye-B., 2017]

The code is formed of $l \times n$ matrices over F, each encoding kl data symbols.

- Parity-check equations:

$$\mathcal{C} = \{(C_1, C_2, \dots, C_n) : \sum_{i=1}^{n} A_{t,i} C_i = 0, t = 1, \dots, r\}$$

- $r \times n$ parity check matrix

$$\begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & A_{2,3} & \dots & A_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_{r,1} & A_{r,2} & A_{r,3} & \dots & A_{r,n} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_n \end{bmatrix} = 0$$

where each A_{ij} is an $l \times l$ matrix and C_i is an l-vector over F

General encoding method [Ye-B., 2017]

The code is formed of $l \times n$ matrices over F, each encoding kl data symbols.

- Parity-check equations:

$$\mathcal{C} = \{(C_1, C_2, \dots, C_n) : \sum_{i=1}^{n} A_{t,i} C_i = 0, t = 1, \dots, r\}$$

- Choose $(A_{t,i})$ above to follow block Vandermonde structure

$$\begin{bmatrix} I & I & I & \dots & I \\ A_1 & A_2 & A_3 & \dots & A_n \\ A_1^2 & A_2^2 & A_3^2 & \dots & A_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_1^{r-1} & A_2^{r-1} & A_3^{r-1} & \dots & A_n^{r-1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_n \end{bmatrix} = 0$$

General encoding method [YE-B., 2017]

The code is formed of $l \times n$ matrices over F, each encoding kl data symbols.

- Parity-check equations:

$$\mathcal{C} = \{(C_1, C_2, \dots, C_n) : \sum_{i=1}^{n} A_{t,i} C_i = 0, t = 1, \dots, r\}$$

- Choose $(A_{t,i})$ above to follow block Vandermonde structure

$$\begin{bmatrix} I & I & I & \dots & I \\ A_1 & A_2 & A_3 & \dots & A_n \\ A_1^2 & A_2^2 & A_3^2 & \dots & A_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_1^{r-1} & A_2^{r-1} & A_3^{r-1} & \dots & A_n^{r-1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_n \end{bmatrix} = 0$$

- Commuting: $A_i A_j = A_j A_i$

General encoding method [YE-B., 2017]

The code is formed of $l \times n$ matrices over F, each encoding kl data symbols.

- Parity-check equations:

$$\mathcal{C} = \{(C_1, C_2, \dots, C_n) : \sum_{i=1}^{n} A_{t,i} C_i = 0, t = 1, \dots, r\}$$

- Choose $(A_{t,i})$ above to follow block Vandermonde structure

$$\begin{bmatrix} I & I & I & \dots & I \\ A_1 & A_2 & A_3 & \dots & A_n \\ A_1^2 & A_2^2 & A_3^2 & \dots & A_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_1^{r-1} & A_2^{r-1} & A_3^{r-1} & \dots & A_n^{r-1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_n \end{bmatrix} = 0$$

- Commuting: $A_i A_j = A_j A_i$

- $A_i - A_j$ invertible

General encoding method [YE-B., 2017]

The code is formed of $l \times n$ matrices over F, each encoding kl data symbols.

- Parity-check equations:

$$\mathcal{C} = \{(C_1, C_2, \dots, C_n) : \sum_{i=1}^{n} A_{t,i} C_i = 0, t = 1, \dots, r\}$$

- Choose $(A_{t,i})$ above to follow block Vandermonde structure

$$\begin{bmatrix} I & I & I & \dots & I \\ A_1 & A_2 & A_3 & \dots & A_n \\ A_1^2 & A_2^2 & A_3^2 & \dots & A_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_1^{r-1} & A_2^{r-1} & A_3^{r-1} & \dots & A_n^{r-1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_n \end{bmatrix} = 0$$

- Commuting: $A_i A_j = A_j A_i$
- $A_i - A_j$ invertible
- Natural choice: Diagonal matrices

Optimal $(1, n-1)$ repair MDS codes

- Take $l = r^n$; take the $l \times l$ matrix $A_i, i = 1, \ldots, n$ in the form

$$A_i = \sum_{a=0}^{l-1} \lambda_{i,a_i} e_a e_a^T$$

where λ_{ij} are distinct elements of F and $(a_n, a_{n-1}, \ldots, a_1)$ is r-ary expansion of a

Optimal $(1, n-1)$ repair MDS codes

- Take $l = r^n$; take the $l \times l$ matrix $A_i, i = 1, \dots, n$ in the form

$$A_i = \sum_{a=0}^{l-1} \lambda_{i,a_i} e_a e_a^T$$

where λ_{ij} are distinct elements of F and $(a_n, a_{n-1}, \dots, a_1)$ is r-ary expansion of a

- The codeword has the form $C = (C_1, \dots, C_n)$, where $C_i = (c_{i,0}, c_{i,1}, \dots, c_{i,l-1})^T$

$$\begin{matrix} c_{1,0} & c_{2,0} & \dots & c_{n,0} \\ c_{1,1} & c_{2,1} & \dots & c_{n,1} \\ \vdots & \vdots & \vdots & \vdots \\ c_{1,l-1} & c_{2,l-1} & \dots & c_{n,l-1} \end{matrix}$$

- Idea: Every row forms an RS code with different evaluation points $\{P_{i,j}\}$
  For $a = 0, 1, \dots, l-1$, write r-ary expansion $a = (a_1, a_2, \dots, a_n)$
  Evaluation points for a-th row: $(\lambda_{1,a_1}, \lambda_{2,a_2}, \dots, \lambda_{n,a_n})$

$(1, n-1)$-optimal repair property

Idea (cont'd): Let $C_i = (c_{i,a}, a = 0, 1, \dots, l-1)$ be the missing node.

Repair the contents by *groups of size* r that differ only in position i of the label

$(1, n-1)$-optimal repair property

Idea (cont'd): Let $C_i = (c_{i,a}, a = 0, 1, \ldots, l-1)$ be the missing node.
Repair the contents by *groups of size* r that differ only in position i of the label

- $a(i,u) = (a_1, a_2, \ldots, a_{i-1}, u, a_{i+1}, a_{i+2}, \ldots, a_n), \quad 0 \leqslant u \leqslant r-1$

$(1, n-1)$-optimal repair property

Idea (cont'd): Let $C_i = (c_{i,a}, a = 0, 1, \ldots, l-1)$ be the missing node.
Repair the contents by *groups of size* r that differ only in position i of the label

- $a(i,u) = (a_1, a_2, \ldots, a_{i-1}, u, a_{i+1}, a_{i+2}, \ldots, a_n), \quad 0 \leqslant u \leqslant r-1$
- $$\lambda^t_{1,a_1} c_{1,a} + \lambda^t_{2,a_2} c_{2,a} + \cdots + \lambda^t_{n,a_n} c_{n,a} = 0, \qquad t = 0, 1, \ldots, r-1$$

$(1, n-1)$-optimal repair property

Idea (cont'd): Let $C_i = (c_{i,a}, a = 0, 1, \dots, l-1)$ be the missing node.
Repair the contents by *groups of size* r that differ only in position i of the label

- $a(i,u) = (a_1, a_2, \dots, a_{i-1}, u, a_{i+1}, a_{i+2}, \dots, a_n), \quad 0 \leqslant u \leqslant r-1$
- $$\lambda_{1,a_1}^t c_{1,a} + \lambda_{2,a_2}^t c_{2,a} + \cdots + \lambda_{n,a_n}^t c_{n,a} = 0, \qquad t = 0, 1, \dots, r-1$$
- $$\lambda_{i,a_i}^t c_{i,a} + \sum_{j \neq i} \lambda_{j,a_j}^t c_{j,a} = 0$$

$(1, n-1)$-optimal repair property

Idea (cont'd): Let $C_i = (c_{i,a}, a = 0, 1, \dots, l-1)$ be the missing node.
Repair the contents by *groups of size* r that differ only in position i of the label

- $a(i,u) = (a_1, a_2, \dots, a_{i-1}, u, a_{i+1}, a_{i+2}, \dots, a_n), \quad 0 \leqslant u \leqslant r-1$
- $$\lambda^t_{1,a_1} c_{1,a} + \lambda^t_{2,a_2} c_{2,a} + \dots + \lambda^t_{n,a_n} c_{n,a} = 0, \qquad t = 0, 1, \dots, r-1$$
- $$\lambda^t_{i,a_i} c_{i,a} + \sum_{j \neq i} \lambda^t_{j,a_j} c_{j,a} = 0$$
- $$\lambda^t_{i,u} c_{i,a(i,u)} + \sum_{j \neq i} \lambda^t_{j,a_j} c_{j,a(i,u)} = 0, \qquad u = 0, 1, \dots, r-1$$

$(1, n-1)$-optimal repair property

Idea (cont'd): Let $C_i = (c_{i,a}, a = 0, 1, \dots, l-1)$ be the missing node.
Repair the contents by *groups of size* r that differ only in position i of the label

- $a(i,u) = (a_1, a_2, \dots, a_{i-1}, u, a_{i+1}, a_{i+2}, \dots, a_n), \quad 0 \leqslant u \leqslant r-1$
- $$\lambda^t_{1,a_1} c_{1,a} + \lambda^t_{2,a_2} c_{2,a} + \cdots + \lambda^t_{n,a_n} c_{n,a} = 0, \qquad t = 0, 1, \dots, r-1$$
- $$\lambda^t_{i,a_i} c_{i,a} + \sum_{j \neq i} \lambda^t_{j,a_j} c_{j,a} = 0$$
- $$\lambda^t_{i,u} c_{i,a(i,u)} + \sum_{j \neq i} \lambda^t_{j,a_j} c_{j,a(i,u)} = 0, \qquad u = 0, 1, \dots, r-1$$
- $$\sum_{u=0}^{r-1} \lambda^t_{i,u} c_{i,a(i,u)} + \sum_{u=0}^{r-1} \sum_{j \neq i} \lambda^t_{j,a_j} c_{j,a(i,u)} = 0$$

$(1, n-1)$-optimal repair property

$$\sum_{u=0}^{r-1} \lambda_{i,u}^t c_{i,a(i,u)} + \sum_{j \neq i} \Big(\sum_{u=0}^{r-1} \lambda_{j,a_j}^t c_{j,a(i,u)} \Big) = 0, \qquad t = 0, 1, \dots, r-1$$

$(1, n-1)$-optimal repair property

$$\sum_{u=0}^{r-1} \lambda_{i,u}^t c_{i,a(i,u)} + \sum_{j \neq i} \Big(\sum_{u=0}^{r-1} \lambda_{j,a_j}^t c_{j,a(i,u)} \Big) = 0, \qquad t = 0, 1, \ldots, r-1$$

$$\underbrace{\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \lambda_{i,0} & \lambda_{i,1} & \ldots & \lambda_{i,r-1} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{i,0}^{r-1} & \lambda_{i,1}^{r-1} & \ldots & \lambda_{i,r-1}^{r-1} \end{bmatrix}}_{\text{Vandermonde, rank } = r} \underbrace{\begin{bmatrix} c_{i,a(i,0)} \\ c_{i,a(i,1)} \\ \vdots \\ c_{i,a(i,r-1)} \end{bmatrix}}_{\text{missing information}}$$

$\sum_{u=0}^{r-1} c_{j,a(i,u)}$ →

$$+ \sum_{j \neq i} \boxed{ \underbrace{\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \lambda_{j,a_j} & \lambda_{j,a_j} & \ldots & \lambda_{j,a_j} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{j,a_j}^{r-1} & \lambda_{j,a_j}^{r-1} & \ldots & \lambda_{j,a_j}^{r-1} \end{bmatrix}}_{\text{aligned, rank } = 1} \begin{bmatrix} c_{j,a(i,0)} \\ c_{j,a(i,1)} \\ \vdots \\ c_{j,a(i,r-1)} \end{bmatrix} } = 0$$

$(1, n-1)$-optimal repair property

- For $u = 0, 1, \ldots, r-1$ let $a(i,u) := (a_n, \ldots, a_{i+1}, u, a_{i-1}, \ldots, a_1)$.
- Partition the symbols on the failed node i into r^{l-1} groups of size r each:

$$\{c_{i,a(i,0)}, c_{i,a(i,1)}, \ldots, c_{i,a(i,r-1)}\}$$

  for some $a \in \{0, 1, \ldots, l-1\}$

- Say C_i is unavailable. The elements in each group can be found by acquiring one element $\sum_{u=0}^{r-1} c_{j,a(i,u)}$ from each of the $n-1$ remaining nodes.
- Total repair bandwidth $= (n-1) \times 1 \times r^{l-1} = (n-1)(l/r)$, matching the lower bound

Repair of several erasures

Centralized and distributed (cooperative) models

Suppose that nodes i and j are erased.

Repair of several erasures

Centralized and distributed (cooperative) models

Suppose that nodes i and j are erased.

Centralized repair: Download information from the set of helper nodes $\mathcal{R}, |\mathcal{R}| = d$ that is used for repair of both C_i and C_j

Repair of several erasures

Centralized and distributed (cooperative) models

Suppose that nodes i and j are erased.

Centralized repair: Download information from the set of helper nodes $\mathcal{R}, |\mathcal{R}| = d$ that is used for repair of both C_i and C_j

Cooperative repair[1]:

- Round 1: Nodes C_i and C_j download (potentially, different) information from $\mathcal{R}$
- Round 2: Information exchange: $C_i \leftrightarrows C_j$

Both rounds of communication contribute to the repair bandwidth.

[1] Originally defined for $T \geqslant 2$ communication rounds (SHUM-HU, T-IT '13); YE-B, 2017 shows that 2 rounds suffice)

Cut-set bound

$$\beta \geqslant \frac{l}{d+1-k} d \qquad \text{(Dimakis et al., 2010)}$$

Cut-set bound

$$\beta \geqslant \frac{l}{d+1-k}d \qquad \text{(Dimakis et al., 2010)}$$

The code meeting this bound with equality is said to afford optimal repair

For $d = n-1$, $r = n-k$

$$\beta \geqslant \frac{l}{r}(n-1)$$

Cut-set bound

$$\beta \geqslant \frac{l}{d+1-k}d \qquad \text{(Dimakis et al., 2010)}$$

The code meeting this bound with equality is said to afford optimal repair

For $d = n-1$, $r = n-k$

$$\beta \geqslant \frac{l}{r}(n-1)$$

The cut-set bound extends to repair of $h \geqslant 1$ erasures (failed nodes):

- Centralized model: $\beta \geqslant \dfrac{hdl}{d+h-k}$ (V. Cadambe et al., '13)
- Cooperative model: $\beta \geqslant \dfrac{h(d+h-1)l}{d+h-k}$ (K. Shum and Y. Hu, '13)

Cut-set bound

$$\beta \geqslant \frac{l}{d+1-k} d \qquad \text{(Dimakis et al., 2010)}$$

The code meeting this bound with equality is said to afford optimal repair

For $d = n-1$, $r = n-k$

$$\beta \geqslant \frac{l}{r}(n-1)$$

The cut-set bound extends to repair of $h \geqslant 1$ erasures (failed nodes):

- Centralized model: $\beta \geqslant \dfrac{hdl}{d+h-k}$ (V. Cadambe et al., '13)
- Cooperative model: $\beta \geqslant \dfrac{h(d+h-1)l}{d+h-k}$ (K. Shum and Y. Hu, '13)

Codes that meet these bounds with equality are said to have (h,d)-optimal repair bandwidth

Universality and Error tolerance under Centralized repair

- Varying number of helpers: Codes that meet the cutset bound universally for $d_1, d_2, \ldots$
- Error tolerance: It is possible to repair a single node from $d+2t$ helper nodes, any t of which provide incorrect information

$$\beta \geqslant \frac{h(d+2t)l}{h+d-k}$$

S. Pawar et al.,*Distributed storage systems with adversarial attacks*, T-IT 2011
K.V. Rashmi et al., *Regenerating codes for errors and erasures*, T-IT 2012

- Universally error resilient MSR codes: Combination of the above features

M. Ye and A.B., *Explicit constructions of high-rate MDS array codes with optimal repair bandwidth*, T-IT 2017

Universality and Error tolerance under Centralized repair

- Varying number of helpers: Codes that meet the cutset bound universally for $d_1, d_2, \ldots$
- Error tolerance: It is possible to repair a single node from $d + 2t$ helper nodes, any t of which provide incorrect information

$$\beta \geqslant \frac{h(d+2t)l}{h+d-k}$$

S. PAWAR ET AL.,*Distributed storage systems with adversarial attacks*, T-IT 2011
K.V. RASHMI ET AL., *Regenerating codes for errors and erasures*, T-IT 2012

- Universally error resilient MSR codes: Combination of the above features

M. YE AND A.B., *Explicit constructions of high-rate MDS array codes with optimal repair bandwidth*, T-IT 2017

Secure distributed storage systems

- S. PAWAR ET AL., *On secure distributed data storage*, ISIT 2010
- V.A. RAMESHWAR AND N. KASHYAP, *Achieving secrecy capacity of MSR codes for all parameters*, 2019

Node size (subpacketization)

- The construction presented above needs $l = r^n$
- Lower bounds for linear repair schemes of MSR codes:

$$l \geqslant \exp(\sqrt{k/(2r-1)}) \qquad \text{(S. Goparaju, I. Tamo, and R. Calderbank, T-IT, 2014)}$$

$$l \geqslant \exp\left(\frac{k}{2}\ln\frac{2r}{r-1}\right) \qquad \text{(O. Alrabiah and V. Guruswami, 2019, arXiv)}$$

- There is a gap between the best known constructions and the bounds

Repair by transfer and Subpacketization (node size) bounds

(Optimal Access)

- Download what you read:
  Let $\mathcal{C}$ be an (n, k, l) MSR code with repair degree d. Suppose that each of the helper nodes provides $l/(d-k+1)$ symbols (i.e., $\mathcal{C}$ has the optimal repair property), and these are exactly the symbols accessed on the helper nodes

Repair by transfer and Subpacketization (node size) bounds

(Optimal Access)

- Download what you read:
  Let $\mathcal{C}$ be an (n, k, l) MSR code with repair degree d. Suppose that each of the helper nodes provides $l/(d-k+1)$ symbols (i.e., $\mathcal{C}$ has the optimal repair property), and these are exactly the symbols accessed on the helper nodes
- Constructions with $l = r^{n/r}$ (YE-B., '16; SASIDHARAN-VAJHA-KUMAR '16)

  Combine layers of independent MDS codes by extending parity checks across layers

  Coupled-layer perspective

  (SVK, '16 and M. VAJHA ET AL., *Clay codes: Moulding MDS codes to yield an MSR code*, USENIX FAST, 2018)

  J. LI, X. TANG AND C. TIAN, *A generic transformation to enable optimal repair in MDS codes*, T-IT 2018

Repair by transfer and Subpacketization (node size) bounds

(Optimal Access)

- Download what you read:
  Let $\mathcal{C}$ be an (n, k, l) MSR code with repair degree d. Suppose that each of the helper nodes provides $l/(d-k+1)$ symbols (i.e., $\mathcal{C}$ has the optimal repair property), and these are exactly the symbols accessed on the helper nodes
- Constructions with $l = r^{n/r}$ (YE-B., '16; SASIDHARAN-VAJHA-KUMAR '16)

  Combine layers of independent MDS codes by extending parity checks across layers

  Coupled-layer perspective

  (SVK, '16 and M. VAJHA ET AL., *Clay codes: Moulding MDS codes to yield an MSR code*, USENIX FAST, 2018)

  J. LI, X. TANG AND C. TIAN, *A generic transformation to enable optimal repair in MDS codes*, T-IT 2018
- By a result of BALAJI-KUMAR '17, the node size $l = r^{n/r}$ is optimal *under linear repair schemes* (if $r \nmid d = n-1$)

ϵ-MSR codes

- Relax the optimal repair condition
  There are constructions of codes that are ϵ-close to the cut-set bound with $l = O(\log n)$ (RAWAT-TAMO-GURUSWAMI-EFREMENKO, '17).

Cooperative repair

Cut-set bound for cooperative repair:

$$\beta \geqslant \frac{h(d+h-1)l}{d+h-k} = h\Big(\frac{dl}{h+d-k} + \frac{(h-1)l}{h+d-k}\Big)$$

Cooperative repair

Cut-set bound for cooperative repair:

$$\beta \geqslant \frac{h(d+h-1)l}{d+h-k} = h\Big(\frac{dl}{h+d-k} + \frac{(h-1)l}{h+d-k}\Big)$$

Structure of optimal codes:

- Each failed node downloads $\frac{l}{h+d-k}$ from the helper nodes
- Each failed node downloads $\frac{l}{h+d-k}$ from each of the other nodes in $\mathcal{F}$

Cooperative repair model is stronger than the centralized model

Theorem

Let $\mathcal{C}$ *be an* (n,k,l) *MDS array code and let* $\mathcal{F},\mathcal{R}\subseteq[n]$ *be two disjoint subsets such that* $|\mathcal{F}|\leqslant r$ *and* $|\mathcal{R}|\geqslant k$. *If*

$$\beta_{\text{coop}}(\mathcal{C})=\frac{|\mathcal{F}|(|\mathcal{R}|+|\mathcal{F}|-1)l}{|\mathcal{F}|+|\mathcal{R}|-k},$$

then

$$\beta_{\text{cent}}(\mathcal{C})=\frac{|\mathcal{F}||\mathcal{R}|l}{|\mathcal{F}|+|\mathcal{R}|-k}.$$

General results

- There is an explicit family of (n, k, l) MDS array codes that can optimally repair any h nodes from any d helper nodes, where $d \geqslant k+1, 2 \leqslant h \leqslant n-d$. The codes can be constructed over any field $F, |F| \geqslant (d+1-k)n$.

(Min Ye and A.B., *Cooperative repair*, T-IT, 2019)

Cooperative repair of two nodes

- Assume that nodes C_1, C_2 are erased.

Cooperative repair of two nodes

- Assume that nodes C_1, C_2 are erased.
- We construct an $(n, k, 3)$ MDS array code, where $k < n \leqslant |F| - 2$.

Cooperative repair of two nodes

- Assume that nodes C_1, C_2 are erased.
- We construct an $(n, k, 3)$ MDS array code, where $k < n \leqslant |F| - 2$.
- Let $\lambda_{1,0}, \lambda_{1,1}, \lambda_{2,0}, \lambda_{2,1}, \lambda_3, \lambda_4, \ldots, \lambda_n \in F$

Cooperative repair of two nodes

- Assume that nodes C_1, C_2 are erased.
- We construct an $(n, k, 3)$ MDS array code, where $k < n \leqslant |F| - 2$.
- Let $\lambda_{1,0}, \lambda_{1,1}, \lambda_{2,0}, \lambda_{2,1}, \lambda_3, \lambda_4, \dots, \lambda_n \in F$

- Parity-check equations:

$$\lambda_{1,0}^t c_{1,0} + \lambda_{2,0}^t c_{2,0} + \sum_{i=3}^{n} \lambda_i^t c_{i,0} = 0$$

$$\lambda_{1,1}^t c_{1,1} + \lambda_{2,0}^t c_{2,1} + \sum_{i=3}^{n} \lambda_i^t c_{i,1} = 0$$

$$\lambda_{1,0}^t c_{1,2} + \lambda_{2,1}^t c_{2,2} + \sum_{i=3}^{n} \lambda_i^t c_{i,2} = 0, \quad t = 0, 1, \dots, r-1$$

Idea of the construction

Lemma

For $i = 1, \dots, n$ *let* $\mu_{i,1} := c_{i,0} + c_{i,1},\ \mu_{i,2} := c_{i,0} + c_{i,2}$. *For any set of helper nodes* $\mathcal{R} \subseteq \{3, 4, \dots, n\}, |\mathcal{R}| = k + 1$, *the values*

$$c_{1,0}, c_{1,1}, \text{ and } \mu_{2,1} = c_{2,0} + c_{2,1}$$

are uniquely determined by $\{\mu_{i,1} : i \in \mathcal{R}\}$. *Similarly, the values of* $c_{2,0}, c_{2,2}$, *and* $\mu_{1,2}$ *are uniquely determined by* $\{\mu_{i,2} : i \in \mathcal{R}\}$.

Idea of the construction

Lemma

For $i = 1, \dots, n$ let $\mu_{i,1} := c_{i,0} + c_{i,1},\ \mu_{i,2} := c_{i,0} + c_{i,2}$. For any set of helper nodes $\mathcal{R} \subseteq \{3, 4, \dots, n\}, |\mathcal{R}| = k + 1$, the values

$$c_{1,0}, c_{1,1}, \text{ and } \mu_{2,1} = c_{2,0} + c_{2,1}$$

are uniquely determined by $\{\mu_{i,1} : i \in \mathcal{R}\}$. Similarly, the values of $c_{2,0}, c_{2,2}$, and $\mu_{1,2}$ are uniquely determined by $\{\mu_{i,2} : i \in \mathcal{R}\}$.

$$\lambda_{1,0}^t c_{1,0} + \lambda_{1,1}^t c_{1,1} + \lambda_{2,0}^t \mu_{2,1} + \sum_{i=3}^{n} \lambda_i^t \mu_{i,1} = 0, t = 0, 1, \dots, r-1$$

Idea of the construction

Lemma

For $i = 1, \ldots, n$ let $\mu_{i,1} := c_{i,0} + c_{i,1},\ \mu_{i,2} := c_{i,0} + c_{i,2}$. For any set of helper nodes $\mathcal{R} \subseteq \{3, 4, \ldots, n\}, |\mathcal{R}| = k + 1$, the values

$$c_{1,0}, c_{1,1}, \text{ and } \mu_{2,1} = c_{2,0} + c_{2,1}$$

are uniquely determined by $\{\mu_{i,1} : i \in \mathcal{R}\}$. Similarly, the values of $c_{2,0}, c_{2,2}$, and $\mu_{1,2}$ are uniquely determined by $\{\mu_{i,2} : i \in \mathcal{R}\}$.

In matrix form:

$$\begin{bmatrix} 1 & 1 \\ \lambda_{1,0} & \lambda_{1,1} \\ \lambda_{1,0}^2 & \lambda_{1,1}^2 \\ \vdots & \vdots \\ \lambda_{1,0}^{r-1} & \lambda_{1,1}^{r-1} \end{bmatrix} \begin{bmatrix} c_{1,0} \\ c_{1,1} \end{bmatrix} = - \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ \lambda_{2,0} & \lambda_3 & \lambda_4 & \ldots & \lambda_n \\ \lambda_{2,0}^2 & \lambda_3^2 & \lambda_4^2 & \ldots & \lambda_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \lambda_{2,0}^{r-1} & \lambda_3^{r-1} & \lambda_4^{r-1} & \ldots & \lambda_n^{r-1} \end{bmatrix} \begin{bmatrix} \mu_{2,1} \\ \mu_{3,1} \\ \mu_{4,1} \\ \vdots \\ \mu_{n,1} \end{bmatrix}.$$

Once we know $\mu_{j,1}, j = 2, 3, \ldots, n$ we also know $c_{1,0}, c_{1,1}$

Idea of the construction

Lemma

For $i = 1, \ldots, n$ *let* $\mu_{i,1} := c_{i,0} + c_{i,1},\ \mu_{i,2} := c_{i,0} + c_{i,2}$. *For any set of helper nodes* $\mathcal{R} \subseteq \{3, 4, \ldots, n\}, |\mathcal{R}| = k + 1$, *the values*

$$c_{1,0}, c_{1,1}, \text{ and } \mu_{2,1} = c_{2,0} + c_{2,1}$$

are uniquely determined by $\{\mu_{i,1} : i \in \mathcal{R}\}$. *Similarly, the values of* $c_{2,0}, c_{2,2}$, *and* $\mu_{1,2}$ *are uniquely determined by* $\{\mu_{i,2} : i \in \mathcal{R}\}$.

Let $p_0(x) = (x - \lambda_{1,0})(x - \lambda_{1,1}), p_i(x) = x^i p_0(x), i = 1, 2, \ldots, r - 3$

Idea of the construction

Lemma

For $i = 1, \dots, n$ *let* $\mu_{i,1} := c_{i,0} + c_{i,1},\ \mu_{i,2} := c_{i,0} + c_{i,2}$. *For any set of helper nodes* $\mathcal{R} \subseteq \{3, 4, \dots, n\}, |\mathcal{R}| = k + 1$, *the values*

$$c_{1,0}, c_{1,1}, \text{ and } \mu_{2,1} = c_{2,0} + c_{2,1}$$

are uniquely determined by $\{\mu_{i,1} : i \in \mathcal{R}\}$. *Similarly, the values of* $c_{2,0}, c_{2,2}$, *and* $\mu_{1,2}$ *are uniquely determined by* $\{\mu_{i,2} : i \in \mathcal{R}\}$.

$$P := \begin{bmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,r-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,r-1} \\ \vdots & \vdots & \vdots & \vdots \\ p_{r-3,0} & p_{r-3,1} & \cdots & p_{r-3,r-1} \end{bmatrix}.$$

Idea of the construction

Lemma

For $i = 1, \dots, n$ *let* $\mu_{i,1} := c_{i,0} + c_{i,1},\ \mu_{i,2} := c_{i,0} + c_{i,2}$. *For any set of helper nodes* $\mathcal{R} \subseteq \{3, 4, \dots, n\}, |\mathcal{R}| = k + 1$, *the values*

$$c_{1,0}, c_{1,1},\ \text{and } \mu_{2,1} = c_{2,0} + c_{2,1}$$

are uniquely determined by $\{\mu_{i,1} : i \in \mathcal{R}\}$. *Similarly, the values of* $c_{2,0}, c_{2,2}$, *and* $\mu_{1,2}$ *are uniquely determined by* $\{\mu_{i,2} : i \in \mathcal{R}\}$.

$$P \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ \lambda_{2,0} & \lambda_3 & \lambda_4 & \dots & \lambda_n \\ \lambda_{2,0}^2 & \lambda_3^2 & \lambda_4^2 & \dots & \lambda_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \lambda_{2,0}^{r-1} & \lambda_3^{r-1} & \lambda_4^{r-1} & \dots & \lambda_n^{r-1} \end{bmatrix} = \begin{bmatrix} p_0(\lambda_{2,0}) & p_0(\lambda_3) & p_0(\lambda_4) & \dots & p_0(\lambda_n) \\ p_0(\lambda_{2,0})\lambda_{2,0} & p_0(\lambda_3)\lambda_3 & p_0(\lambda_4)\lambda_4 & \dots & p_0(\lambda_n)\lambda_n \\ p_0(\lambda_{2,0})\lambda_{2,0}^2 & p_0(\lambda_3)\lambda_3^2 & p_0(\lambda_4)\lambda_4^2 & \dots & p_0(\lambda_n)\lambda_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_0(\lambda_{2,0})\lambda_{2,0}^{r-3} & p_0(\lambda_3)\lambda_3^{r-3} & p_0(\lambda_4)\lambda_4^{r-3} & \dots & p_0(\lambda_n)\lambda_n^{r-3} \end{bmatrix}$$

Idea of the construction

Lemma

For $i = 1, \dots, n$ *let* $\mu_{i,1} := c_{i,0} + c_{i,1},\ \mu_{i,2} := c_{i,0} + c_{i,2}$. *For any set of helper nodes* $\mathcal{R} \subseteq \{3, 4, \dots, n\}, |\mathcal{R}| = k + 1$, *the values*

$$c_{1,0}, c_{1,1}, \text{ and } \mu_{2,1} = c_{2,0} + c_{2,1}$$

are uniquely determined by $\{\mu_{i,1} : i \in \mathcal{R}\}$. *Similarly, the values of* $c_{2,0}, c_{2,2}$, *and* $\mu_{1,2}$ *are uniquely determined by* $\{\mu_{i,2} : i \in \mathcal{R}\}$.

$$\begin{bmatrix} p_0(\lambda_{2,0}) & p_0(\lambda_3) & p_0(\lambda_4) & \dots & p_0(\lambda_n) \\ p_0(\lambda_{2,0})\lambda_{2,0} & p_0(\lambda_3)\lambda_3 & p_0(\lambda_4)\lambda_4 & \dots & p_0(\lambda_n)\lambda_n \\ p_0(\lambda_{2,0})\lambda_{2,0}^2 & p_0(\lambda_3)\lambda_3^2 & p_0(\lambda_4)\lambda_4^2 & \dots & p_0(\lambda_n)\lambda_n^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_0(\lambda_{2,0})\lambda_{2,0}^{r-3} & p_0(\lambda_3)\lambda_3^{r-3} & p_0(\lambda_4)\lambda_4^{r-3} & \dots & p_0(\lambda_n)\lambda_n^{r-3} \end{bmatrix} \begin{bmatrix} \mu_{2,1} \\ \mu_{3,1} \\ \mu_{4,1} \\ \vdots \\ \mu_{n,1} \end{bmatrix} = 0$$

The vector $(\mu_{2,1}, \mu_{3,1}, \dots, \mu_{n,1})$ forms a codeword in an $(n-1, k+1)$ (G)RS code

Parameters of the constructions

	Repairing the first h nodes		Repairing any h nodes	
Values of $h = \|\mathcal{F}\|, d = \|\mathcal{R}\|$	$\|F\|$	l	$\|F\|$	l
$h = 2, d = k+1$	$n+2$	3	$2n$	$3^{\binom{n}{2}}$
$h = 2$, any d	$n+2(s-1)$	s^2-1	sn	$(s^2-1)^{\binom{n}{2}}$
any h, $d = k+1$	$n+h$	$h+1$	$2n$	$(h+1)^{\binom{n}{h}}$
any h, any d	$n+h(s-1)$	$(h+d-k)(s-1)^{h-1}$	sn	$((h+d-k)(s-1)^{h-1})^{\binom{n}{h}}$

Repair of Reed-Solomon codes

Problem introduced by K. Shanmugam et al., 2014. It was developed by V. Guruswami and M. Wootters (T-IT, Sept. 2017):

Repair of Reed-Solomon codes

Problem introduced by K. Shanmugam et al., 2014. It was developed by V. Guruswami and M. Wootters (T-IT, Sept. 2017):

- Characterized repair schemes of RS codes
- Analyzed full-length RS codes for single-node repair

Repair of Reed-Solomon codes

Problem introduced by K. Shanmugam et al., 2014. It was developed by V. Guruswami and M. Wootters (T-IT, Sept. 2017):

- Characterized repair schemes of RS codes
- Analyzed full-length RS codes for single-node repair

Min Ye and A.B., RS codes with asymptotically optimal repair bandwidth, ISIT'16

H. Dau and O. Milenkovic, Optimal repair schemes of some families of full-length RS codes, ISIT'17

A. Chowdhuri and A. Vardy, Schemes for asymptotically optimal repair of MDS codes, 2017

Repair of Reed-Solomon codes

Problem introduced by K. Shanmugam et al., 2014. It was developed by V. Guruswami and M. Wootters (T-IT, Sept. 2017):

- Characterized repair schemes of RS codes
- Analyzed full-length RS codes for single-node repair

Min Ye and A.B., RS codes with asymptotically optimal repair bandwidth, ISIT'16

H. Dau and O. Milenkovic, Optimal repair schemes of some families of full-length RS codes, ISIT'17

A. Chowdhuri and A. Vardy, Schemes for asymptotically optimal repair of MDS codes, 2017

Optimal-repair (shortened) RS codes (work with I. Tamo and Min Ye '17):

Repair of Reed-Solomon codes

Problem introduced by K. Shanmugam et al., 2014. It was developed by V. Guruswami and M. Wootters (T-IT, Sept. 2017):

- Characterized repair schemes of RS codes
- Analyzed full-length RS codes for single-node repair

Min Ye and A.B., RS codes with asymptotically optimal repair bandwidth, ISIT'16

H. Dau and O. Milenkovic, Optimal repair schemes of some families of full-length RS codes, ISIT'17

A. Chowdhuri and A. Vardy, Schemes for asymptotically optimal repair of MDS codes, 2017

Optimal-repair (shortened) RS codes (work with I. Tamo and Min Ye '17):

- Construction of RS codes for single-node repair with optimal repair bandwidth

Repair of Reed-Solomon codes

Problem introduced by K. Shanmugam et al., 2014. It was developed by V. Guruswami and M. Wootters (T-IT, Sept. 2017):

- Characterized repair schemes of RS codes
- Analyzed full-length RS codes for single-node repair

Min Ye and A.B., RS codes with asymptotically optimal repair bandwidth, ISIT'16

H. Dau and O. Milenkovic, Optimal repair schemes of some families of full-length RS codes, ISIT'17

A. Chowdhuri and A. Vardy, Schemes for asymptotically optimal repair of MDS codes, 2017

Optimal-repair (shortened) RS codes (work with I. Tamo and Min Ye '17):

- Construction of RS codes for single-node repair with optimal repair bandwidth
- Lower bound on sub-packetization parameter l

Repair of Reed-Solomon codes

Problem introduced by K. SHANMUGAM ET AL., 2014. It was developed by V. GURUSWAMI AND M. WOOTTERS (T-IT, Sept. 2017):

- Characterized repair schemes of RS codes
- Analyzed full-length RS codes for single-node repair

MIN YE AND A.B., RS codes with asymptotically optimal repair bandwidth, ISIT'16

H. DAU AND O. MILENKOVIC, Optimal repair schemes of some families of full-length RS codes, ISIT'17

A. CHOWDHURI AND A. VARDY, Schemes for asymptotically optimal repair of MDS codes, 2017

Optimal-repair (shortened) RS codes (work with I. TAMO AND MIN YE '17):

- Construction of RS codes for single-node repair with optimal repair bandwidth
- Lower bound on sub-packetization parameter l
- Construction of RS codes that universally achieve the cut-set bound for any number of erasures

Repair bandwidth of Reed-Solomon codes

Idea: [Shanmugam-Papailiopoulos-Dimakis, '14] Consider the RS code $\mathcal{C}$ over F as a code over a subfield B ("vectorize" $\mathcal{C}$)

Repair bandwidth of Reed-Solomon codes

Idea: [Shanmugam-Papailiopoulos-Dimakis, '14] Consider the RS code $\mathcal{C}$ over F as a code over a subfield B ("vectorize" $\mathcal{C}$)

Example:

Repair bandwidth of Reed-Solomon codes

Idea: [Shanmugam-Papailiopoulos-Dimakis, '14] Consider the RS code $\mathcal{C}$ over F as a code over a subfield B ("vectorize" $\mathcal{C}$)

Example:

- Consider an RS code over $F = \mathbb{F}_{16}$ as an array code over $B = \mathbb{F}_2$, i.e., $l = 4$

Repair bandwidth of Reed-Solomon codes

Idea: [SHANMUGAM-PAPAILIOPOULOS-DIMAKIS, '14] Consider the RS code $\mathcal{C}$ over F as a code over a subfield B ("vectorize" $\mathcal{C}$)

Example:

- Consider an RS code over $F = \mathbb{F}_{16}$ as an array code over $B = \mathbb{F}_2$, i.e., $l = 4$
- F can be represented as a 4-dimensional vector space over $B = \{0, 1\}$

Repair bandwidth of Reed-Solomon codes

Idea: [SHANMUGAM-PAPAILIOPOULOS-DIMAKIS, '14] Consider the RS code $\mathcal{C}$ over F as a code over a subfield B ("vectorize" $\mathcal{C}$)

Example:

- Consider an RS code over $F = \mathbb{F}_{16}$ as an array code over $B = \mathbb{F}_2$, i.e., $l = 4$
- F can be represented as a 4-dimensional vector space over $B = \{0, 1\}$
- To "compress" the values of the helper nodes we project them on a subfield of F

Repair bandwidth of Reed-Solomon codes

Idea: [Shanmugam-Papailiopoulos-Dimakis, '14] Consider the RS code $\mathcal{C}$ over F as a code over a subfield B ("vectorize" $\mathcal{C}$)

Example:

- Consider an RS code over $F = \mathbb{F}_{16}$ as an array code over $B = \mathbb{F}_2$, i.e., $l = 4$
- F can be represented as a 4-dimensional vector space over $B = \{0, 1\}$
- To "compress" the values of the helper nodes we project them on a subfield of F
- Let $\alpha \in F$ be such that $\alpha^4 = \alpha + 1$, then $(1, \alpha, \alpha^2, \alpha^3)$ form a basis of F over B

Repair bandwidth of Reed-Solomon codes

Idea: [Shanmugam-Papailiopoulos-Dimakis, '14] Consider the RS code $\mathcal{C}$ over F as a code over a subfield B ("vectorize" $\mathcal{C}$)

Example:

- Consider an RS code over $F = \mathbb{F}_{16}$ as an array code over $B = \mathbb{F}_2$, i.e., $l = 4$
- F can be represented as a 4-dimensional vector space over $B = \{0, 1\}$
- To "compress" the values of the helper nodes we project them on a subfield of F
- Let $\alpha \in F$ be such that $\alpha^4 = \alpha + 1$, then $(1, \alpha, \alpha^2, \alpha^3)$ form a basis of F over B
- Trace $\mathrm{tr}(x) = x + x^2 + x^{2^2} + x^{2^3}$ is a map from F to B:

$$\mathrm{tr}(0) = 0, \mathrm{tr}(1) = 0, \mathrm{tr}(\alpha) = 1, \quad \text{etc.}$$

Repair bandwidth of Reed-Solomon codes

Idea: [Shanmugam-Papailiopoulos-Dimakis, '14] Consider the RS code $\mathcal{C}$ over F as a code over a subfield B ("vectorize" $\mathcal{C}$)

Example:

- Consider an RS code over $F = \mathbb{F}_{16}$ as an array code over $B = \mathbb{F}_2$, i.e., $l = 4$
- F can be represented as a 4-dimensional vector space over $B = \{0, 1\}$
- To "compress" the values of the helper nodes we project them on a subfield of F
- Let $\alpha \in F$ be such that $\alpha^4 = \alpha + 1$, then $(1, \alpha, \alpha^2, \alpha^3)$ form a basis of F over B
- Trace $\mathrm{tr}(x) = x + x^2 + x^{2^2} + x^{2^3}$ is a map from F to B:

$$\mathrm{tr}(0) = 0, \mathrm{tr}(1) = 0, \mathrm{tr}(\alpha) = 1, \quad \text{etc.}$$

- For any $c \in F$ the values $\mathrm{tr}(c), \mathrm{tr}(\alpha c), \mathrm{tr}(\alpha^2 c), \mathrm{tr}(\alpha^3 c)$ suffice to recover c

General repair scheme

The repair scheme of GURUSWAMI-WOOTTERS '16:

General repair scheme

The repair scheme of GURUSWAMI-WOOTTERS '16:

- Let $B \subset F$ be finite fields, $[F : B] = l$; $\Omega \subset F$; $|\Omega| = \{P_1, \ldots, P_n\}$
  Let $\mathcal{C} = RS_F(n, k, \Omega)$ be the RS code; $r = n - k$

General repair scheme

The repair scheme of GURUSWAMI-WOOTTERS '16:

- Let $B \subset F$ be finite fields, $[F : B] = l$; $\Omega \subset F$; $|\Omega| = \{P_1, \dots, P_n\}$
  Let $\mathcal{C} = RS_F(n, k, \Omega)$ be the RS code; $r = n - k$

- Let c_i be erased.

General repair scheme

The repair scheme of GURUSWAMI-WOOTTERS '16:

- Let $B \subset F$ be finite fields, $[F : B] = l$; $\Omega \subset F$; $|\Omega| = \{P_1, \dots, P_n\}$
  Let $\mathcal{C} = RS_F(n, k, \Omega)$ be the RS code; $r = n - k$

- Let c_i be erased.

- Let $b_1, b_2, \dots, b_l \in \mathcal{C}^\perp$ be such that $b_{1,i}, \dots, b_{l,i}$ form a basis of F over B. The values $\mathrm{tr}(b_{ji} c_i)$ suffice to recover c_i

General repair scheme

The repair scheme of GURUSWAMI-WOOTTERS '16:

- Let $B \subset F$ be finite fields, $[F : B] = l$; $\Omega \subset F$; $|\Omega| = \{P_1, \dots, P_n\}$
  Let $\mathcal{C} = RS_F(n, k, \Omega)$ be the RS code; $r = n - k$

- Let c_i be erased.

- Let $b_1, b_2, \dots, b_l \in \mathcal{C}^\perp$ be such that $b_{1,i}, \dots, b_{l,i}$ form a basis of F over B. The values $\mathrm{tr}(b_{ji}c_i)$ suffice to recover c_i

- We have $c_i b_{j,i} + \sum_{m \neq i}^{n} c_m b_{j,m} = 0, \ \ j = 1, \dots, l$

General repair scheme

The repair scheme of GURUSWAMI-WOOTTERS '16:

- Let $B \subset F$ be finite fields, $[F:B]=l$; $\Omega \subset F$; $|\Omega| = \{P_1, \dots, P_n\}$
  Let $\mathcal{C} = RS_F(n,k,\Omega)$ be the RS code; $r = n-k$

- Let c_i be erased.

- Let $b_1, b_2, \dots, b_l \in \mathcal{C}^\perp$ be such that $b_{1,i}, \dots, b_{l,i}$ form a basis of F over B. The values $\mathrm{tr}(b_{ji}c_i)$ suffice to recover c_i

- We have $c_i b_{j,i} + \sum_{m \neq i}^{n} c_m b_{j,m} = 0, \;\; j = 1, \dots, l$

- We have $\mathrm{tr}(b_{ji}c_i) = -\sum_{t\neq i} \mathrm{tr}(b_{jt}c_t), \;\; j = 1, \dots, l$

General repair scheme

The repair scheme of GURUSWAMI-WOOTTERS '16:

- Let $B \subset F$ be finite fields, $[F : B] = l$; $\Omega \subset F$; $|\Omega| = \{P_1, \dots, P_n\}$
  Let $\mathcal{C} = RS_F(n, k, \Omega)$ be the RS code; $r = n - k$
- Let c_i be erased.
- Let $b_1, b_2, \dots, b_l \in \mathcal{C}^\perp$ be such that $b_{1,i}, \dots, b_{l,i}$ form a basis of F over B. The values $\mathrm{tr}(b_{ji}c_i)$ suffice to recover c_i
- We have $c_i b_{j,i} + \sum_{m \neq i}^{n} c_m b_{j,m} = 0, \;\; j = 1, \dots, l$
- We have $\mathrm{tr}(b_{ji}c_i) = -\sum_{t \neq i} \mathrm{tr}(b_{jt}c_t), \;\; j = 1, \dots, l$
- We need $\{\mathrm{tr}(b_{jt}c_t), j = 1, \dots, l; t \neq i\}$

General repair scheme

The repair scheme of GURUSWAMI-WOOTTERS '16:

- Let $B \subset F$ be finite fields, $[F : B] = l$; $\Omega \subset F$; $|\Omega| = \{P_1, \dots, P_n\}$
  Let $\mathcal{C} = RS_F(n, k, \Omega)$ be the RS code; $r = n - k$
- Let c_i be erased.
- Let $b_1, b_2, \dots, b_l \in \mathcal{C}^\perp$ be such that $b_{1,i}, \dots, b_{l,i}$ form a basis of F over B. The values $\mathrm{tr}(b_{ji}c_i)$ suffice to recover c_i
- We have $c_i b_{j,i} + \sum_{m \neq i}^{n} c_m b_{j,m} = 0, \;\; j = 1, \dots, l$
- We have $\mathrm{tr}(b_{ji}c_i) = -\sum_{t \neq i} \mathrm{tr}(b_{jt}c_t), \;\; j = 1, \dots, l$
- We need $\{\mathrm{tr}(b_{jt}c_t), j = 1, \dots, l; t \neq i\}$
- Let B_t be a maximum-size linearly independent subset of $\{b_{jt}, j = 1 \dots l\}$
  We can find c_i from $\bigcup_{t \neq i} \{\mathrm{tr}(\beta c_t), \beta \in B_t\}$

General repair scheme

The repair scheme of GURUSWAMI-WOOTTERS '16:

- Let $B \subset F$ be finite fields, $[F : B] = l$; $\Omega \subset F$; $|\Omega| = \{P_1, \dots, P_n\}$
  Let $\mathcal{C} = RS_F(n, k, \Omega)$ be the RS code; $r = n - k$
- Let c_i be erased.
- Let $b_1, b_2, \dots, b_l \in \mathcal{C}^\perp$ be such that $b_{1,i}, \dots, b_{l,i}$ form a basis of F over B. The values $\mathrm{tr}(b_{ji}c_i)$ suffice to recover c_i
- We have $c_i b_{j,i} + \sum_{m \neq i}^{n} c_m b_{j,m} = 0, \quad j = 1, \dots, l$
- We have $\mathrm{tr}(b_{ji}c_i) = -\sum_{t \neq i} \mathrm{tr}(b_{jt}c_t), \quad j = 1, \dots, l$
- We need $\{\mathrm{tr}(b_{jt}c_t), j = 1, \dots, l; t \neq i\}$
- Let B_t be a maximum-size linearly independent subset of $\{b_{jt}, j = 1 \dots l\}$
  We can find c_i from $\bigcup_{t \neq i} \{\mathrm{tr}(\beta c_t), \beta \in B_t\}$

This is essentially the only possible linear repair scheme

Basics of RS repair

Basics of RS repair

Theorem: [GURUSWAMI-WOOTTERS, 2016]

Linear repair scheme for coordinate i with bandwidth b

$$\Updownarrow$$

1. there is a subset of codewords $P_i \subset \mathcal{C}^\perp, |P_i| = l$ such that $\dim_B(\{x_i : x \in P_i\}) = l$
2. $b \geqslant \sum_{j\in[n]\setminus\{i\}} \dim_B(\{x_j : x \in P_i\})$

Basics of RS repair

Theorem: [Guruswami-Wootters, 2016]

Linear repair scheme for coordinate i with bandwidth b

$$\Updownarrow$$

1. there is a subset of codewords $P_i \subset \mathcal{C}^\perp, |P_i| = l$ such that $\dim_B(\{x_i : x \in P_i\}) = l$
2. $b \geqslant \sum_{j \in [n] \setminus \{i\}} \dim_B(\{x_j : x \in P_i\})$

If l is small compared to $n - k$ (for instance, $n = |F|$), a lower bound on the repair bandwidth is

$$b \geqslant k + l - 1$$

Thus, for repair of full-length RS codes the cutset bound is not attainable.

RS codes for repair of a single node from d helper nodes

RS codes for repair of a single node from d helper nodes

- Let $\Omega = \{\alpha_1, \dots, \alpha_n\}$, where $\alpha_i, i = 1, \dots, n$ are algebraic elements over $\mathbb{F}_q$;

RS codes for repair of a single node from d helper nodes

- Let $\Omega = \{\alpha_1, \dots, \alpha_n\}$, where $\alpha_i, i = 1, \dots, n$ are algebraic elements over $\mathbb{F}_q$;
- $F_i := \mathbb{F}_q(\{\alpha_j, j \neq i\})$

RS codes for repair of a single node from d helper nodes

- Let $\Omega = \{\alpha_1, \ldots, \alpha_n\}$, where $\alpha_i, i = 1, \ldots, n$ are algebraic elements over $\mathbb{F}_q$;
- $F_i := \mathbb{F}_q(\{\alpha_j, j \neq i\})$
- $\mathbb{F} := \mathbb{F}_q(\alpha_1, \ldots, \alpha_n)$

RS codes for repair of a single node from d helper nodes

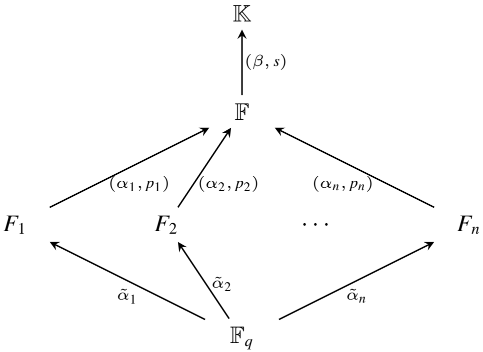
- Let $\Omega = \{\alpha_1, \dots, \alpha_n\}$, where $\alpha_i, i = 1, \dots, n$ are algebraic elements over $\mathbb{F}_q$;
- $F_i := \mathbb{F}_q(\{\alpha_j, j \neq i\})$
- $\mathbb{F} := \mathbb{F}_q(\alpha_1, \dots, \alpha_n)$
- $\mathbb{K} := \mathbb{F}(\beta)$, where $\deg_{\mathbb{F}}(\beta) = s := d + k - 1$

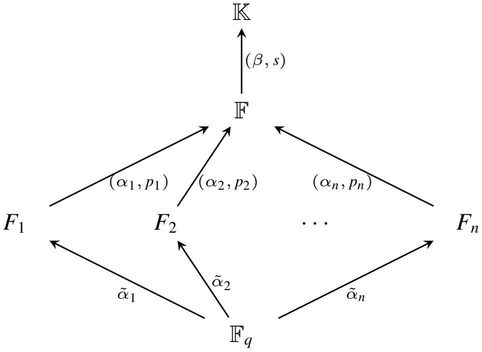
RS codes for repair of a single node from d helper nodes

- Let $\Omega = \{\alpha_1, \ldots, \alpha_n\}$, where $\alpha_i, i = 1, \ldots, n$ are algebraic elements over $\mathbb{F}_q$;
- $F_i := \mathbb{F}_q(\{\alpha_j, j \neq i\})$
- $\mathbb{F} := \mathbb{F}_q(\alpha_1, \ldots, \alpha_n)$
- $\mathbb{K} := \mathbb{F}(\beta)$, where $\deg_{\mathbb{F}}(\beta) = s := d + k - 1$
- $RS_{\mathbb{K}}(n, k, \{\alpha_1, \ldots, \alpha_n\})$

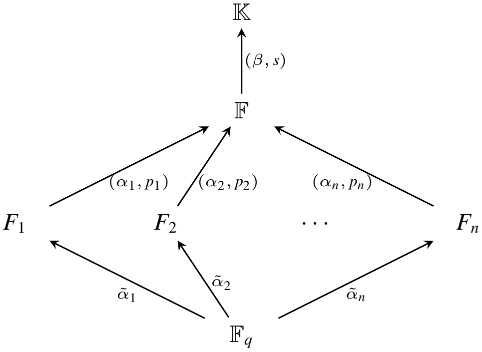
RS codes for repair of a single node from d helper nodes

- Let $\Omega = \{\alpha_1, \dots, \alpha_n\}$, where $\alpha_i, i = 1, \dots, n$ are algebraic elements over $\mathbb{F}_q$;
- $F_i := \mathbb{F}_q(\{\alpha_j, j \neq i\})$
- $\mathbb{F} := \mathbb{F}_q(\alpha_1, \dots, \alpha_n)$
- $\mathbb{K} := \mathbb{F}(\beta)$, where $\deg_{\mathbb{F}}(\beta) = s := d + k - 1$
- $RS_{\mathbb{K}}(n, k, \{\alpha_1, \dots, \alpha_n\})$
- Suppose that $\alpha_i \notin \mathbb{F}_q(\{\alpha_j, j \neq i\})$ and $\deg_{F_i}(\alpha_i) \equiv 1 \bmod s$

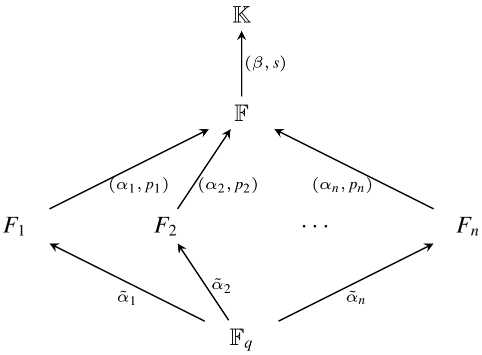




$\mathbb{K}$
(β, s)
$\mathbb{F}$
(α_1, p_1)
(α_2, p_2)
(α_n, p_n)
F_1
F_2
$\cdots$
F_n
$\tilde{\alpha}_1$
$\tilde{\alpha}_2$
$\tilde{\alpha}_n$
$\mathbb{F}_q$



Consider the RS code
$\mathcal{C} := RS_{\mathbb{K}}(n, k, \{\alpha_1, \ldots, \alpha_n\})$



Consider the RS code
$\mathcal{C} := RS_{\mathbb{K}}(n, k, \{\alpha_1, \ldots, \alpha_n\})$

Repair of the node i is performed over F_i

Subpacketization for linear repair of scalar codes

- Given n, we have

$$l := [\mathbb{K} : \mathbb{F}_q] = s \prod_{\substack{i=1 \\ p_i \equiv 1 \bmod s}}^{n} p_i$$

Subpacketization for linear repair of scalar codes

- Given n, we have

$$l := [\mathbb{K} : \mathbb{F}_q] = s \prod_{\substack{i=1 \\ p_i \equiv 1 \bmod s}}^{n} p_i$$

- Thus, $\mathcal{C} = \mathrm{RS}_{\mathbb{K}}(n, k, \Omega)$ where

$$q = p^l, l \approx \exp((1 + o(1))n \log n)$$

Subpacketization for linear repair of scalar codes

- Given n, we have

$$l := [\mathbb{K} : \mathbb{F}_q] = s \prod_{\substack{i=1 \\ p_i \equiv 1 \bmod s}}^{n} p_i$$

- Thus, $\mathcal{C} = \mathrm{RS}_{\mathbb{K}}(n, k, \Omega)$ where

$$q = p^l, l \approx \exp((1 + o(1))n \log n)$$

- Is l too large?

Subpacketization for linear repair of scalar codes

- Given n, we have

$$l := [\mathbb{K} : \mathbb{F}_q] = s \prod_{\substack{i=1 \\ p_i \equiv 1 \text{ mod } s}}^{n} p_i$$

- Thus, $\mathcal{C} = \text{RS}_{\mathbb{K}}(n, k, \Omega)$ where

$$q = p^l, l \approx \exp((1 + o(1))n \log n)$$

- Is l too large?
  In fact $l = \exp((1 + o(1))k \log k)$ is necessary!

Subpacketization for linear repair of scalar codes

Theorem

- *Let $B = \mathbb{F}_q$ and $F = \mathbb{F}_{q^l}$ for a prime power q.*

Subpacketization for linear repair of scalar codes

Theorem

- *Let* $B = \mathbb{F}_q$ *and* $F = \mathbb{F}_{q^l}$ *for a prime power* q*.*
- $k + 1 \leqslant d \leqslant n - 1$

Subpacketization for linear repair of scalar codes

Theorem

- *Let $B = \mathbb{F}_q$ and $F = \mathbb{F}_{q^l}$ for a prime power q.*
- $k + 1 \leqslant d \leqslant n - 1$
- $\mathcal{C} \subseteq F^n$ *an (n, k) scalar linear MDS code with a linear repair scheme over F*

Subpacketization for linear repair of scalar codes

Theorem

- *Let* $B = \mathbb{F}_q$ *and* $F = \mathbb{F}_{q^l}$ *for a prime power* q.
- $k + 1 \leqslant d \leqslant n - 1$
- $\mathcal{C} \subseteq F^n$ *an* (n, k) *scalar linear MDS code with a linear repair scheme over* F
- *Suppose that* $\mathcal{C}$ *supports optimal repair of a single node from* d *helper nodes*

Subpacketization for linear repair of scalar codes

Theorem

- *Let* $B=\mathbb{F}_q$ *and* $F=\mathbb{F}_{q^l}$ *for a prime power* q.
- $k+1 \leqslant d \leqslant n-1$
- $\mathcal{C} \subseteq F^n$ *an* (n,k) *scalar linear MDS code with a linear repair scheme over* F
- *Suppose that* $\mathcal{C}$ *supports optimal repair of a single node from* d *helper nodes*
- *Then*

$$l \geqslant \prod_{i=1}^{k-1} p_i$$

where p_i *is the* i*-th smallest prime.*

Subpacketization for linear repair of scalar codes

Theorem

- *Let* $B = \mathbb{F}_q$ *and* $F = \mathbb{F}_{q^l}$ *for a prime power* q.
- $k + 1 \leqslant d \leqslant n - 1$
- $\mathcal{C} \subseteq F^n$ *an* (n, k) *scalar linear MDS code with a linear repair scheme over* F
- *Suppose that* $\mathcal{C}$ *supports optimal repair of a single node from* d *helper nodes*
- *Then*

$$l \geqslant \prod_{i=1}^{k-1} p_i$$

where p_i *is the* i*-th smallest prime.*

To summarize: Sub-packetization for MDS codes with optimal repair satisfies

Subpacketization for linear repair of scalar codes

Theorem

- *Let* $B = \mathbb{F}_q$ *and* $F = \mathbb{F}_{q^l}$ *for a prime power* q*.*
- $k+1 \leqslant d \leqslant n-1$
- $\mathcal{C} \subseteq F^n$ *an* (n,k) *scalar linear MDS code with a linear repair scheme over* F
- *Suppose that* $\mathcal{C}$ *supports optimal repair of a single node from* d *helper nodes*
- *Then*

$$l \geqslant \prod_{i=1}^{k-1} p_i$$

where p_i *is the* i*-th smallest prime.*

To summarize: Sub-packetization for MDS codes with optimal repair satisfies

- ***Scalar codes:*** $\exp((1+o(1))k\log k) \leqslant l \leqslant \exp((1+o(1))n\log n)$

Subpacketization for linear repair of scalar codes

Theorem

- *Let $B = \mathbb{F}_q$ and $F = \mathbb{F}_{q^l}$ for a prime power q.*
- $k + 1 \leqslant d \leqslant n - 1$
- $\mathcal{C} \subseteq F^n$ *an (n, k) scalar linear MDS code with a linear repair scheme over F*
- *Suppose that $\mathcal{C}$ supports optimal repair of a single node from d helper nodes*
- *Then*

$$l \geqslant \prod_{i=1}^{k-1} p_i$$

where p_i is the i-th smallest prime.

To summarize: Sub-packetization for MDS codes with optimal repair satisfies

- ***Scalar codes:*** $\exp((1 + o(1))k \log k) \leqslant l \leqslant \exp((1 + o(1))n \log n)$
- ***Vector codes:*** $l = r^{\lceil n/r \rceil}$

Multiple erasures

Results for 2,3 erasures (full-length RS codes):

DAU-DUURSMA-KIAH-MILENKOVIC, Repairing Reed-Solomon codes with multiple erasures, 2016

B. BARTAN AND M. WOOTTERS, Repairing multiple failures for scalar MDS codes, 2017

Multiple erasures

Results for 2,3 erasures (full-length RS codes):

DAU-DUURSMA-KIAH-MILENKOVIC, Repairing Reed-Solomon codes with multiple erasures, 2016

B. BARTAN AND M. WOOTTERS, Repairing multiple failures for scalar MDS codes, 2017

The construction discussed above can be extended to optimal repair of multiple erasures:

Theorem

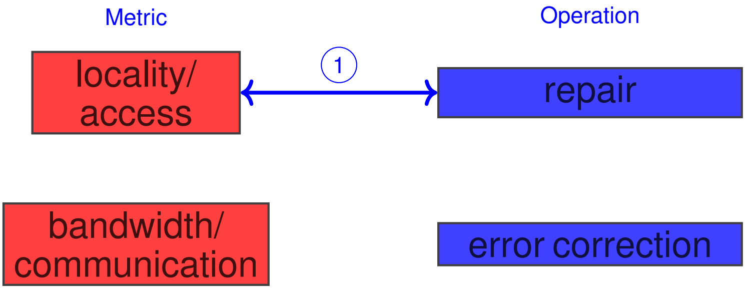
- k, n *positive integers,* $k < n$
- *Let* $h \leqslant r; k \leqslant d \leqslant n - h; s := r!$
- $\Omega = \{\alpha_1, \dots, \alpha_n\}$, *where* $\deg_{\mathbb{F}_q}(\alpha_i) = p_i, i = 1, \dots, n$ *and* $p_i \equiv 1 \bmod s$ *is the* i*th smallest prime*
- *Let* $\mathbb{K} := \mathbb{F}_q(\alpha_1, \dots, \alpha_n, \beta)$, *where* $\deg_{\mathbb{F}}(\beta) = s$
- *The code* $\mathcal{C} := \mathrm{RS}_{\mathbb{K}}(n, k, \Omega)$ *has the universal* (h, d)*-optimal repair property for all* $h \leqslant r$ *and all* $k \leqslant d \leqslant n - h$ *simultaneously.*

I. TAMO, M. YE, AND A.B., *The repair problem for Reed-Solomon codes*, T-IT, May 2019

Local Information Processing

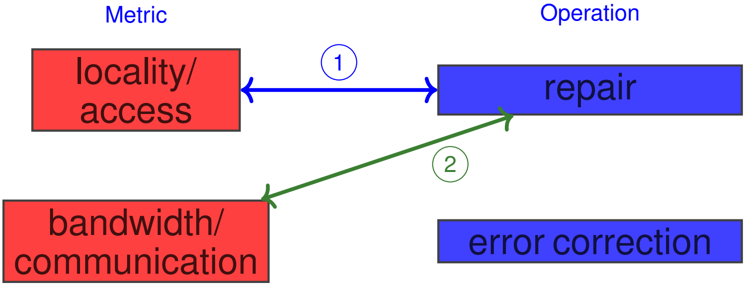
Metric	Operation
locality/ access	repair
bandwidth/ communication	error correction

Local Information Processing



(1) Locally Recoverable codes (local recovery)

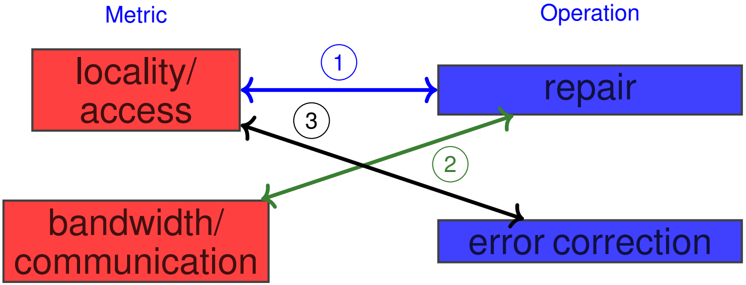
Local Information Processing



(1) Locally Recoverable codes (local recovery)

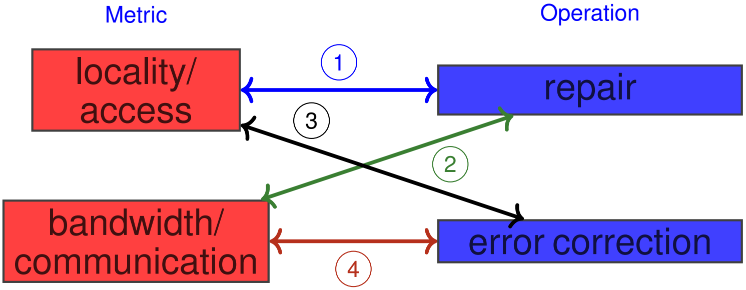
(2) Regenerating codes (local recovery)

Local Information Processing



1. Locally Recoverable codes (local recovery)
2. Regenerating codes (local recovery)
3. LDPC codes (global recovery)

Local Information Processing



1. Locally Recoverable codes (local recovery)
2. Regenerating codes (local recovery)
3. LDPC codes (global recovery)
4. Fractional decoding (global recovery)

I. TAMO, M. YE, AND A.B., *Fractional decoding: Error correction from partial information*, 2018

Heterogeneous storage

- A multitude of models

Heterogeneous storage

- A multitude of models
- The nodes are split into clusters A_1 and A_2. Downloading β bits from A_1 incurs higher cost that from A_2 (S. Akhlaghi et al. 2010)

Heterogeneous storage

- A multitude of models
- The nodes are split into clusters A_1 and A_2. Downloading β bits from A_1 incurs higher cost that from A_2 (S. AKHLAGHI ET AL. 2010)
- The cost of communication between a pair of nodes depends on their relative location in the system (B. GASTÓN ET AL., 2013)

Heterogeneous storage

- A multitude of models
- The nodes are split into clusters A_1 and A_2. Downloading β bits from A_1 incurs higher cost that from A_2 (S. AKHLAGHI ET AL. 2010)
- The cost of communication between a pair of nodes depends on their relative location in the system (B. GASTÓN ET AL., 2013)
- Capacity of clustered storage (J-Y.SOHN ET AL, 2018)

Heterogeneous storage

- A multitude of models
- The nodes are split into clusters A_1 and A_2. Downloading β bits from A_1 incurs higher cost that from A_2 (S. AKHLAGHI ET AL. 2010)
- The cost of communication between a pair of nodes depends on their relative location in the system (B. GASTÓN ET AL., 2013)
- Capacity of clustered storage (J-Y.SOHN ET AL, 2018)
- Rack-aware storage model: Processing of information within the helper rack before downloading (Y. HU, P.C. LEE, AND X. ZHANG, 2016)

Heterogeneous (clustered) model

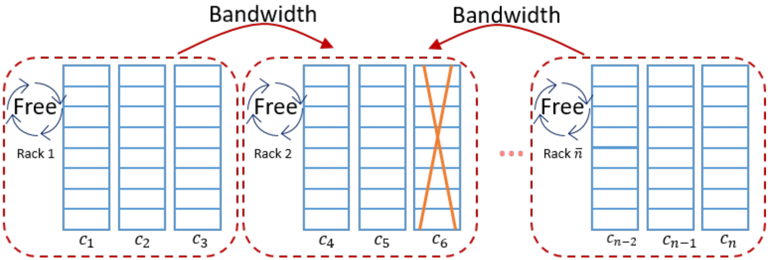
The $n = \bar{n}u$ nodes are further grouped into $\bar{n}$ *racks* of size u each

$$(C_1, \dots, C_u), \dots, (C_{(m-1)u+1}, \dots, C_{(m-1)u+u}), \dots, (C_{(\bar{n}-1)u+1}, \dots, C_{\bar{n}u})$$

Communication *within* each group is free, inter-rack communication counts toward the repair bandwidth

X. Hu, P.P.C. Lee, and X. Zhang, Double regenerating codes for hierarchical data centers, ISIT 2016

Rack-aware storage model



- Encoding of length n is stored in $\bar{n}$ racks, each containing u nodes
- Code length $n = \bar{n}u$
- Only communication between the racks counts toward repair bandwidth

Rack-aware storage model: Repairing single node

Cut-set bound (Hu, Lee, and Zhang): Let $k = \bar{k}u + v$, let $\bar{d}$ be the number of *helper racks*, then

$$\beta \geqslant \frac{\bar{d}l}{\bar{d} - \bar{k} + 1}$$

MSR codes for rack-aware storage

Optimal-repair codes for all parameters

- A combination of the construction of [YE-B., 2017] and subgroup structure of F^*
- Let $\bar{s} = \bar{d} - \bar{k} + 1$. We construct $(n, k, l = \bar{s}^{\bar{n}})$ codes over $F, |F| = q > \bar{s}n$

MSR codes for rack-aware storage

Optimal-repair codes for all parameters

- A combination of the construction of [YE-B., 2017] and subgroup structure of F^*
- Let $\bar{s} = \bar{d} - \bar{k} + 1$. We construct $(n, k, l = \bar{s}^{\bar{n}})$ codes over $F, |F| = q > \bar{s}n$
- Suppose that $\bar{s}n | (q-1)$, let $\lambda \in F$: $\mathrm{ord}(\lambda) = \bar{s}n$.

MSR codes for rack-aware storage

Optimal-repair codes for all parameters

- A combination of the construction of [YE-B., 2017] and subgroup structure of F^*
- Let $\bar{s} = \bar{d} - \bar{k} + 1$. We construct $(n, k, l = \bar{s}^{\bar{n}})$ codes over F, $|F| = q > \bar{s}n$
- Suppose that $\bar{s}n|(q-1)$, let $\lambda \in F$: ord$(\lambda) = \bar{s}n$.

- Parity-check equations of the code $\mathcal{C}$:

$$\sum_{e=1}^{\bar{n}} \lambda^{t((e-1)\bar{s}+j_e)} \sum_{i=1}^{u} \lambda^{t(i-1)\bar{s}\,\bar{n}} C_{(e-1)u+i,j} = 0$$

for all $t = 0, \ldots, r-1; j = 0, \ldots, l-1, j = (j_{\bar{n}}, \ldots, j_1)$

Z. CHEN AND A.B., *MSR codes for the rack-aware model*, ISIT 2019, arXiv:1901.04419

Beyond algebraic constructions: New directions

Beyond algebraic constructions: New directions

- Most questions mentioned below are open

Beyond algebraic constructions: New directions

- Most questions mentioned below are open
- Main idea: Representation and recovery of information in a network

Beyond algebraic constructions: New directions

- Most questions mentioned below are open
- Main idea: Representation and recovery of information in a network
- All the problems considered so far assumed total connectivity

Beyond algebraic constructions: New directions

- Most questions mentioned below are open
- Main idea: Representation and recovery of information in a network
- All the problems considered so far assumed total connectivity
- Repair problem on a graph
  - Structure of the repair protocol: Forward or process?
  - Random graph: thresholds for repair?
  - Random errors in links
  - Adversarial nodes

Beyond algebraic constructions: New directions

- Most questions mentioned below are open
- Main idea: Representation and recovery of information in a network
- All the problems considered so far assumed total connectivity
- Repair problem on a graph
  - Structure of the repair protocol: Forward or process?
  - Random graph: thresholds for repair?
  - Random errors in links
  - Adversarial nodes
- Dynamical models of networks

Z. GOLDFELD, G. BRESLER, AND Y. POLYANSKIY, *Information storage in the Ising model*, 2018

Capacity of dynamical networks

- Previously considered problems: worst-case analysis (min-cut)
- The evolution of the network occurs in time
- Suppose that the nodes fail independently at a fixed Poisson rate
- We are interested in the time-average file size that can be stored in the system for a given repair bandwidth
- Assume moreover that $[n] = U \cup L$, where the nodes in U contribute β_2 symbols, the nodes in L contribute β_1 symbols, and $\beta_2 > \beta_1$
- [O. Elishco and A.B., ISIT 2019] shows that the average size of the file can be higher than the worst-case
- New set of tools: Markov random walk on permutations, mixing times

It’s a holiday!

It's a holiday!