Codes for distributed storage, private information retrieval and low-latency streaming

Myna Vajha

Advisor: Prof. P Vijay Kumar

Department of Electrical Communications Engineering,
Indian Institute of Science, Bangalore

PhD Thesis Defence

Date: 2nd December, 2020

Codes for reliable and efficient distributed storage

Joint work with:

- Birenjith Sasidharan, Balaji S. B (MSR constructions)

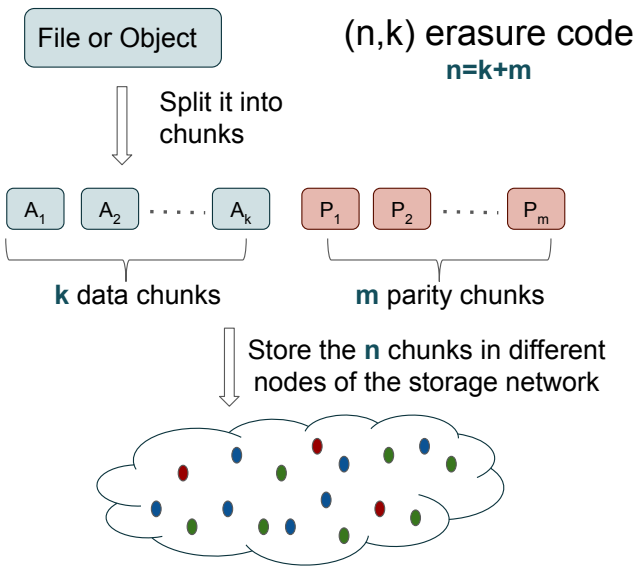

- Vinayak Ramkumar, Bhagyasree Puranik, Ganesh Kini and Elita Lobo(Systems implementation and evaluation)




- Srinivas Narayanamurthy, Syed Hussain, Siddhartha Nandi (Netapp team on Systems work)
- Min Ye and Alexandar Barg (University of Maryland team on Systems work)

Erasure Coding for Fault Tolerance

- Disk/node/rack failures are common in distributed storage
- Erasure coding is used to provide fault tolerance



The n chunks taken together, form a stripe.

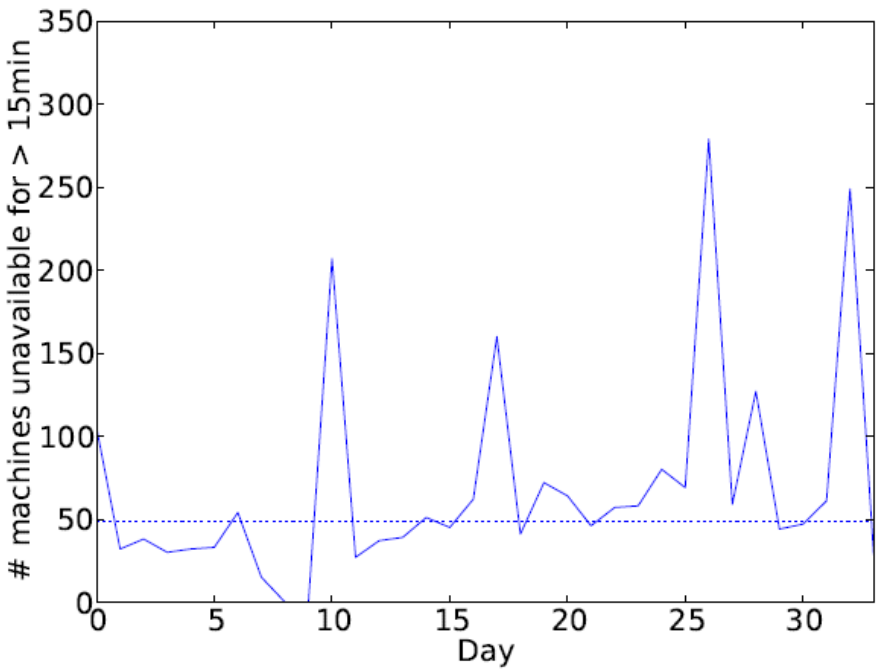
Two Key Performance Measures

1. Storage Overhead $\frac{n}{k}$
2. Fault Tolerance - at most m storage units

MDS Codes

1. For given (n, k), **MDS erasure codes** have the maximum-possible fault tolerance
2. RAID 6 and Reed-Solomon codes are examples of MDS codes.

Erasure Codes and Node Failures

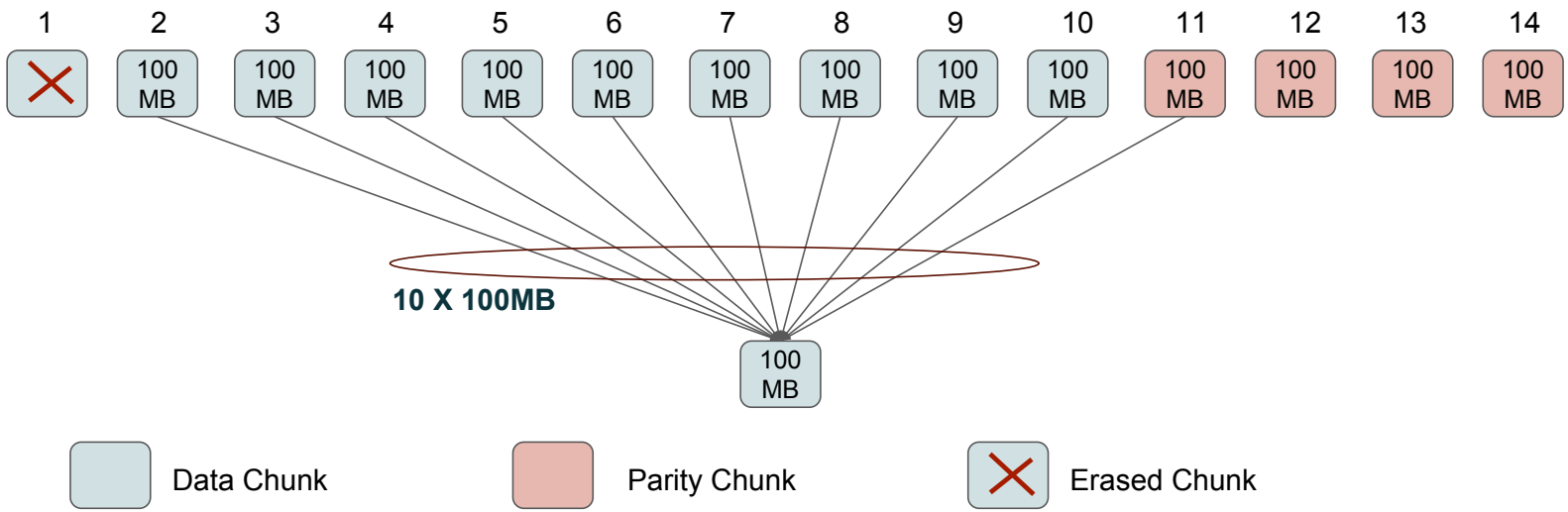


- A median of 50 nodes are unavailable per day.
- 98% of the failures are single node failures.
- A median of 180TB of network traffic per day is generated in order to reconstruct the RS coded data corresponding to unavailable machines.

- **Thus there is a strong need for erasure codes that can efficiently recover from single-node failures.**

Image courtesy: Rashmi et al.: "A Solution to the Network Challenges of Data Recovery in Erasure-coded Distributed Storage Systems: A Study on the Facebook Warehouse Cluster," USENIX Hotstorage, 2013.

Conventional Node Repair of an RS Code



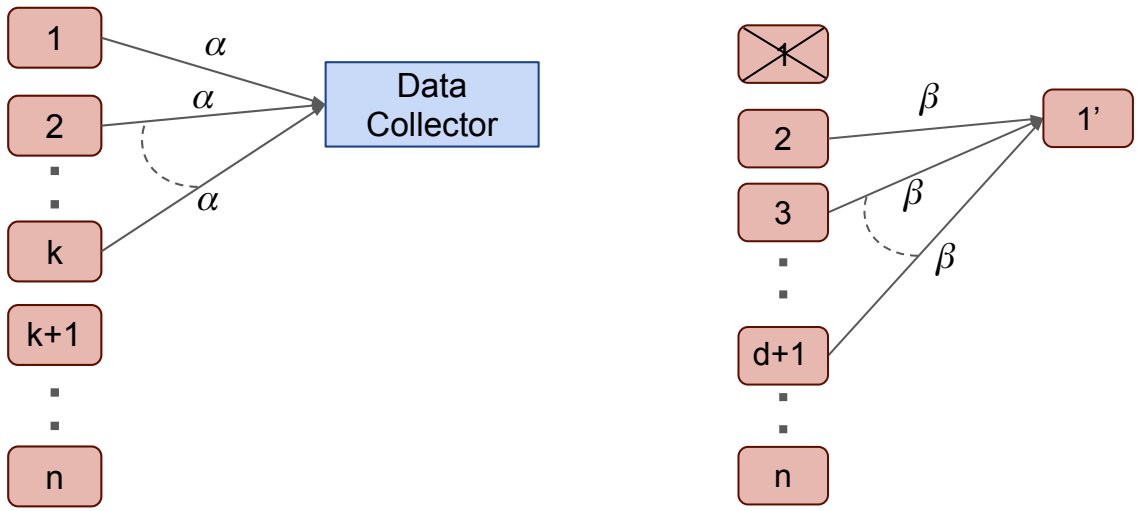
In the example $(14, 10)$ RS code,

1. the amount of data downloaded to repair 100MB of data equals 1GB.

clearly, there is room for improvement...

Regenerating Codes

Parameters: ((n, k, d), (α, β), B, $\mathbb{F}_q$)



- Data (of size B) can be recovered by connecting to any k of n nodes
- A failed node can be repaired by connecting to any d nodes, downloading β symbols from each node; ($d\beta << $ file size B)

A. Dimakis, P. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems", IEEE Transactions on Information Theory, 2010

Regenerating Codes

1. File size B possible by an (n, k, d, α, β) regenerating code:

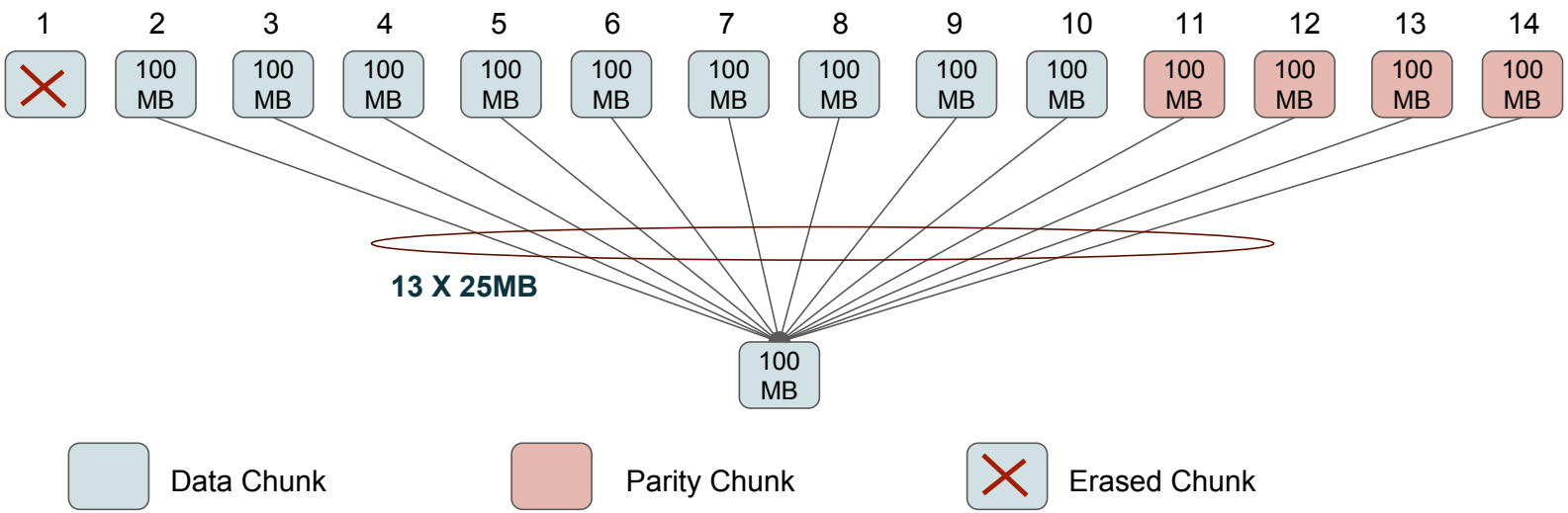
$$\begin{aligned} B &\leq \sum_{i=0}^{k-1} \min(\alpha, (d-i)\beta) \\ &= k\alpha \quad \text{(when storage } \alpha \text{ is minimized)} \end{aligned}$$

2. Minimum storage regenerating (MSR) codes are a subclass of regenerating codes such that:

$$\alpha = \frac{B}{k}, \quad \beta = \frac{\alpha}{d-k+1}$$

3. We restrict to Minimum-Storage-Regenerating (MSR) codes – repair-optimal MDS codes.

MSR Code

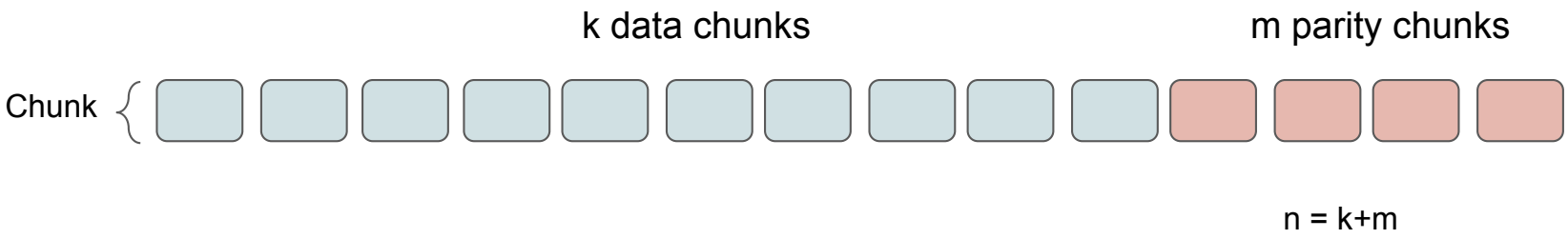


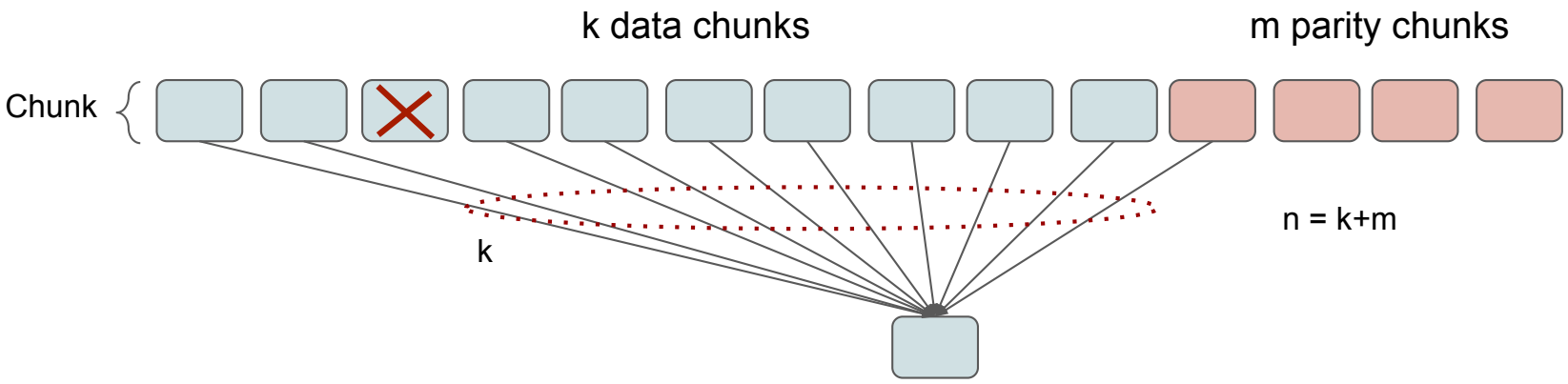
1. Size of failed node's contents: 100MB
2. RS repair BW: 1 GB
3. MSR Repair BW: 325 MB

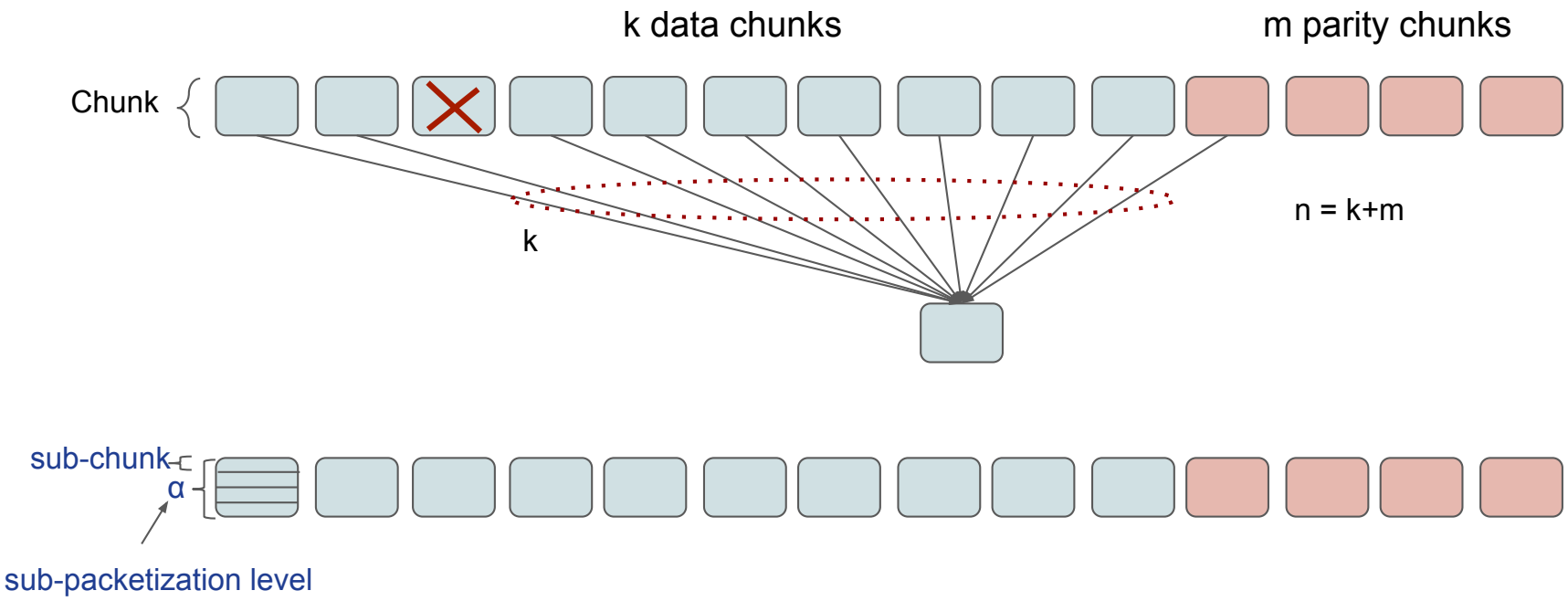
Key to the Impressive, Low-Repair BW of MSR Codes

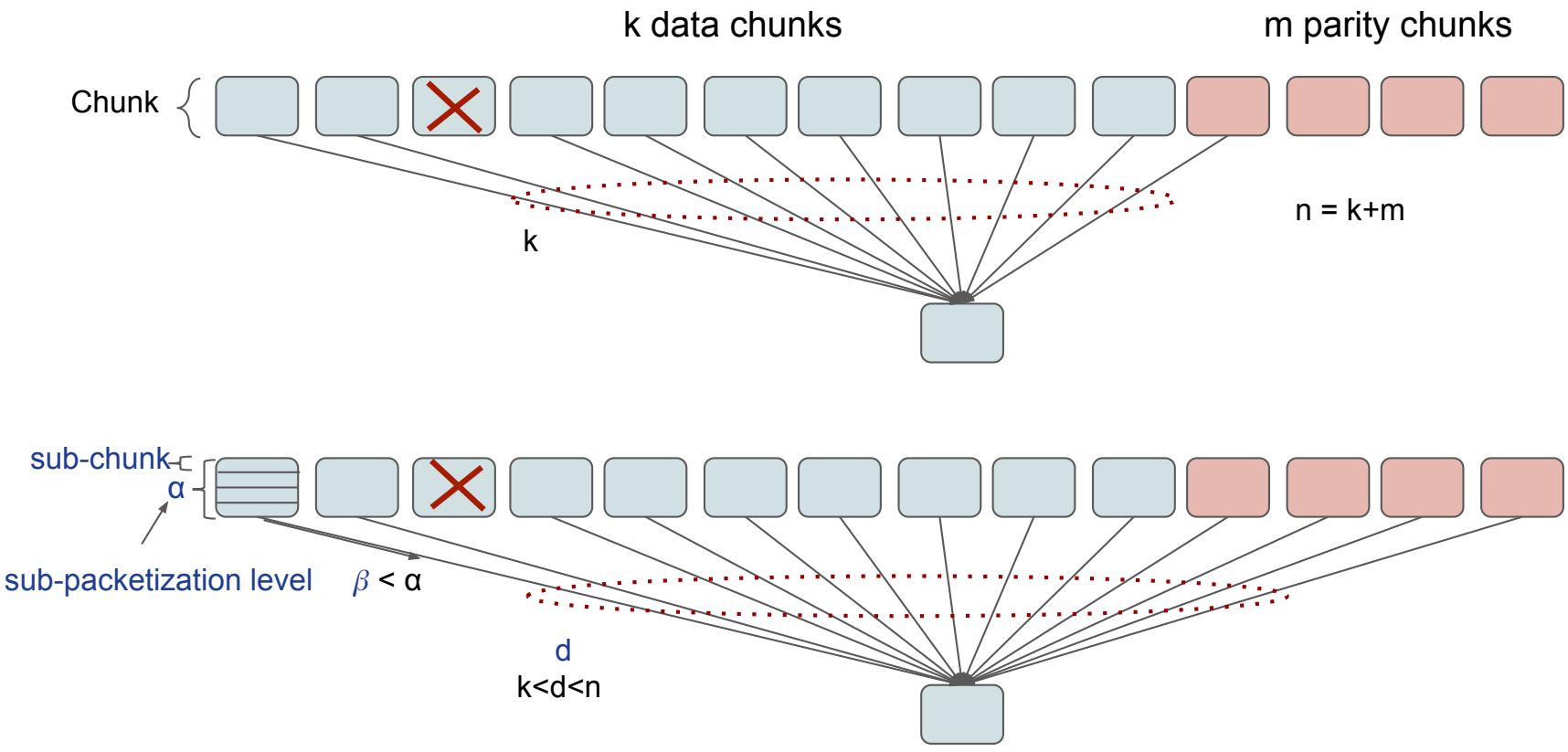
Key to the Impressive, Low-Repair BW of MSR Codes

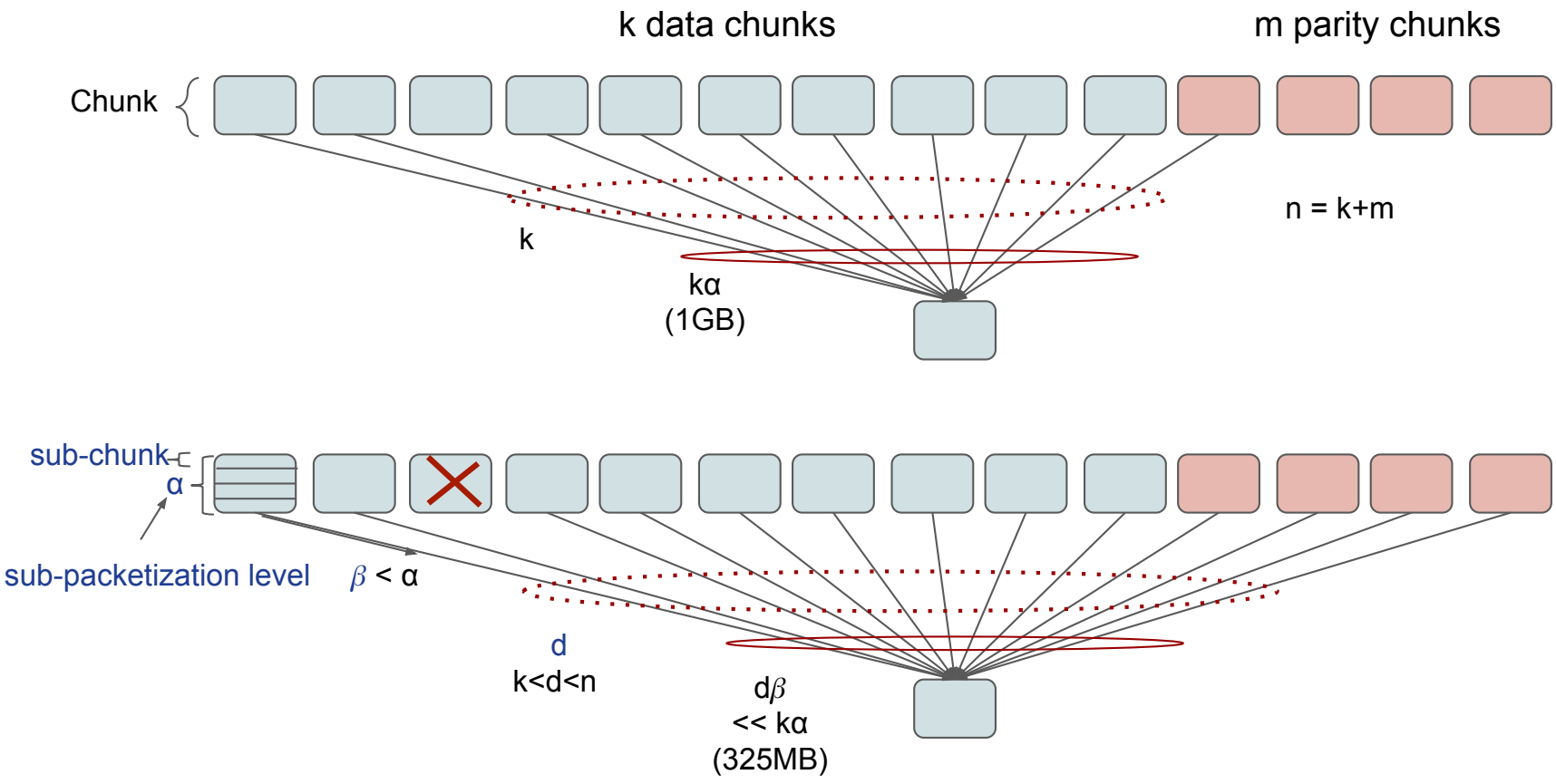
In a nutshell: sub-packetization... we explain...

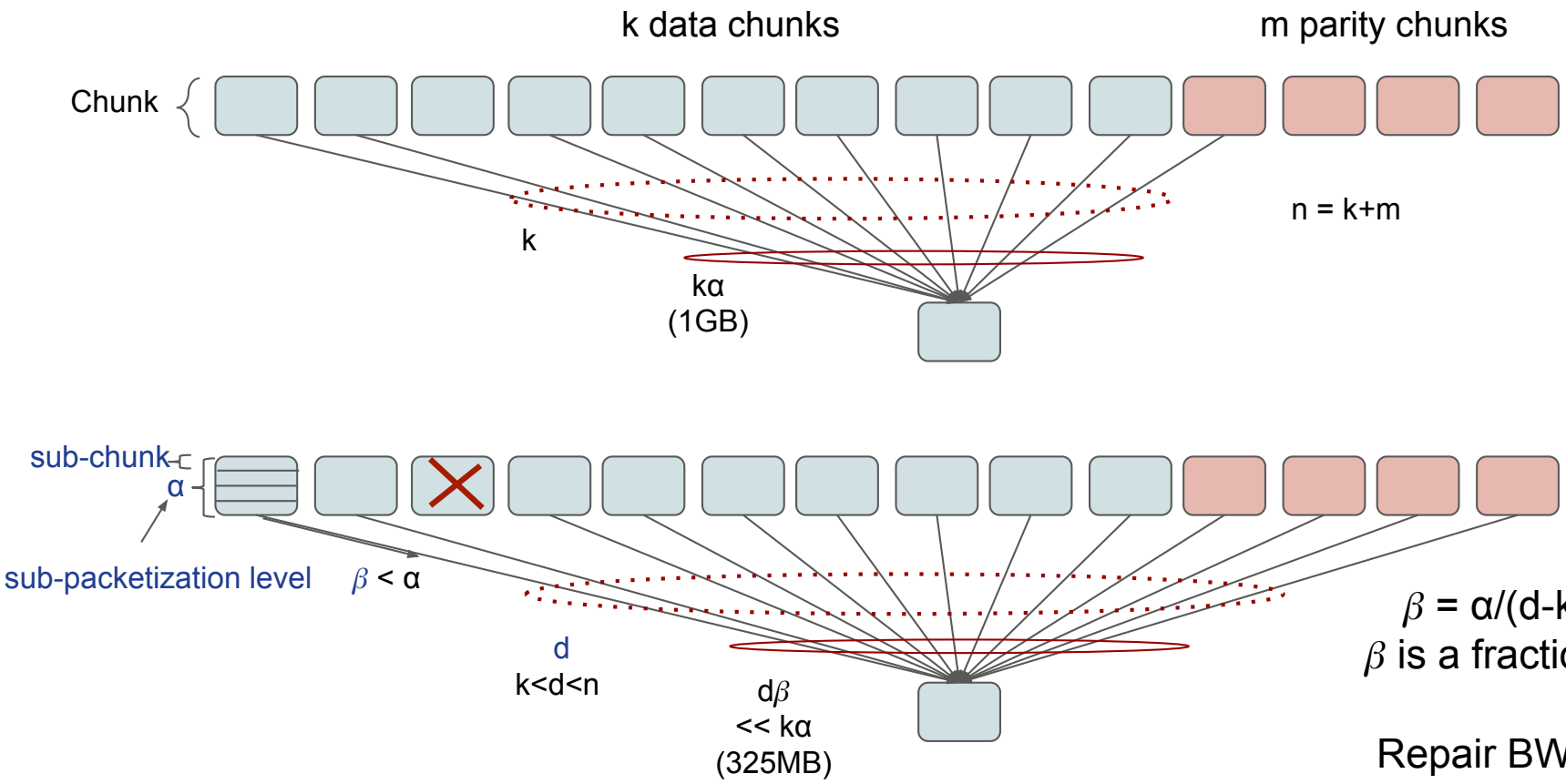

k data chunks
m parity chunks
Chunk
n = k+m


k data chunks
m parity chunks
Chunk
k
n = k+m


k data chunks
m parity chunks
Chunk
k
n = k+m
sub-chunk
α
sub-packetization level


k data chunks
m parity chunks
Chunk
k
n = k+m
sub-chunk
α
sub-packetization level
β < α
d
k<d<n


k data chunks
m parity chunks
Chunk
k
kα
(1GB)
n = k+m
sub-chunk
α
sub-packetization level
β < α
d
k<d<n
dβ
<< kα
(325MB)



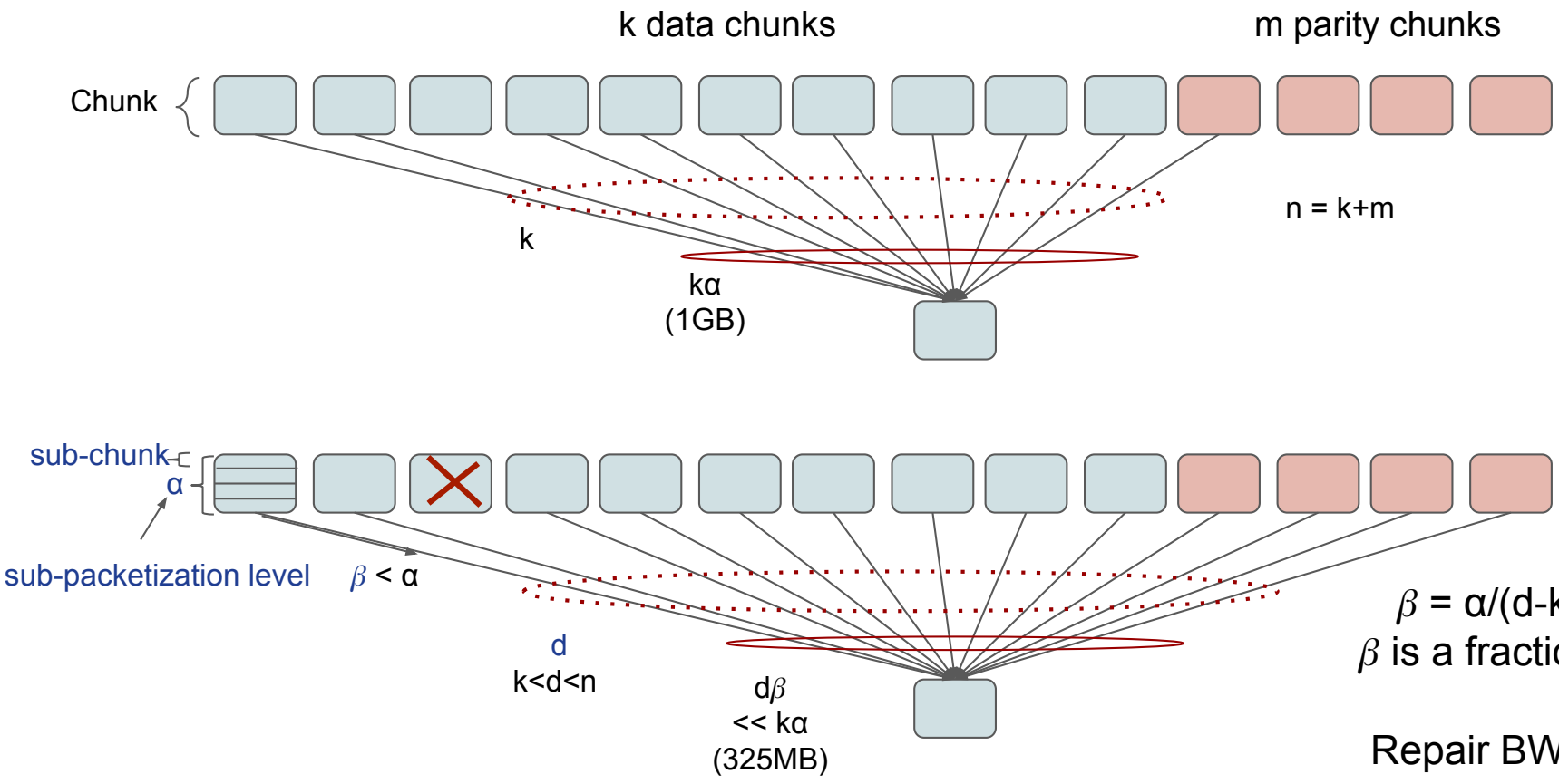
$\beta = \alpha/(d-k+1)$

β is a fraction of α

Repair BW = dβ

We consider d=n-1, then

Repair BW = (n-1)α/(n-k)



$\beta = \alpha/(d-k+1)$
β is a fraction of α

Repair BW = dβ
We consider d=n-1, then
Repair BW = (n-1)α/(n-k)

Larger the m=n-k, larger the savings!!

Additional Properties Desired of an MSR Code

1. Minimal Disk Read (IO Optimality): Read exactly what is needed to be transferred

2. Minimize sub-packetization level α
   - sub-chunk size $= \frac{\text{chunk size}}{\alpha} = N$ bytes.
   - During repair, β sub-chunks are read.
   - If sub-chunks are not contiguous, only N bytes are read sequentially.
   - Smaller the α better the sequentiality!!

3. Small field size, low-complexity implementation.

4. Two family of constructions
   - Coupled Layer (CLay) MSR code ($d = n - 1$)
   - Small d MSR code ($d = k + 1, k + 2, k + 3$)

4-way Optimality of Clay code

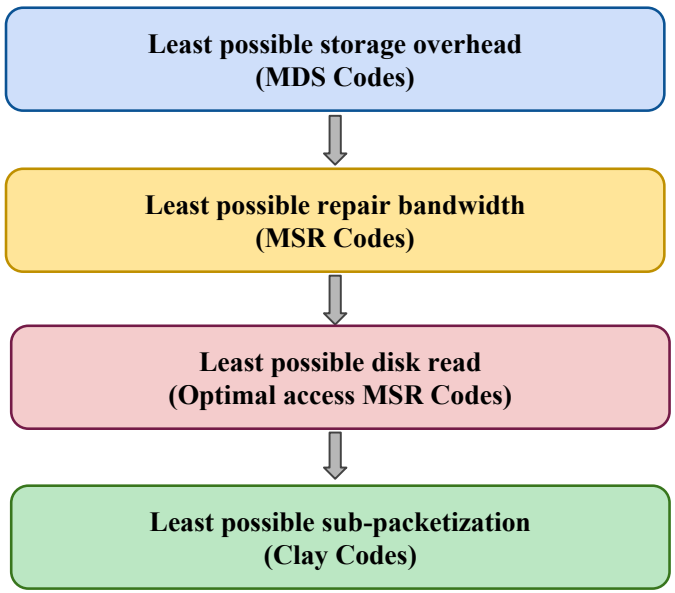


Image courtesy: denverpost.com

Putting Clay codes in perspective

- $s = d - k + 1$, $r = n - k$

MSR Code	Parameters	α	Field Size (q)	All Node Repair	Optimal Access
Shah et al.	$(n, k, d = n - 1 \geq 2k - 1)$	r	$2r$	No	Yes
Suh et al.	$(n, k, d \geq 2k - 1)$ $(n, k \leq 3, d)$	s	$2r$	Yes	No
Rashmi et al.	$(n \geq 2k - 1, k, d)$	r	n	Yes	No
Papailiopoulos et al.	$(n, k, d = n - 1)$	r^k	non-explicit	No	No
Tamo et al. Wang et al.	$(n, k, d = n - 1)$	r^{k+1}	≤ 4 when $r \leq 3$, else non-explicit	Yes	Yes
Cadambe et al.	$(n \geq \frac{3k}{2}, k, d = n - 1)$	$O(k^2)$	non-explicit	No	Yes
Sasidharan et al.	$(n, k, d = n - 1)$	$r^{\lceil \frac{n}{r} \rceil}$	$O(n^r)$	Yes	Yes
Goparaju et al.	(n, k, d)	$s^{k\binom{r}{s}}$	-	No	Yes
Rawat et al.	(n, k, d)	$s^{\lceil \frac{n}{s} \rceil}$	$O(n^r)$	Yes	Yes
Ye & Barg (1a)	(n, k, d)	s^n	sn	Yes	No
Ye & Barg (1b)	(n, k, d)	s^{n-1}	$n + 1$	Yes	Yes

Literature on High-Rate, OA MSR Codes with Optimum α

$$(n, k, d = n-1, \alpha = r^{\lceil \frac{n}{r} \rceil}),\ q \geq r\lceil \frac{n}{r} \rceil$$

- Three teams independently discovered the Clay code construction with slight variations. Ours being one team and the others by (1) Ye & Barg and (2) Li, Tang & Tian.

- Sub-packetization bounds for optimal access MSR codes
  - Shown to be $\alpha \geq r^{\frac{k}{r}}$ for $d = n-1$ by Tamo et al.
  - This bound is tightened to $\alpha \geq r^{\frac{n}{r}}$ by Balaji et al.

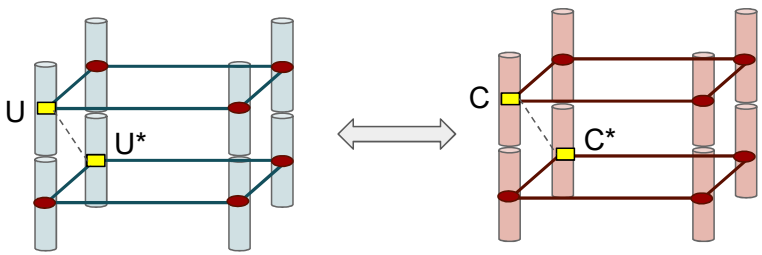
Systems Implementations of Bandwidth Efficient MDS codes

Code	MDS	Least Repair BW	Least Disk Read	Least α	Restrictions	Implemented Distributed Systems
Piggybacked RS (Sigcomm 2014)	✔	✗	✗	-	None	HDFS
Product Matrix (FAST 2015)	✔	✔	✔	✔	**Limited to Storage Overhead > 2**	Own System
Butterfly Code (FAST 2016)	✔	✔	✗	✗	Limited to the 2 parity nodes	HDFS, Ceph
HashTag Code (Trans. on Big Data 2017)	✔	✗	✗	-	Only systematic node repair	HDFS
Clay (FAST 2018)	✔	✔	✔	✔	**None!**	Ceph

- The Butterfly, HashTag codes have least disk read for systematic node repair.

Moulding an MDS Code to Yield the Clay Code

$(n = 4, k = 2)$ MDS code with optimal repair of systematic nodes, $\alpha = 2$



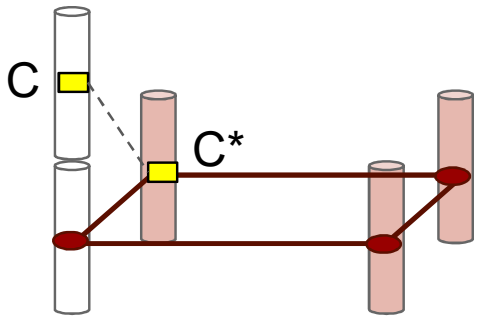
UnCoupled Code Coupled Code

- Uncoupled code has 2 planes, where each plane corresponds to an $[4, 2]$ MDS code
- Coupled code symbols are obtained by:
  - Copying symbols with red dots
  - Pair of yellow symbols $\{C, C^*\}$ are obtained by transformation

$$\begin{bmatrix} C \\ C^* \end{bmatrix} = \begin{bmatrix} 1 & \gamma \\ \gamma & 1 \end{bmatrix} \begin{bmatrix} U \\ U^* \end{bmatrix}$$

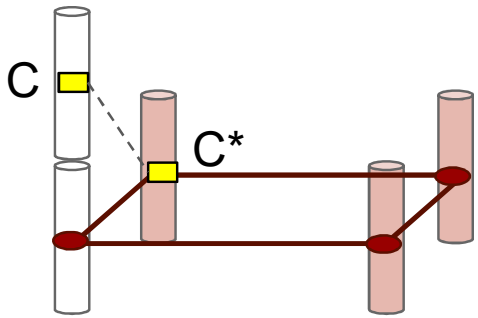
- Note that recovery of any failed node in Uncoupled code requires 4 symbols

Repair from single node loss



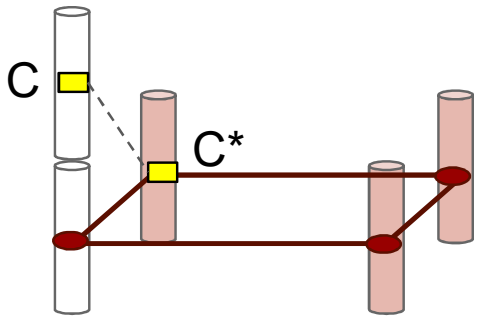
symbols that are available as part of helper information

Repair from single node loss

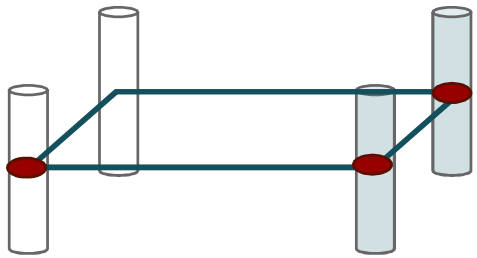


symbols that are available as part of helper information

Repair from single node loss

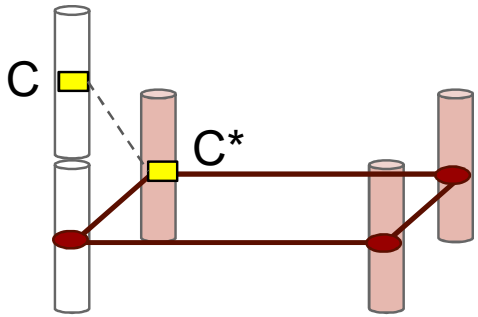


symbols that are available as part of helper information

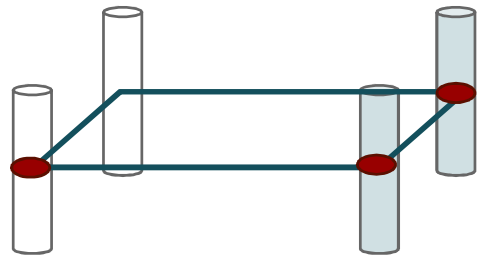


symbols that are computable in uncoupled cube

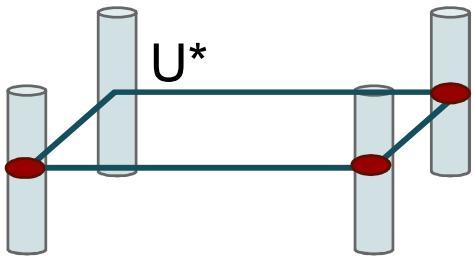
Repair from single node loss



symbols that are available as part of helper information

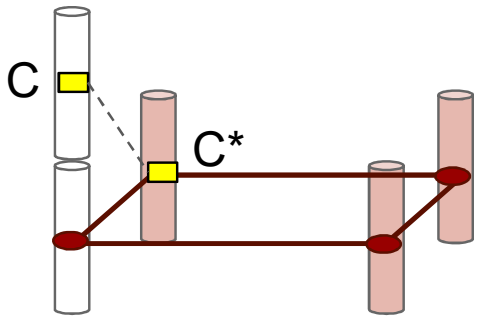


symbols that are computable in uncoupled cube

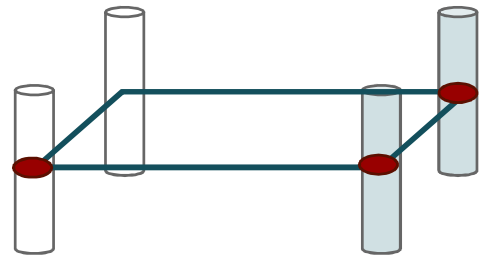


symbols recovered after using the $[4, 2]$ MDS code

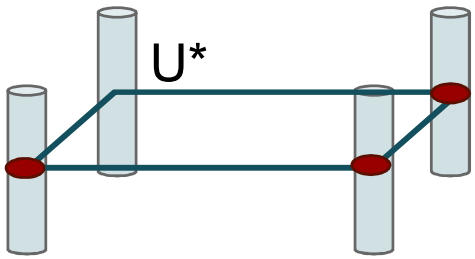
Repair from single node loss



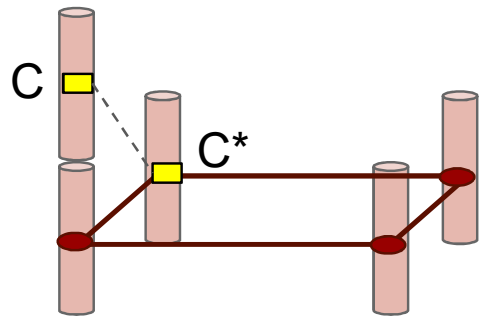
symbols that are available as part of helper information



symbols that are computable in uncoupled cube



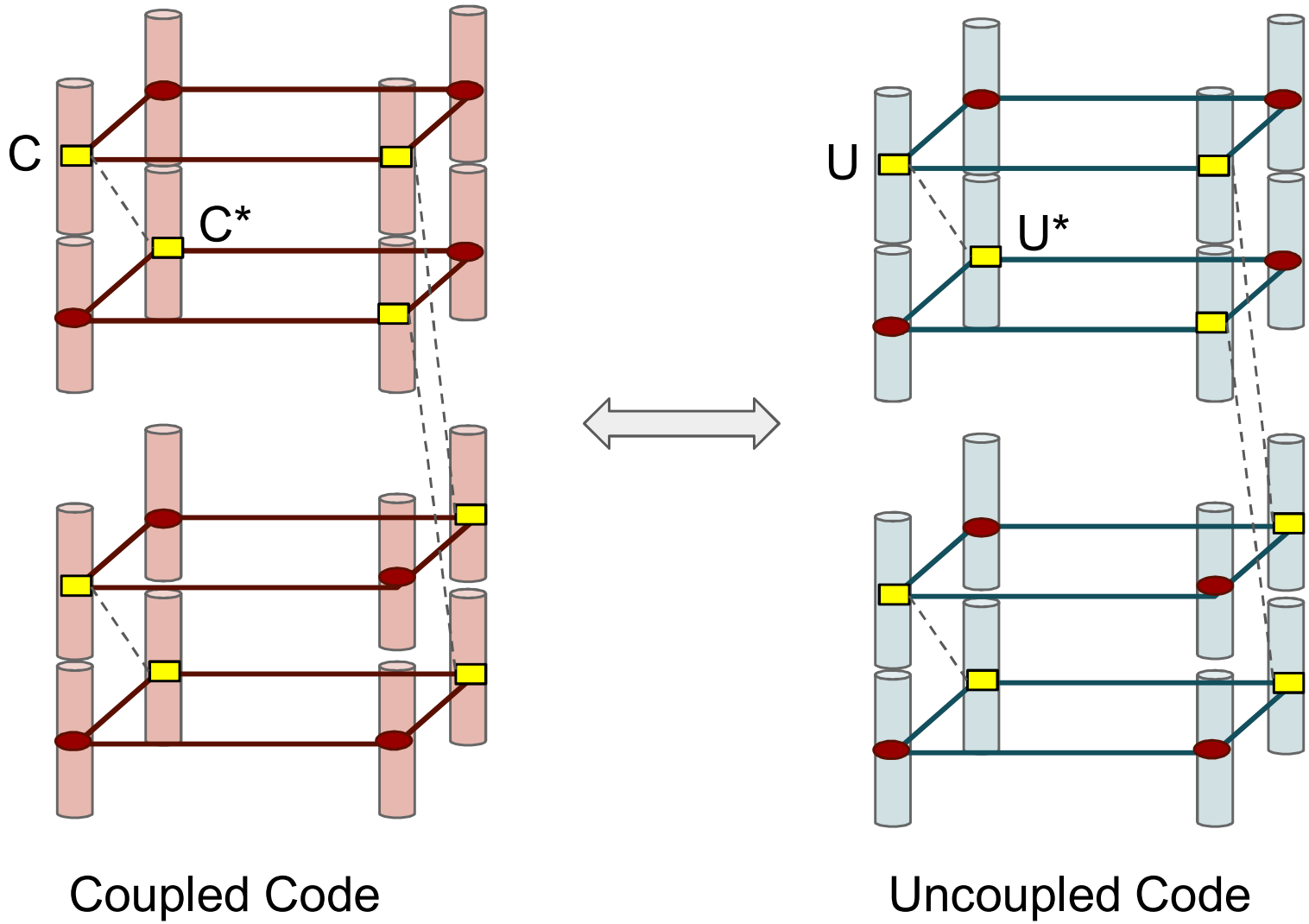
symbols recovered after using the $[4, 2]$ MDS code



C recovered from C^*, U^*

Clay Code

$(n = 4, k = 2, d = 3)$ MSR code with all node optimal repair



- The same construction extends to any (n, k, d)

Ceph: Contributions

A popular opensource distributed storage system used by CERN, Flipkart, Cisco etc

Ceph: Contributions

A popular opensource distributed storage system used by CERN, Flipkart, Cisco etc

"Us (+Vinayak) pitching Clay codes to Ceph in April 2017"

We have introduced Clay code as erasure code plugin. It is part of Ceph's Nautilus release (March 2019). As part of this we also introduced support for vector codes in Ceph.

Clay Code Summary

- The open-source implementation of Clay code that we provide is for any (n, k, d) parameters.
- In comparison to $(20, 16)$ RS code, for Workloads with large sized objects (64MB), the Clay code $(20, 16, 19)$:
  - resulted in repair time reduction by $3X$.
  - Improved degraded read and write performance by 27.17% and 106.68% respectively.
- For the case when $d < n - 1$, the clay codes are not exactly MSR, though they have optimal repair bandwidth. (few compulsory helper nodes $(d - k)$ need to be contacted during a node's recovery).

Second MSR Constructions

- MSR Construction ($d < n-1$) with parameters:

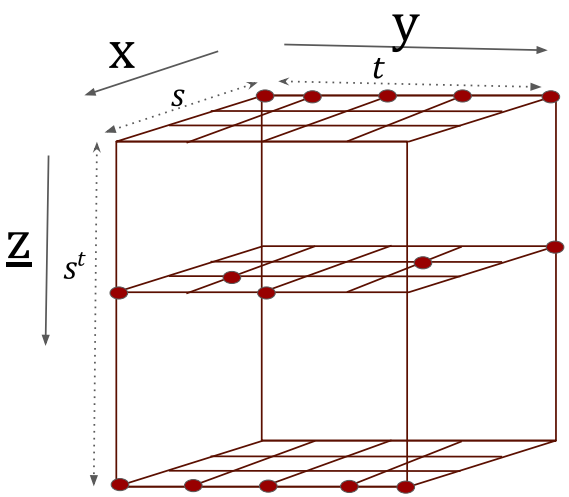
$$(n = st,\ k,\ d),\ (\alpha = s^t,\ \beta = s^{t-1},\ \mathbb{F}_q)$$

  for any $s \in \{2,3,4\},\ t \geq 2$, where $s = d - k + 1$.

- (n,k,d) MSR codes for $d \in \{k+1, k+2, k+3\}$ can be obtained by shortening

MSR Construction: 3D Representation of a Codeword

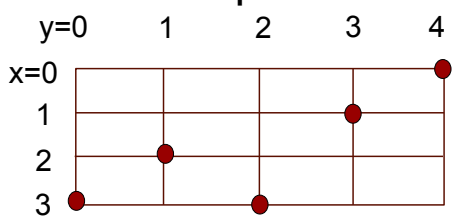
$$(n = st,\ k,\ d),\ (\alpha = s^t,\ \beta = s^{t-1},\ \mathbb{F}_q),\ s = d - k + 1,\quad r = n - k$$



$s = 4,\ t = 5$

- There are $n\alpha = s \times t \times s^t$ code symbols in $\mathbb{F}_q$.
- They can be indexed by 3-tuple $(x, y; \underline{z})$ where $x \in \mathbb{Z}_s$, $y \in \mathbb{Z}_t$, $\underline{z} \in \mathbb{Z}_s^t$.
- (x, y) tuple indicates node, $\underline{z}$ index the symbols index within α symbols.

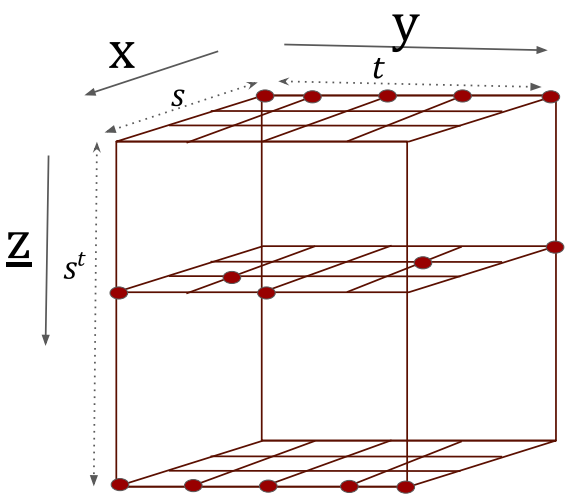
Plane dot representation



$\underline{z} = (3, 2, 3, 1, 0)$

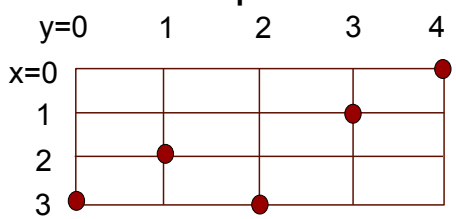
MSR Construction: 3D Representation of a Codeword

$$(n = st,\ k,\ d),\ (\alpha = s^t,\ \beta = s^{t-1},\ \mathbb{F}_q),\ s = d - k + 1,\quad r = n - k$$



$s = 4,\ t = 5$

Plane dot representation



$\underline{z} = (3, 2, 3, 1, 0)$

- There are $n\alpha = s \times t \times s^t$ code symbols in $\mathbb{F}_q$.
- They can be indexed by 3-tuple $(x, y; \underline{z})$ where $x \in \mathbb{Z}_s$, $y \in \mathbb{Z}_t$, $\underline{z} \in \mathbb{Z}_s^t$.
- (x, y) tuple indicates node, $\underline{z}$ index the symbols index within α symbols.
- The code is described by $r\alpha$ parity check equations over $n\alpha$ symbols.
  - This can be viewed as r equations per plane.

MSR Construction: Parity Checks

- The r parity check equations corresponding to plane $\underline{z}$ are given by:

$$\underbrace{\sum_{y\in\mathbb{Z}_t}\sum_{x\in\mathbb{Z}_s}\theta^{\ell}_{x,y;z_y}C(x,y,\underline{z})}_{\text{in-plane symbols}}+\underbrace{\sum_{y\in\mathbb{Z}_t}\sum_{x\neq z_y}\gamma\theta^{\ell}_{z_y,y;x}C(z_y,y,\underline{z}(y,x))}_{\text{out-of-plane symbols}} \quad = \quad 0$$

for all $\ell\in[0,r-1]$, where $\underline{z}=(z_0,z_1,\cdots,z_{t-1})$ and

$$\underline{z}(y,x)=(z_0,\cdots,z_{y-1},x,z_{y+1},\cdots,z_{t-1}),\ \gamma^2\neq 0,1.$$

MSR Construction: Parity Checks

- The r parity check equations corresponding to plane $\underline{z}$ are given by:

$$\underbrace{\sum_{y\in\mathbb{Z}_t}\sum_{x\in\mathbb{Z}_s}\theta^{\ell}_{x,y;z_y}C(x,y,\underline{z})}_{\text{in-plane symbols}}+\underbrace{\sum_{y\in\mathbb{Z}_t}\sum_{x\neq z_y}\gamma\theta^{\ell}_{z_y,y;x}C(z_y,y,\underline{z}(y,x))}_{\text{out-of-plane symbols}} \quad = \quad 0$$

  for all $\ell \in [0, r-1]$, where $\underline{z} = (z_0, z_1, \cdots, z_{t-1})$ and

$$\underline{z}(y,x) = (z_0, \cdots, z_{y-1}, x, z_{y+1}, \cdots, z_{t-1}),\ \gamma^2 \neq 0, 1.$$

- The in-plane symbols and out-of plane symbols

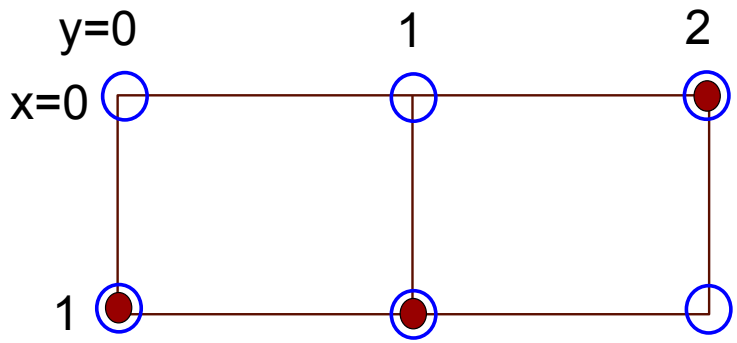
$$\{C(x,y,\underline{z}) \mid x \in \mathbb{Z}_s, \mathbb{Z}_t\}, \quad \{C(z_y,y,\underline{z}(y,x)), y \in \mathbb{Z}_t, x \in \mathbb{Z}_s \setminus \{z_y\}\}$$

  together form a GRS code that can correct from any r erasures.

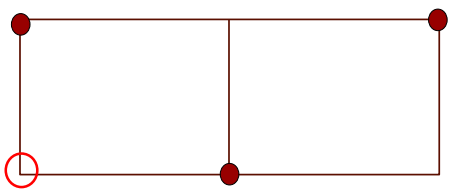
- GRS type property is not needed when $d = n-1$ i.e. for the Clay codes (hence θ's) repeat for clay codes.

MSR Construction: Out of Plane Symbols

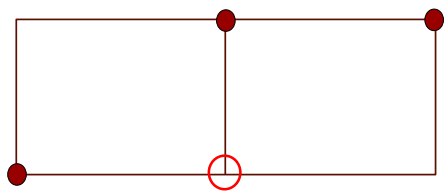
- $s = 2,\ t = 3$
- Circled symbols are involved in parity checks of plane $\underline{z} = (1, 1, 0)$
- Blue circles are in-plane and Red are out-of-plane



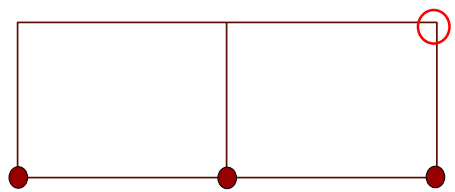
$\underline{z} = (1, 1, 0)$



$\underline{z}(0, 0) = (0, 1, 0)$



$\underline{z}(1, 0) = (1, 0, 0)$



$\underline{z}(2, 1) = (1, 1, 1)$

MSR Construction: Theta Assignment

- The θ's that appear in parity checks of plane $\underline{z}$ are:

$$\underbrace{\{\theta_{x,y,z_y} \mid x \in \mathbb{Z}_s, y \in \mathbb{Z}_t\}}_{\text{in-plane } \theta\text{'s}} \cup \underbrace{\{\theta_{z_y,y,x} \mid x \in \mathbb{Z}_s \setminus \{z_y\}, y \in \mathbb{Z}_t\}}_{\text{out-of-plane } \theta\text{'s}}$$

- Need that collection to have distinct θ's for GRS property.
- We define Θ_y for every $y \in \mathbb{Z}_t$:

$$\Theta_y(x, x') = \theta_{x,y,x'}, \quad \forall x, x' \in \mathbb{Z}_s$$

- The collection of elements in i-th row, i-th column $\{\Theta_y(i, x), \Theta_y(x, i) \mid x \in \mathbb{Z}_s\}$ are all distinct for every $i \in \mathbb{Z}_s \implies$ to get the GRS code for plane where $z_y = i$
- $\theta_{x,y,x} = \theta_y$ for all $x \in \mathbb{Z}_s \implies$ to satisfy MDS property
- For $s = 2$

$$\Theta_y = \begin{bmatrix} \theta_y & \theta_{1,y} \\ \theta_{2,y} & \theta_y \end{bmatrix}$$

- For $s = 2$, need $q \geq 3t = \frac{3n}{2}$

An Example: $s = 2, r = 3$

$$(n = 2t, k = n - 3, d = n - 2)$$

Node Repair:

- During node repair $n - 2$ nodes are contacted out of the $n - 1$ remaining nodes.
- 1 node remains aloof during repair process.

An Example: $s = 2, r = 3$

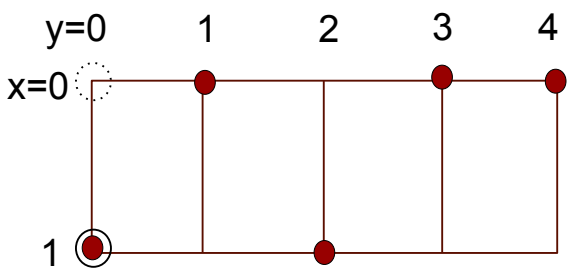
$$(n = 2t, k = n - 3, d = n - 2)$$

Node Repair:

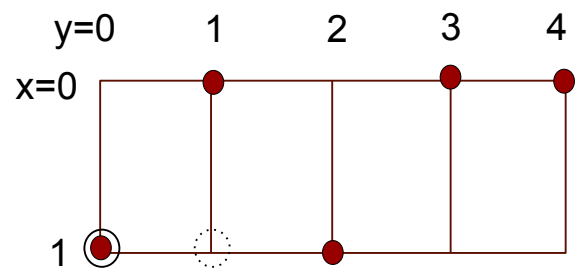
- During node repair $n - 2$ nodes are contacted out of the $n - 1$ remaining nodes.
- 1 node remains aloof during repair process.
- Let (x_0, y_0) be the lost node, (x_1, y_1) be the aloof node.
- Helper information sent by node (x, y) is given by:

$$\{C(x, y, \underline{z}) \mid z_{y_0} = x_0\} \implies \beta = 2^{t-1}$$

- There are two possibilities for aloof node: 1) $y_1 = y_0$, 2) $y_1 \neq y_0$.



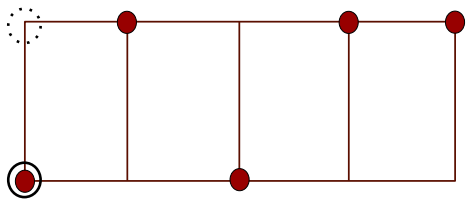
$(x_0, y_0) = (1, 0)$, $(x_1, y_1) = (0, 0)$

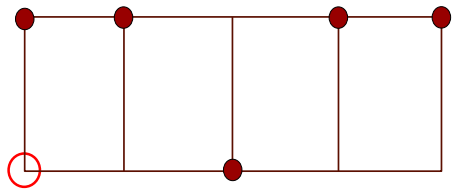


$(x_0, y_0) = (1, 0)$, $(x_1, y_1) = (1, 1)$

An Example: $s = 2, r = 3$

- $y_1 = y_0$, for all $\ell \in [0, 2]$,





$$\theta^{\ell}_{x_0,y_0,x_0} C(x_0, y_0, \underline{z}) + \theta^{\ell}_{x_1,y_1,z_{y_1}} C(x_1, y_1, \underline{z}) + \gamma\theta^{\ell}_{x_0,y_0,\bar{x}_0} C(x_0, y_0, \underline{z}(y_0, \bar{x}_0)) = \kappa^*$$

- For each $\underline{z}$ such that $z_{y_0} = x_0$, two lost node symbols with indices $\underline{z}$, $\underline{z}(y_0, \bar{x}_0)$ are obtained.
- There are 2^{t-1} planes with $z_{y_0} = x_0$, therefore $2 \times 2^{t-1} = \alpha$ symbols are recovered.

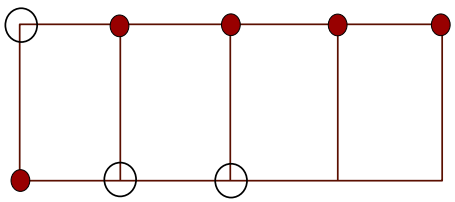
An Example: $s = 2, r = 3$

MDS Property:

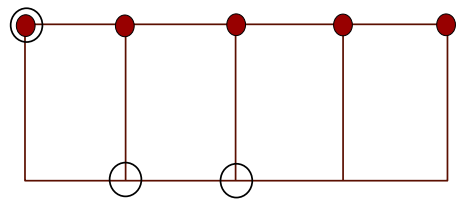
- The code should be able to recover from any $r = 3$ erasure patterns.
- Given an r erasure pattern E, each plane is associated with a score called Intersection Score (IS).

$$IS(E, \underline{z}) = |\{(z_y, y) \in E | y \in \mathbb{Z}_t\}|$$

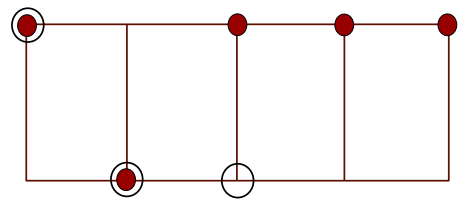
- For, $E = \{(0,0), (1,1), (1,2)\}$



IS $= 0$



IS $= 1$



IS $= 2$

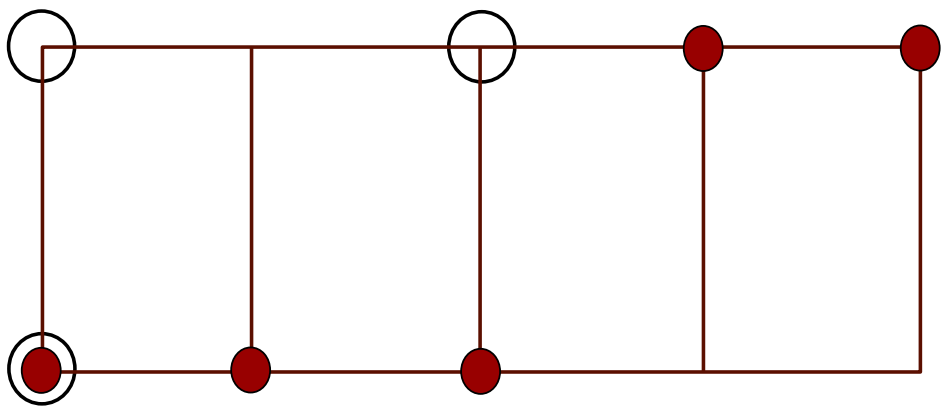
- Intersection score is number of hole-dot pairs in the plane-dot representation.

An Example: $s = 2, r = 3$

- Planes are ordered by their intersection score and erased symbols are recovered sequentially.
- Sometimes few planes with same intersection scores are to be solved together.

An Example: $s = 2, r = 3$

- Planes are ordered by their intersection score and erased symbols are recovered sequentially.
- Sometimes few planes with same intersection scores are to be solved together.
- We will look at an erasure pattern of the form:
  - Two erasures with same y value i.e., $E = \{(0, y_1), (1, y_1), (x_2, y_2)\}$

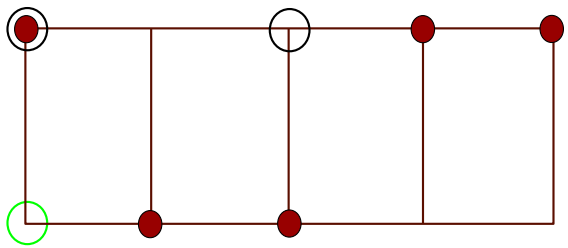


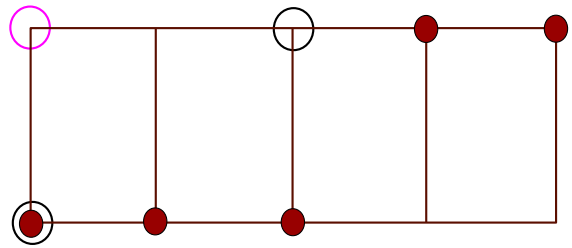
- For this case, planes can have intersection scores $1, 2$

An Example: $s = 2, r = 3$

Two erasures with same y value i.e., $E = \{(0, y_1), (1, y_1), (x_2, y_2)\}$

$$\sum_{y \in \mathbb{Z}_t} \sum_{x \in \mathbb{Z}_s} \theta^{\ell}_{x,y;z_y} C(x, y, \underline{z}) + \sum_{y \in \mathbb{Z}_t} \sum_{x \neq z_y} \gamma_{x,z_y} \theta^{\ell}_{z_y,y;x} C(z_y, y, \underline{z}(y, x)) \quad = \quad 0$$





- $IS(E, \underline{z}) = 1$, $z_{y_1} = 0$, $z_{y_2} \neq x_2$ reduces to:

$$\sum_{(x,y) \in E} \theta^{\ell}_{x,y;z_y} C(x, y, \underline{z}) + \gamma_{1,0} \theta^{\ell}_{0,y_1,1} C(0, y_1, \underline{z}(y_1, 1)) \quad = \quad \kappa^*$$

- Look at plane $\underline{z}' = \underline{z}(y_1, 1)$, $IS(E, \underline{z}) = 1$ and

$$\sum_{(x,y) \in E} \theta^{\ell}_{x,y,z'_y} C(x, y, \underline{z}') + \gamma_{0,1} \theta^{\ell}_{1,y_1,0} C(1, y_1, \underline{z}) \quad = \quad \kappa^*$$

- 6 equations and 6 unknowns.

An Example: $s = 2, r = 3$

$$H_{E,\underline{z}} = \left[\begin{array}{ccc|ccc} 1 & 1 & 1 & \gamma & & \\ \theta_{0,y_1,0} & \theta_{1,y_1,0} & \theta_{x_2,y_2,z_{y_2}} & \gamma\theta_{0,y_1,1} & & \\ \theta_{0,y_1,0}^2 & \theta_{1,y_1,0}^2 & \theta_{x_2,y_2,z_{y_2}}^2 & \gamma\theta_{0,y_1,1}^2 & & \\ \hline & \gamma & & 1 & 1 & 1 \\ & \gamma\theta_{1,y_1,0} & & \theta_{0,y_1,1} & \theta_{1,y_1,1} & \theta_{x_2,y_2,z_{y_2}} \\ & \gamma\theta_{1,y_1,0}^2 & & \theta_{0,y_1,1}^2 & \theta_{1,y_1,1}^2 & \theta_{x_2,y_2,z_{y_2}}^2 \end{array}\right]$$

- Let $v = (f_0, f_1, f_2, g_0, g_1, g_2)^T$ be an vector in left null space of H_S. Let,

$$f(x) = \sum_{j=0}^{2} f_j x^j \text{ and } g(x) = \sum_{j=0}^{2} g_j x^j.$$

An Example: $s = 2, r = 3$

$$H_{E,\underline{z}} = \left[\begin{array}{ccc|ccc} 1 & 1 & 1 & \gamma & & \\ \theta_{0,y_1,0} & \theta_{1,y_1,0} & \theta_{x_2,y_2,z_{y_2}} & \gamma\theta_{0,y_1,1} & & \\ \theta_{0,y_1,0}^2 & \theta_{1,y_1,0}^2 & \theta_{x_2,y_2,z_{y_2}}^2 & \gamma\theta_{0,y_1,1}^2 & & \\ \hline & \gamma & & 1 & 1 & 1 \\ & \gamma\theta_{1,y_1,0} & & \theta_{0,y_1,1} & \theta_{1,y_1,1} & \theta_{x_2,y_2,z_{y_2}} \\ & \gamma\theta_{1,y_1,0}^2 & & \theta_{0,y_1,1}^2 & \theta_{1,y_1,1}^2 & \theta_{x_2,y_2,z_{y_2}}^2 \end{array}\right]$$

- Let $v = (f_0, f_1, f_2, g_0, g_1, g_2)^T$ be an vector in left null space of H_S. Let,

$$f(x) = \sum_{j=0}^{2} f_j x^j \text{ and } g(x) = \sum_{j=0}^{2} g_j x^j.$$

- $vH_s = 0$ implies that

$$\begin{aligned} f(\theta_{y_1}) = f(\theta_{x_2,y_2,z_{y_2}}) = g(\theta_{y_1}) = g(\theta_{x_2,y_2,z_{y_2}}) &= 0 \text{ where } \theta_{0,y_1,0} = \theta_{1,y_1,1} = \theta_{y_1} \\ f(\theta_{1,y_1,0}) + \gamma g(\theta_{1,y_1,0}) &= 0, \quad \gamma f(\theta_{0,y_1,1}) + g(\theta_{0,y_1,1}) = 0 \end{aligned}$$

- Substituting $f(x) = f_2(x - \theta_{y_1})(x - \theta_{x_2,y_2,z_{y_2}})$ and $g(x) = g_2(x - \theta_{y_1})(x - \theta_{x_2,y_2,z_{y_2}})$ we get

$$\begin{bmatrix} 1 & \gamma \\ \gamma & 1 \end{bmatrix} \begin{bmatrix} f_2 \\ g_2 \end{bmatrix} = 0 \implies f_2 = g_2 = 0 \implies f = g = 0$$

MSR Codes: Summary

- Clay code construction for $d = n - 1$
- Systems implementation and evaluation of Clay codes over Ceph.
- Small d constructions for $d \in \{k + 1, k + 2, k + 3\}$.
- Our follow up work proves that the parity check support seen in the constructions is forced due to the optimal access, optimal sub-packetization properties of the codes.

Codes for Private Information Retrieval

Joint Work with Vinayak Ramkumar

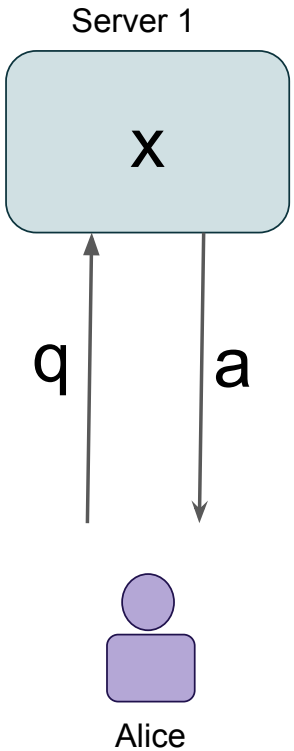
PIR: Motivation

- User accessing some critical/confidential data.

- How to retrieve information from a database without allowing the server to learn the identity of retrieved item ?
  - Example: Financial Records (Stock Prices)

Image courtesy: https://pxhere.com/en/photo/1435307

PIR: Single Server



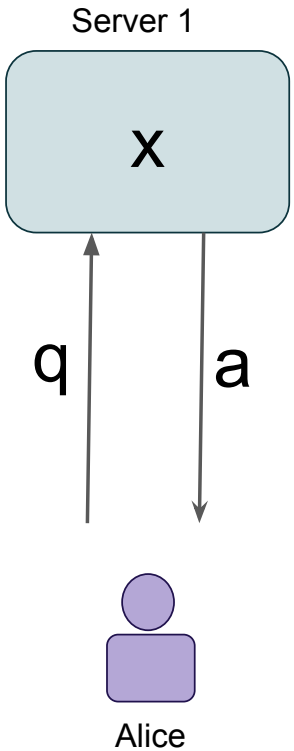
- Alice wants x_j from database $\mathbf{x}$ (of size B bits) without revealing any information to server about the index j.
- Privacy constraint : Information theoretic

$$\underbrace{I(Q;J)=0}_{\text{privacy}}, \qquad \underbrace{H(x_J|A)}_{\text{information retrieval}} = 0$$

- PIR protocol is comprised of (Q, A, R), query, answer and reconstruction functions.

- Performance metric : Communication complexity
  - Number of bits communicated through query and answers to achieve PIR

PIR: Single Server



- Alice wants x_j from database **x** (of size B bits) without revealing any information to server about the index j.
- Privacy constraint : Information theoretic

$$\underbrace{I(Q;J)=0}_{\text{privacy}}, \qquad \underbrace{H(x_J|A)}_{\text{information retrieval}} = 0$$

- PIR protocol is comprised of (Q, A, R), query, answer and reconstruction functions.

- Performance metric : Communication complexity
  - Number of bits communicated through query and answers to achieve PIR
- It was shown in [1] that $\Omega(B)$ bits need to be communicated to achieve PIR using a single server.

---

[1] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," Journal of the ACM, 45, 1998

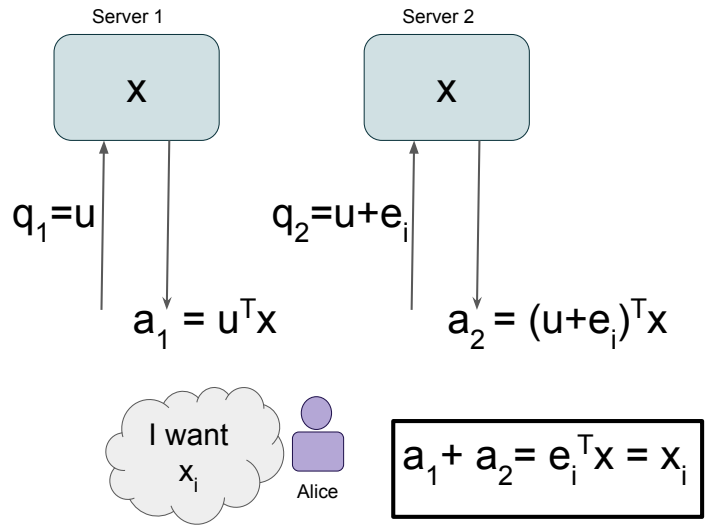
PIR: τ Non-communicating Replicated Servers

$$\underbrace{I(Q_t; J) = 0, \forall t \in [\tau],}_{\text{privacy}} \quad \underbrace{H(x_J | A_1, A_2 \cdots A_\tau) = 0}_{\text{information retrieval}}$$

PIR: τ Non-communicating Replicated Servers

$$\underbrace{I(Q_t; J) = 0, \forall t \in [\tau],}_{\text{privacy}} \quad \underbrace{H(x_J | A_1, A_2 \cdots A_\tau) = 0}_{\text{information retrieval}}$$

- An example PIR protocol with $\tau = 2$ replicated servers:

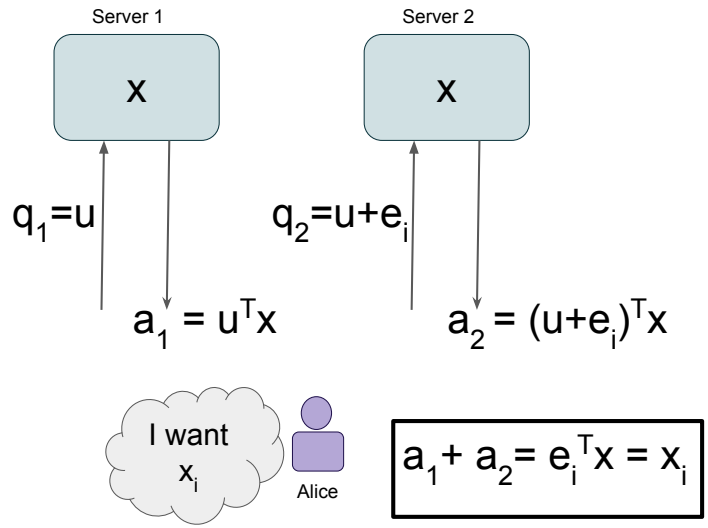


- Communication complexity $C(B, 2) = 2\text{B}{+}2 = O(B)$.

PIR: τ Non-communicating Replicated Servers

$$\underbrace{I(Q_t; J) = 0, \forall t \in [\tau],}_{\text{privacy}} \quad \underbrace{H(x_J | A_1, A_2 \cdots A_\tau) = 0}_{\text{information retrieval}}$$

- An example PIR protocol with $\tau = 2$ replicated servers:



- Communication complexity $C(B, 2) = 2\text{B}{+}2 = O(B)$.
- A 2-replicated server PIR protocol with communication complexity $O(B^{\frac{1}{3}})$ is given in [1].
- In general communication complexity $C(B, \tau)$ reduces with increase in τ.

---

[1] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," Journal of the ACM, 45, 1998

Can one do better in terms of storage overhead than replication?

Can one do better in terms of storage overhead than replication?

Yes by Coding!!

Coded PIR : History

- N. Shah, K. V. Rashmi, K. Ramchandran, "One extra bit of download ensures perfectly private information retrieval", 2014.
  - Introduced coded PIR.
  - PIR protocol specific to the code used.

- TH Chan, SW Ho, H Yamamoto, "Private information retrieval for coded storage", 2015
  - Studied trade-off between storage-overhead and communication complexity for large file sizes.

There's so much of work on replicated server protocols!! Can one use the replicated protocols over coded databases?

There's so much of work on replicated server protocols!! Can one use the replicated protocols over coded databases?

Yes!!

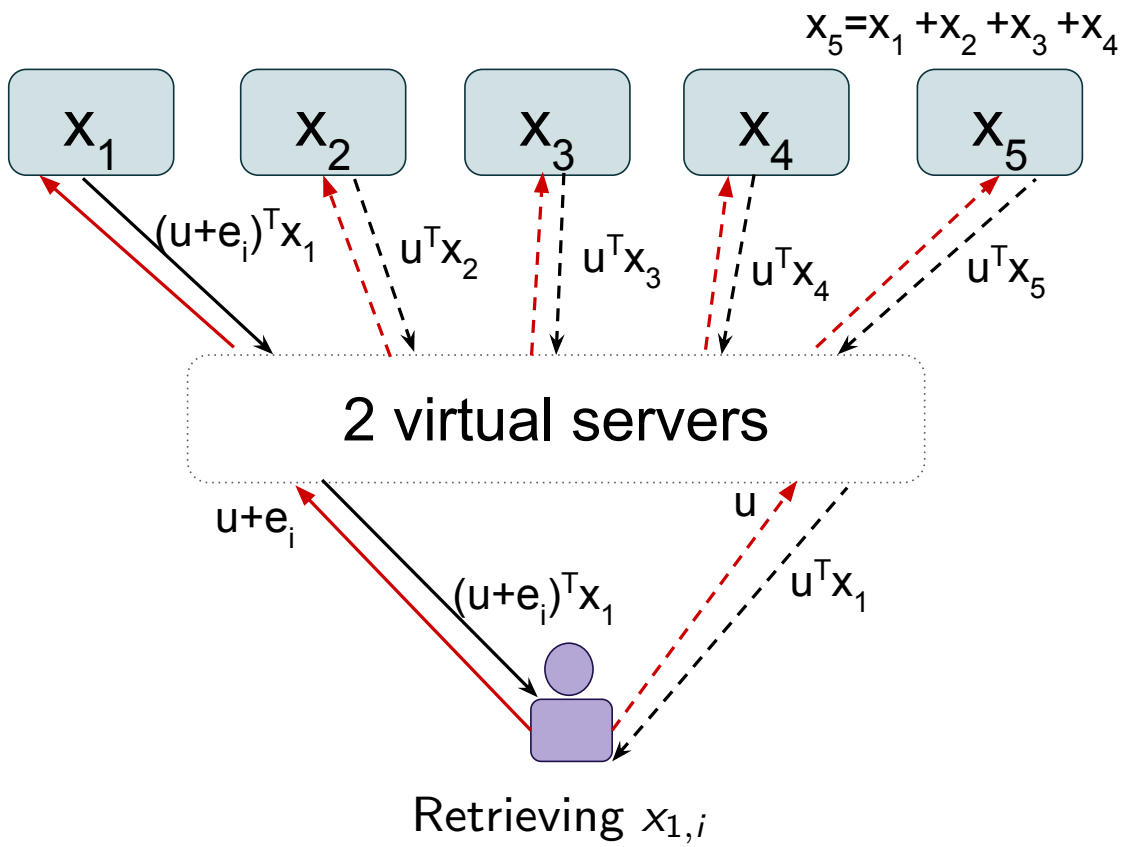
How?

PIR Code

---

[1] A. Fazeli, A. Vardy, and E. Yaakobi, "Codes for distributed PIR with low storage overhead", ISIT 2015.

Coded PIR: An Example



- $\frac{B}{4}$ bits stored in each server
- Storage overhead is 1.25
- Communication complexity is $5 * (\frac{B}{4} + 1)$

$(n, k), \tau$ - server PIR Code

- τ disjoint recovery sets for each message symbol.
- Single parity check code shown is a $\tau = 2$ PIR code.
- $(n, k), \tau$-server PIR code helps to emulate any $\tau-$replicated-server PIR protocol on coded servers.
  - This assumes the answer function A is linear in the database content $\mathbf{X}$.
  - $n * C(\frac{B}{k}, \tau)$ is the communication complexity.
- To minimize storage overhead, we want to have smallest possible n for a given k, τ.

---

[1] A. Fazeli, A. Vardy, and E. Yaakobi, "PIR with low storage overhead: Coding instead of replication," CoRR, vol. abs/1505.06241, 2015.

Reed Muller Codes & Disjoint Recovery Set Property

- (r, m) Reed Muller (RM) codes are generated by evaulations of polynomials in m variables of degree $\leq r$ in $\mathbb{F}_2^m$

$$f(x_1, \cdots, x_m) = \sum_{S \subseteq [m], |S| \leq r} a_S \prod_{i \in S} x_i, \quad a_S \in \mathbb{F}_2$$

$$n = |\mathbb{F}_2^m| = 2^m, \; k = \sum_{i=0}^{r} \binom{m}{i}.$$

- Majority Logic Decoding of a_S where $|S| = r$

$$a_S = \sum_{x_S \in F_2^r} f(x_S, b) \text{ for any } b \in \mathbb{F}_2^{m-r}$$

- Smaller degree monomial coefficients decoded sequentially

Shortened Reed Muller (SRM) Code

- A code vector in binary $SRM(r, m)$ code corresponds to evaluations of r-degree homogeneous polynomial in m binary variables at points from $\mathbb{F}_2^m$.

$$f(x_1, \cdots, x_m) = \sum_{S \subseteq [m], |S|=r} a_S \prod_{i \in S} x_i, \quad a_S \in \mathbb{F}_2$$

$$n = |\mathbb{F}_2^m| = 2^m, \; k = \binom{m}{r}.$$

- It is clear to see the above polynomials are evaluated to 0 for all $\underline{x}$ such that $w_H(\underline{x}) < r$.
- Can restrict to evaluations at $\underline{x}$ such that $w_H(\underline{x}) \geq r$.

$$n = \sum_{i=r}^{m} \binom{m}{i}, \quad k = \binom{m}{r}.$$

Shortened Reed Muller (SRM) code for PIR

- SRM$(2,4)$: $r = 2, m = 4$
  - Any code vector corresponds to the evaluation of polynomials of form

$$f(\underline{x}) = a_{12}x_1x_2 + a_{13}x_1x_3 + a_{14}x_1x_4 + a_{23}x_2x_3 + a_{24}x_2x_4 + a_{34}x_3x_4$$

    of degree 2 in 4 variables at points $\underline{x} = (x_1, x_2, x_3, x_4)$ such that $w_H(\underline{x}) \geq 2$.

  - Message symbol recovery (Majority Logic Decoding)

$$\begin{aligned} a_{12} &= \sum_{x_1,x_2} f(x_1x_2b_3b_4) \\ &= f(1100) \\ &= f(0110) + f(1010) + f(1110) \\ &= f(0101) + f(1001) + f(1101) \\ &= f(0011) + f(0111) + f(1011) + f(1111). \end{aligned}$$

  - This gives $(n = 11,\ k = 6),\ (\tau = 4)$-server systematic PIR code.

Shortened Reed Muller (SRM) code for PIR

- SRM$(2,4)$: $r = 2, m = 4$
  - Any code vector corresponds to the evaluation of polynomials of form

$$f(\underline{x}) = a_{12}x_1x_2 + a_{13}x_1x_3 + a_{14}x_1x_4 + a_{23}x_2x_3 + a_{24}x_2x_4 + a_{34}x_3x_4$$

    of degree 2 in 4 variables at points $\underline{x} = (x_1, x_2, x_3, x_4)$ such that $w_H(\underline{x}) \geq 2$.

  - Message symbol recovery (Majority Logic Decoding)

$$\begin{aligned} a_{12} &= \sum_{x_1,x_2} f(x_1x_2b_3b_4) \\ &= f(1100) \\ &= f(0110) + f(1010) + f(1110) \\ &= f(0101) + f(1001) + f(1101) \\ &= f(0011) + f(0111) + f(1011) + f(1111). \end{aligned}$$

  - This gives $(n = 11,\ k = 6),\ (\tau = 4)$-server systematic PIR code.

Theorem

SRM(r, m) code is a $(n = \sum_{i=r}^{m} \binom{m}{i},\ k = \binom{m}{r}), (\tau = 2^{m-r})$-server PIR code.

PIR Code for any k

- For any $k < \binom{m}{r}$, set $\gamma = \binom{m}{r} - k$.
- Shorten SRM(m, r) by setting γ message symbols to 0. This retains τ
- Use the Support-Set Viewpoint of SRM Code to pick the γ message symbols carefully
- $f(\underline{x}) \equiv f(\text{Supp}(\underline{x}))$
- Each code symbol can be indexed by a subset of $[m]$ with size $\geq r$.
- Each message symbol (& its corresponding monomial) can be indexed by a subset of $[m]$ with size r.

$$f(S) \quad = \quad \sum_{R_i \subseteq S} f(R_i), \quad \text{where } R_i \subseteq [m] \text{ with } |R_i| = r.$$

- For example SRM$(2, 4)$ code has, $f(\{1, 3, 4\}) = f(\{1, 3\}) + f(\{1, 4\}) + f(\{3, 4\})$.

Shortening SRM Code

- Assume that we set $a_{R_i} = f(R_i) = 0$, $\forall R_i \subseteq S$.

$$f(S) \quad = \quad \sum_{R_i \subseteq S} f(R_i), \quad \text{where } R_i \subseteq [m] \text{ with} |R_i| = r.$$

- Then, $f(S) = \sum_{R_i \subseteq S} f(R_i) = 0$ in all codewords.

Key Idea

If we shorten the SRM(r,m) code by setting all the message symbols corresponding to r-element subsets of a fixed set S to zero, then we can delete the code coordinate corresponding to S.

Shortened SRM Code

- γ message symbols of SRM(r, m) code set to zero.
- $\Gamma(r, m, \gamma)$: Block length reduction when γ message symbols are set to zero.
- Clearly, $\Gamma(r, m, \gamma) \geq \gamma$.

$$\begin{aligned} \text{Dimension } k &= \binom{m}{r} - \gamma, \\ \text{Block Length } n &= \sum_{i=r}^{m} \binom{m}{i} - \Gamma(r, m, \gamma). \end{aligned}$$

- How to maximize $\Gamma(r, m, \gamma)$ for a given $r,\ m,\ \gamma$?
- Aim: Algorithm to carefully pick γ message symbols to set to zero so that $\Gamma(r, m, \gamma)$ is large.

Shortening Algorithm: Example

- r=2, m=5.

k	γ	$\mathbb{S}$	$\Gamma(2,5,\gamma)$	n
10	0	ϕ	0	26
9	1	$\{1,2\}$	1	25
8	2	$\{1,3\}$	2	24
7	3	$\{2,3\}$	4	22
6	4	$\{1,4\}$	5	21
5	5	$\{2,4\}$	7	19
4	6	$\{3,4\}$	11	15
3	7	$\{1,5\}$	12	14
2	8	$\{2,5\}$	14	12
1	9	$\{3,5\}$	18	8
0	10	$\{4,5\}$	26	0

- Yellow: code-symbol corresponding to 3-element set $\{1,2,3\}$ also becomes 0.
  - results in reduction of $1+3=4$
- Green: code-symbol corresponding to 4-element set $\{1,2,3,4\}$ also becomes 0.
  - results in reduction of $1+\binom{4}{3}+\binom{4}{2}=11$
- Red: code-symbol corresponding to 5-element set $\{1,2,3,4,5\}$ also become 0.

Order of picking

- The order in which the r-element subsets are picked here is called co-lexicographic order, where a set $A > B$ iff $\max(A \Delta B) \in A$.
  - $\{1,2\}$, $\{1,3\}$, $\{2,3\}$, $\{1,4\}$, $\{2,4\}$, $\{3,4\}$, $\{1,5\}$, $\{2,5\}$, $\{3,5\}$, $\{4,5\}$ are in co-lexicographic order.

- Shortening algorithm: The $\binom{m}{r}$ r-element sets are arranged in co-lexicographic order and first γ sets are picked as message symbols that are set to 0.

Obtaining $\Gamma(r, m, \gamma)$ from γ

1. Unique $\underline{\rho}$ representation of γ
   - $\underline{\rho}$ is of length $(m - r)$ and has weight $\leq r$.

2. $\Gamma(r, m, \gamma)$ is computed from $\underline{\rho}$

$$\gamma \implies \underline{\rho} \implies \Gamma(r, m, \gamma)$$

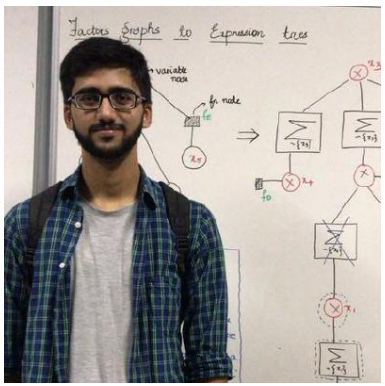
M. Vajha, V. Ramkumar, P. V. Kumar, "Binary, Shortened Projective Reed Muller Codes for Coded Private Information Retrieval", ISIT 2017

Summary

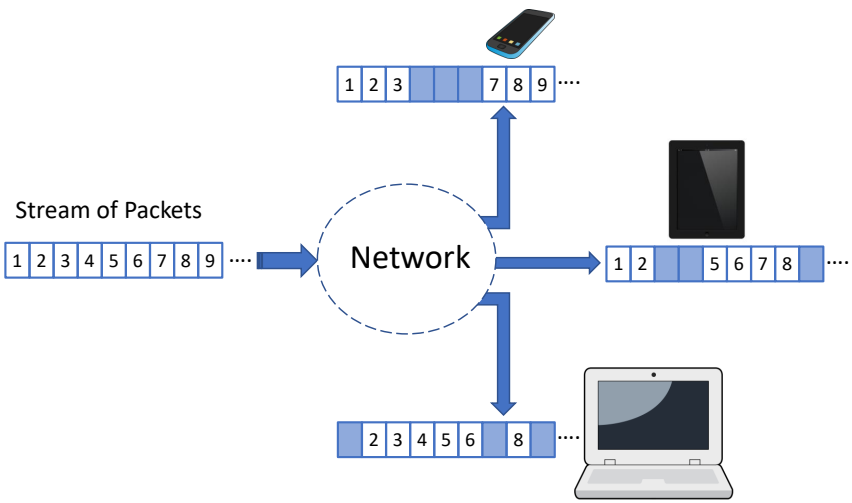
- PIR code constructions
  - Binary Shortened Reed Muller(SRM) code as PIR code.
  - Shortening Algorithm to get PIR codes for τ of form $2^\ell, 2^\ell - 1$ and any k.
    - ★ Upper bounds on GHW of binary SRM codes.
    - ★ We prove that this shortening algorithm is optimal by showing a lower bound resulting in Complete weight hierarchy of SRM codes
- These constructions have smaller block length in comparison to known codes in literature for small k.
- Optimality
  - One bit improvement from Rao-Vardy lower bound for $n(k, \tau)$ assuming systematic PIR code.
  - Optimal systematic PIR codes for $\tau = 3, 4$.
  - These codes are optimal batch codes for $\tau = 3, 4$ as well.

Streaming Codes for Low Latency

Joint work with Nikhil Krishnan, Vinayak Ramkumar and Mayank Jhamtani



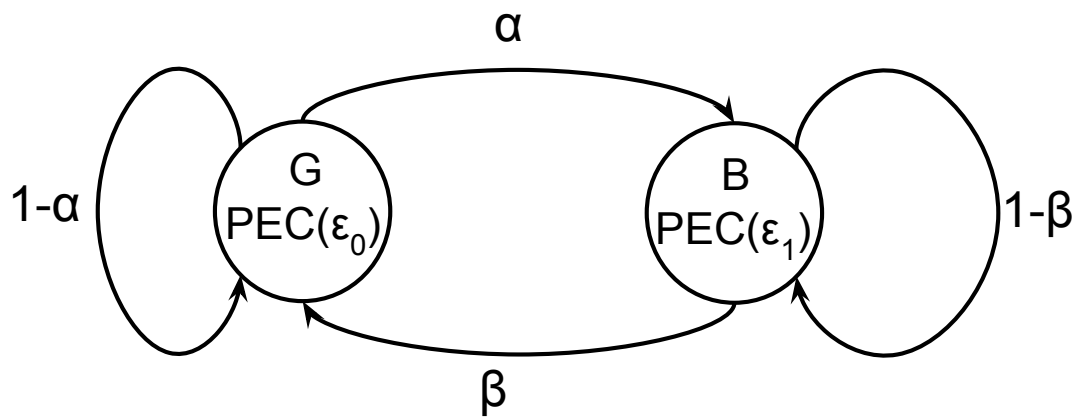
Setting: Packet Erasures Over a Network



- Packet erasures caused by dropped packets, slow fading, etc . . .
- Strict decoding delay constraint of τ
- Goal: Communication with low latency, high throughput, high reliability

- ARQ scheme results in large delays.
- FECs are therefore preferred method for interactive voice-video (like Skype, Hangouts, Facetime, Zoom). FECs used in practise are either designed for burst alone or random erasures alone.
- Need FECs that can guarantee small delay along with burst and random erasure correctability.

Gilbert-Elliott (GE) Channel Model



- $\epsilon_0 << \epsilon_1$
- Commonly-accepted channel model
- But less tractable: harder to design codes for this channel to ensure desired reliability levels

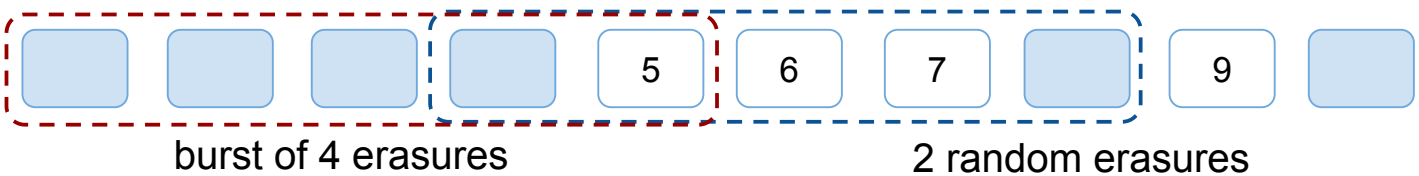
Delay-Constrained Sliding-Window (DC-SW) Channel Model

Channel model characterized by:

(i) Admissible erasure patterns (AEP): within a sliding window of size w:

either $\leq a$ random erasures, or a burst of $\leq b$ erasures

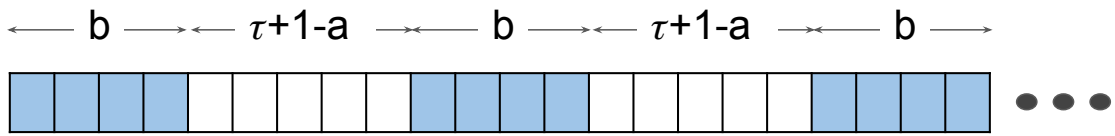
(ii) Decoding-Delay Parameter: τ



$(a = 2, b = 4, w = 5, \tau = 4)$

Upper Bound on Code Rate

- A code $\mathcal{C}$ is said to be streaming code if it can correct all AEPs of the DC-SW channel with decoding delay $\leq \tau$.
- We can assume WOLOG that $w = \tau + 1$.
- Shown below is an admissible erasure pattern.

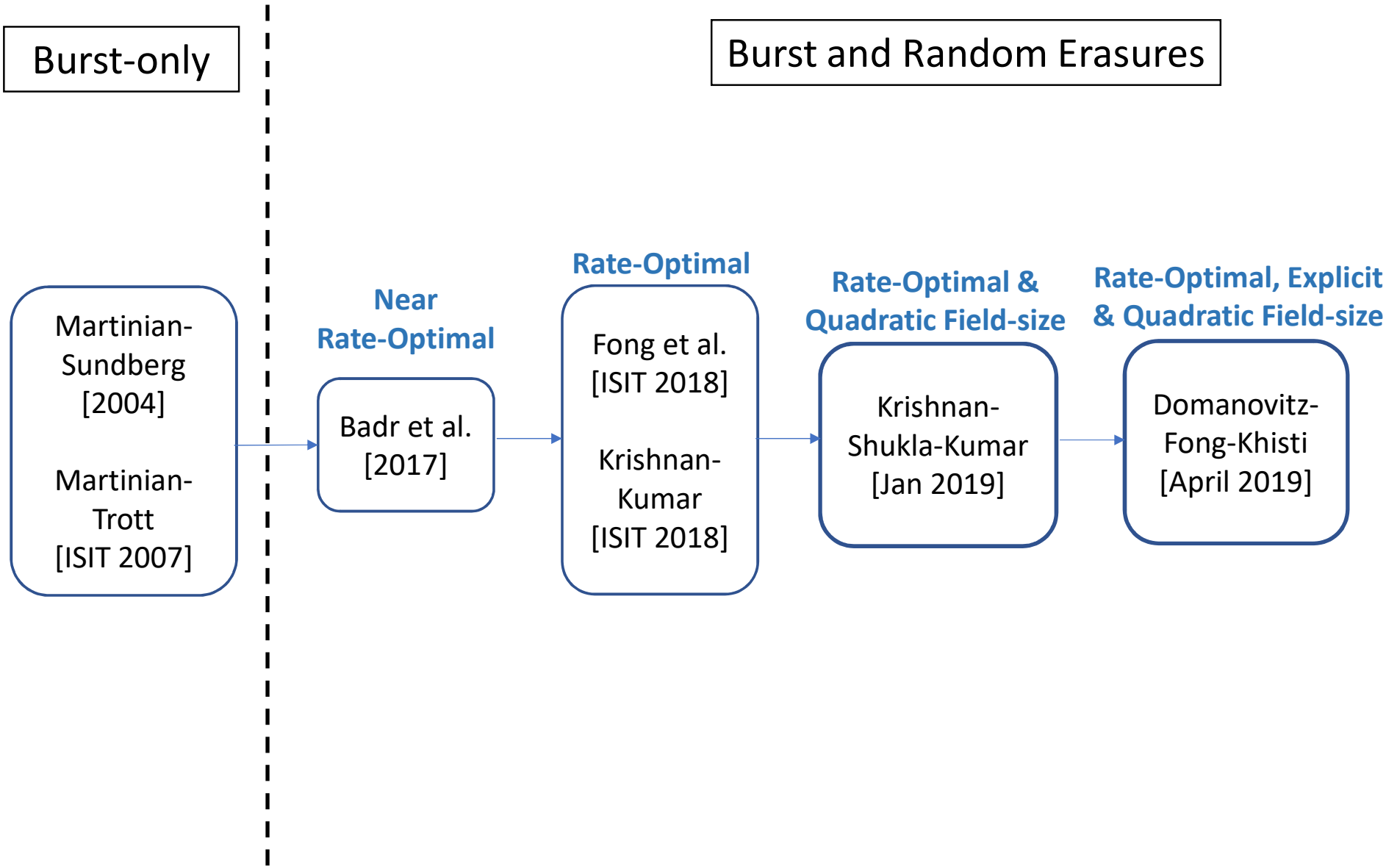


- It follows that the rate R of (a, b, τ) streaming code $\mathcal{C}$ satisfies:

$$R \leq \frac{(\tau + 1) - a}{(\tau + 1) + b - a} \triangleq R_{\text{opt}}.$$

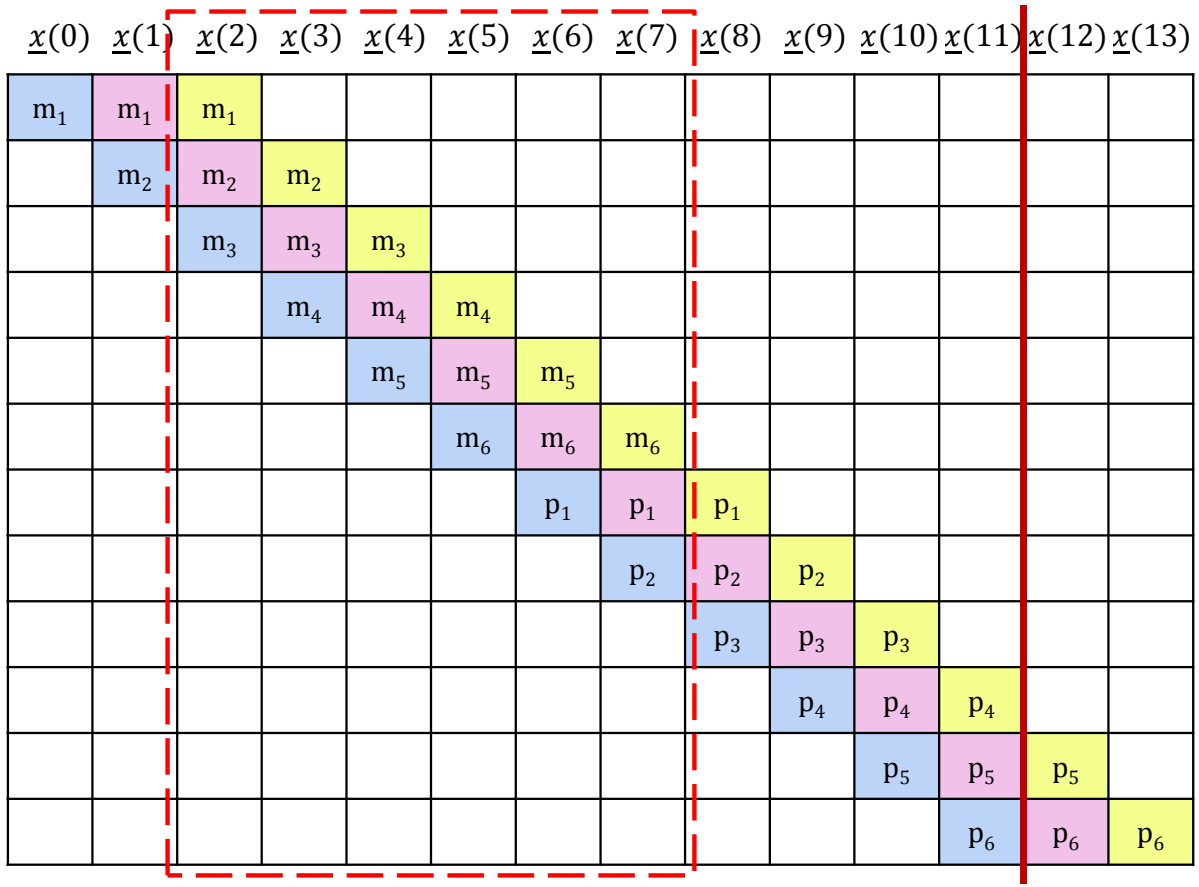
Badr et al., "Layered Constructions for Low-Delay Streaming Codes," *Trans. IT*, 2017.

Prior Work on Streaming Codes



Diagonal Embedding (DE)

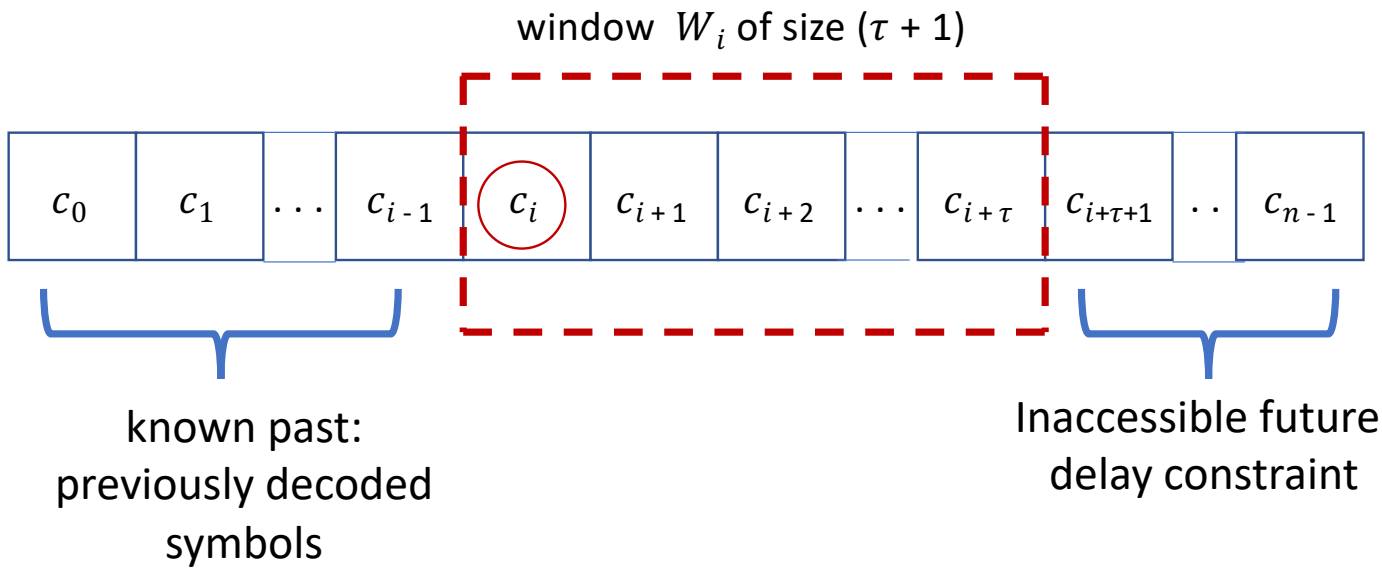
- Codewords of $[n, k]$ scalar block code are diagonally placed in the packet stream.
- This approach results in a rate-optimal $\{a, b, \tau\}$ streaming code only if:
  - $n - k \geq b$
  - $n \geq \tau + 1 - a + b$
- Clearly, MDS code won't work



DE of $[12, 6]$ scalar code where $a = 4, b = 6, \tau = 9$

Diagonal Embedding (DE)

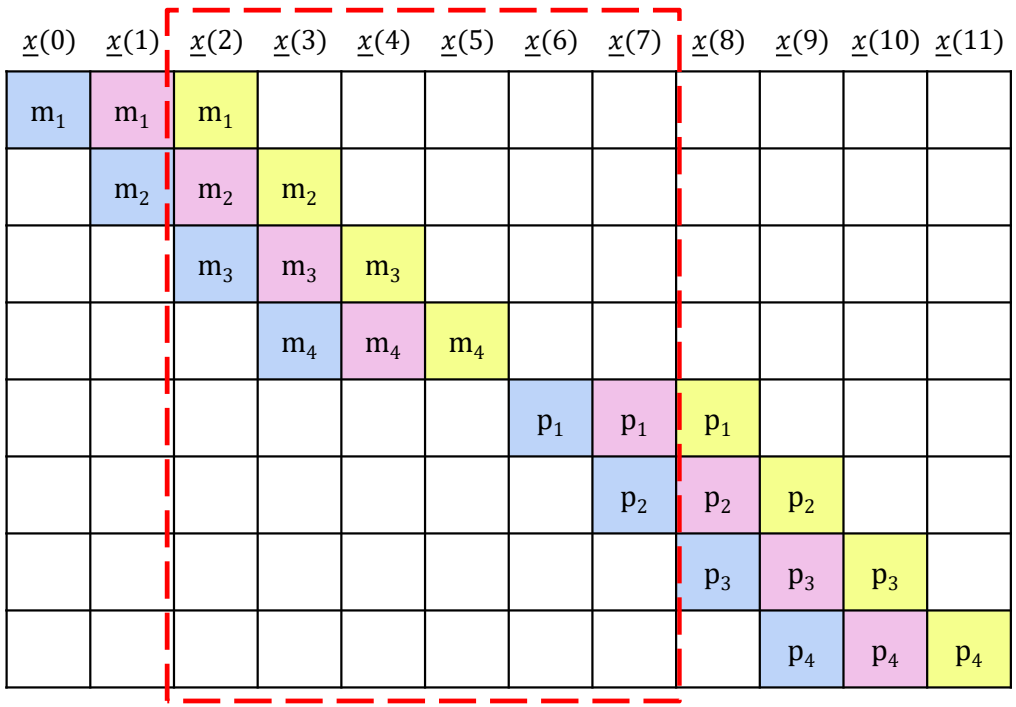
- Since the block length $n = (\tau + 1 + b - a)$ exceeds the delay constraint, partial-knowledge decoding is needed:



- Best known explicit construction requires field-size $> (\tau + b - a)^2$

Staggered Diagonal Embedding(SDE)

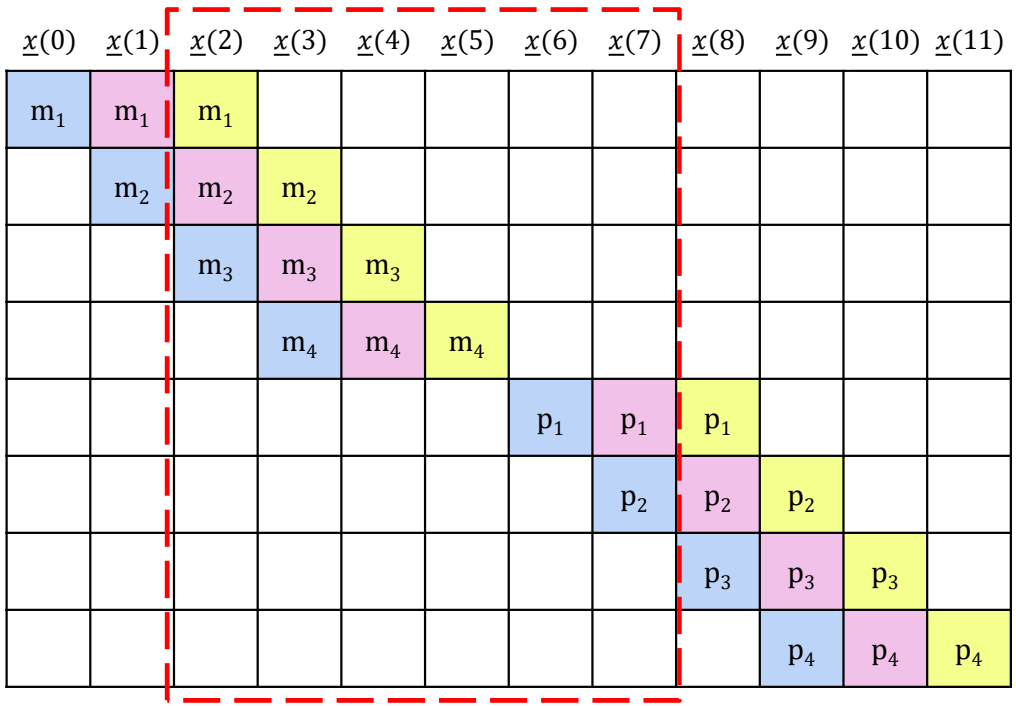
- Codewords of an $[n, k]$ scalar block code are embedded diagonally with gaps in the packet stream as shown below.
- This approach lowers the needed number of parities and block length.
  - $n - k \geq a$
  - $n \geq \frac{a}{b}(\tau + b - a + 1)$ for rate-optimal streaming code



SDE of $[8, 4]$ scalar code where $a = 4, b = 6, \tau = 9$

Staggered Diagonal Embedding(SDE)

- Let N be the span of codewords of the $[n, k]$ scalar block code.
- Setting $N \leq \tau + 1$, ensures that no partial knowledge decoding is needed..



SDE of $[8, 4]$ scalar code where $N = 10, \tau = 9$

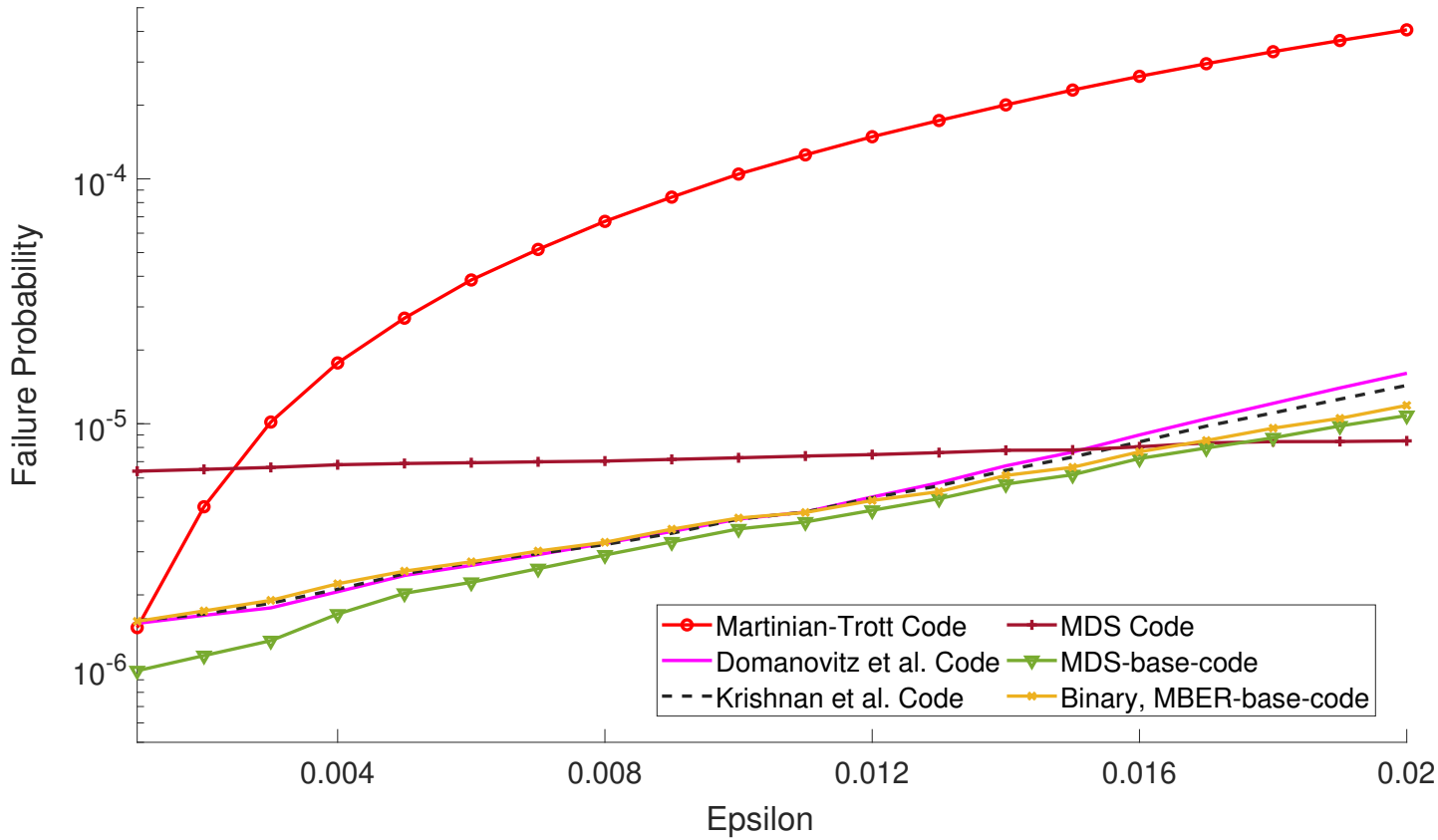
Simple Streaming Codes

- $N \leq \tau + 1$
- SDE of MDS code with a parities where a consecutive symbols are followed by $b - a$ gaps.
- Block length n is given by:

$$n = ma + \min\{\delta_1, a\} \text{ where } \tau + 1 = mb + \delta_1.$$

- These codes are rate optimal when $b|(\tau + 1 - a)$ and have rate of form $R = \frac{m}{m+1}$.

Simulation over GE channel with $\alpha = 10^{-4}, \beta = 0.6$



- All the codes are rate 0.5 and have decoding delay constraint $\tau = 9$.

Streaming Codes Summary and Followup Work

- For $N \leq \tau + 1$ we have provided constructions that are optimal within this setting. We also prove that the only optimal streaming code under this setting is when $b | \tau + 1 - a$.

- In a follow up, we were able to generalize this construction to the case when $\gcd(\tau + 1 - a, b) \geq a$ by relaxing the $N \leq \tau + 1$ restriction to $N < \tau + 1 + b - a$.

- To provide an explicit streaming code construction with field size of τ^2 for any (a, b, τ).

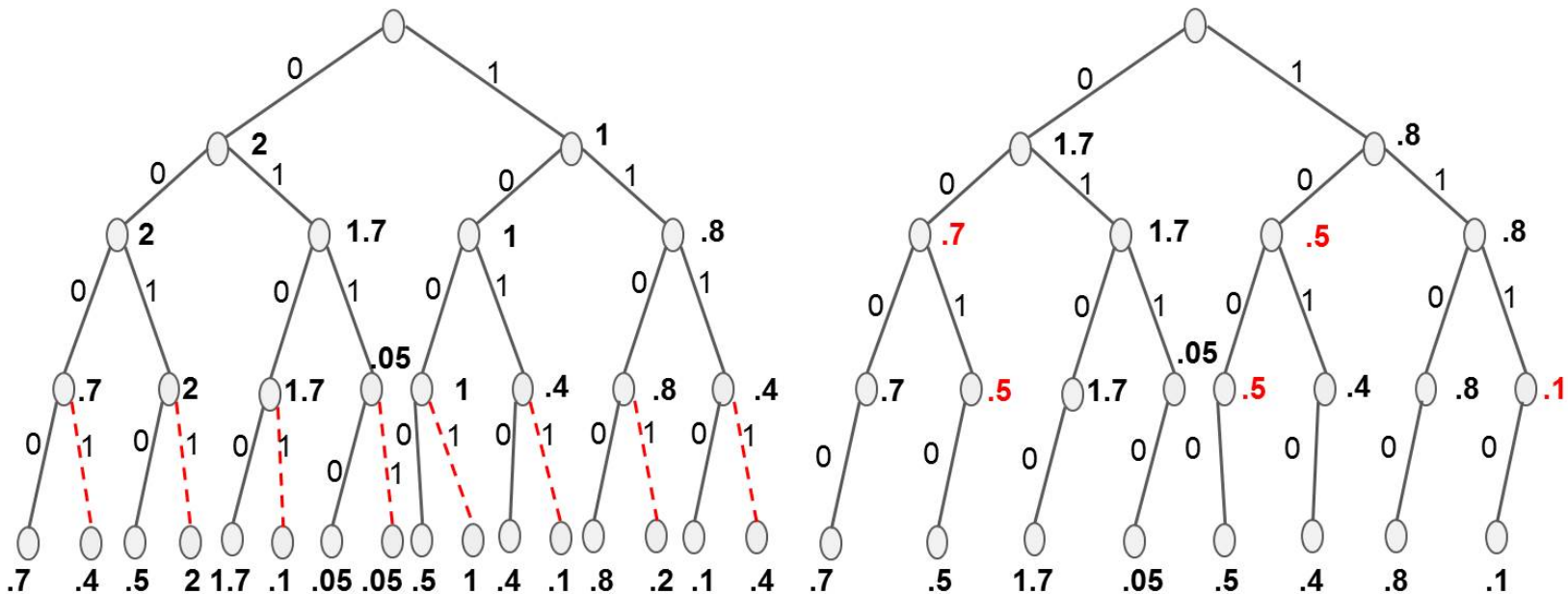
- Analysing the approximation of sliding window channel to GE channel, we were able to give close upper and lower bounds to the probability of admissible erasure patterns of an DCSW channel over GE channel.
  - This helps in choosing a streaming code with performance guarantees over GE channel.

- Design and implementation of an adaptive video streaming application.

Other Results: Decoding Algorithms for Polar Codes

- We introduced Successive Cancellation Backtracking (SCBT) and Successive Cancellation Look Ahead (SCLA) decoding of Polar codes that improve upon Successive Cancellation (SC) decoding while using $O(N)$ memory.

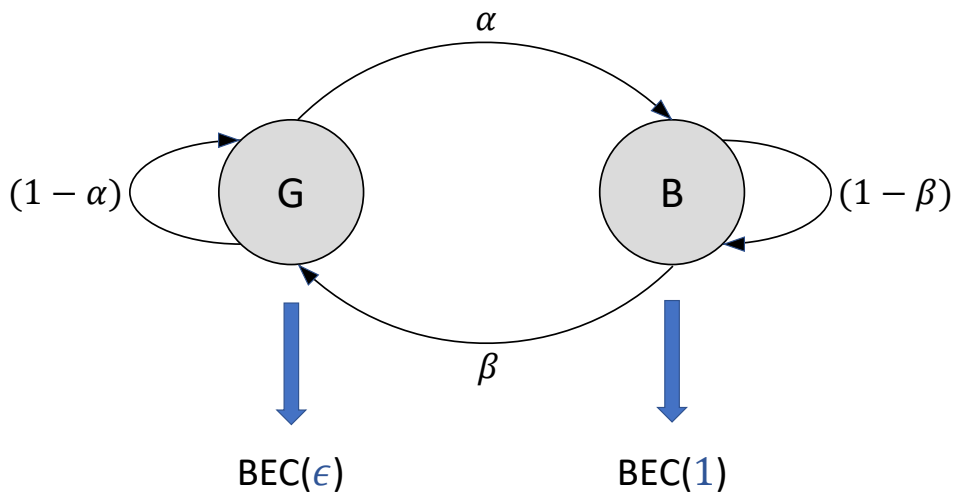
- These algorithms are a result of an observation, that the decoding of polar codes is a binary search over tree with noisy metric information.



UA Metric vs Actual Metric (under Max Metric)

Video Demo Setting

- The Channels:
  - C0: perfect channel (no erasures)
  - C1: GE ($\alpha = 0.01, \beta = 0.5, \epsilon = 0.001$)



- τ is set to 9
- Video1: We set $(a = 0, b = 0)$ at the start of experiment and move from C0 to C1
- Video2: After the adaptation converges

Acknowledgements

My heartfelt thanks to

- PVK and Sudha Maam
- My Thesis reviewers Prasad Krishnan and Xiaohu Tang
- Professors Navin Kashyap, Himanshu Tyagi and ECE faculty in general
- Professors at Math, CSA departments
- Srinivasa Murthy
- Labmates Nikhil, Birenjith, Lalitha, Vinayak, Balaji, Tarun, Bhagya, Ganesh, Pallavi, Mayank, Elita, Chaitanya, Deeptanshu
- ECEmates Vinay B R and Sahasranand for awesome time spent organising student seminar series
- My friends Krishna, Priya and Bala who are always up for a conversation over coffee
- My friends Jolie, Shravya, Karthik, Saraswati, Gouthami and Sawan for constantly inspiring me
- My parents, grandparents, Chinnu, Shilpakka and Vinay for the unconditional love.

Publications and Preprints

Codes for distributed storage

- M. Vajha, B. S. Babu and P. V. Kumar, "Explicit MSR Codes with Optimal Access, Optimal Sub-Packetization and Small Field Size for d=k+1, k+2, k+3", ISIT 2018.

- M. Vajha, V. Ramkumar, B. Puranik, G. Kini, E. Lobo, B. Sasidharan, P. V. Kumar, A. Barg, M. Ye, S. Narayanamurthy, S. Hussain, and S. Nandi, "Clay Codes: Moulding MDS Codes to Yield an MSR Code", in USENIX, Files and Storage Technologies, FAST 2018.

- S. B. Balaji, M. N. Krishnan, M. Vajha, V. Ramkumar, B. Sasidharan and P. V. Kumar, "Erasure coding for distributed storage: an overview", Science China Information Sciences, 2018.

- B. Sasidharan, M. Vajha and P. V. Kumar, "An explicit, coupled-layer construction of a high-rate MSR code with low sub-packetization level, small field size and all-node repair", arXiv preprint arXiv:1607.07335, 2016

- B. Sasidharan, N. Prakash, M. N. Krishnan, M. Vajha, K. Senthoor and P. V. Kumar, "Outer bounds on the storage-repair bandwidth trade-off of exact-repair regenerating codes", in International Journal of Information and Coding Theory, 2016.

Publications and Preprints

Codes for Private Information Retrieval

- M. Vajha, V. Ramkumar, and P. V. Kumar, "Binary, shortened projective reed muller codes for coded private information retrieval", in IEEE International Symposium on Information Theory, ISIT, 2017.

- V. Ramkumar, M. Vajha, and P. V. Kumar, "Determining the Generalized Hamming Weight Hierarchy of the Binary Projective Reed-Muller Code", in National Conference on Communications, NCC, 2018.

Polar Codes

- Myna Vajha, V. S. Chaitanya Mukka and P Vijay Kumar, "Backtracking and Look-Ahead Decoding Algorithms for Improved Successive Cancellation Decoding Performance of Polar Codes", ISIT 2019.

Publications and Preprints

Low Latency Streaming

- M. N. Krishnan*, V. Ramkumar*, M. Vajha*, P. V. Kumar*, "Simple Streaming Codes for Reliable, Low-Latency Communication", IEEE Communications Letters, 2019
- Vinayak Ramkumar*, Myna Vajha*, M. Nikhil Krishnan*, P. Vijay Kumar*, "Staggered Diagonal Embedding Based Linear Field Size Streaming Codes", ISIT, 2020.
- Myna Vajha, Vinayak Ramkumar, Mayank Jhamtani, P Vijay Kumar, " On Sliding Window Approximation of Gilbert-Elliott Channel for Delay Constrained Setting", arxiv, 2020.

* indicates equal contribution

Thanks!

Outer Bounds for Exact Repair Regenerating Codes

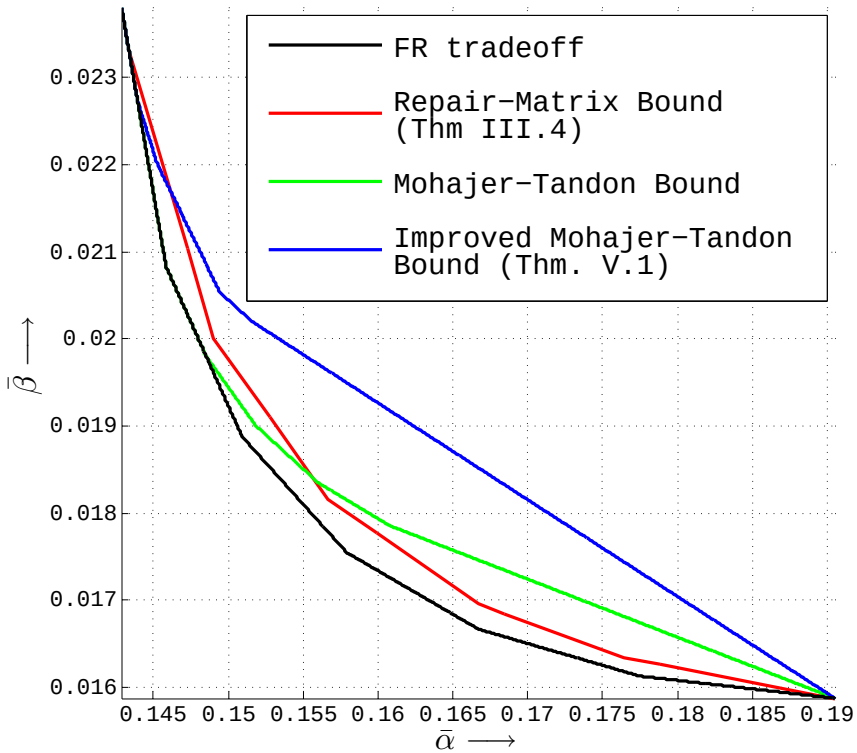


Figure: $(n = 13, k = 7, d = 12)$.

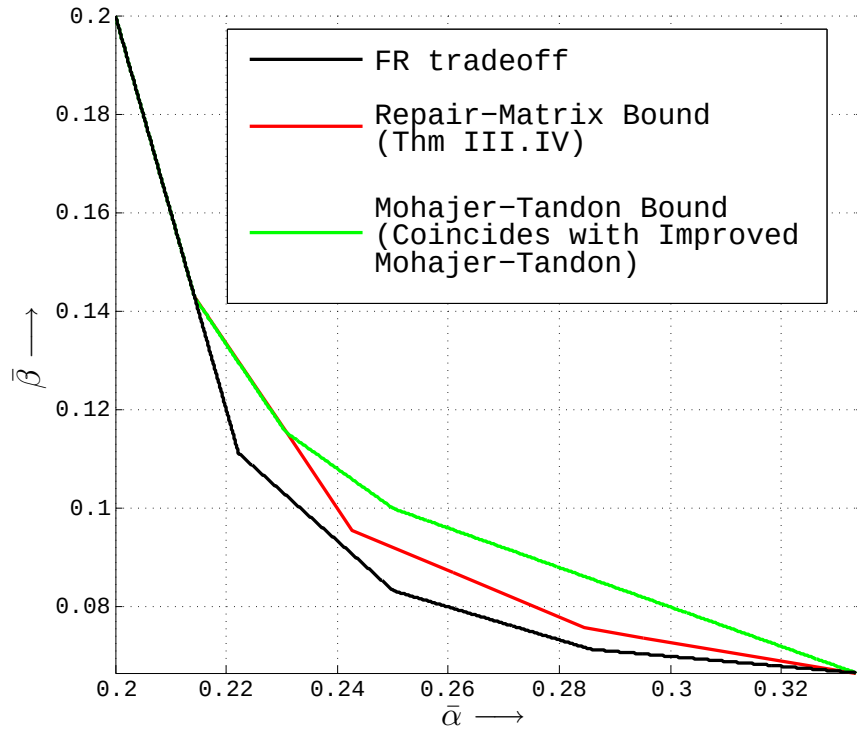


Figure: $(n = 6, k = 5, d = 5)$.

- Improved outer bounds for the case of $d > k$.

- B. Sasidharan, N. Prakash, M. Nikhil Krishnan, M Vajha, K Senthoor, P V Kumar: Outer bounds on the storage-repair bandwidth trade-off of exact-repair regenerating codes, IJICoT 3(4): 255-298 (2016).

Updates in the Thesis

- We have added a description to the introduction that provides perspective on the four areas considered in the thesis.
- We have also added a conclusion.

Fragmented Read

Best and worst case, disk read during repair of (20,16,19) code for stripe sizes 1MB, 64MB

- During repair of a chunk only $\beta < \alpha$ sub-chunks are read from each helper nodes.
- During worst case failures, the sub-chunks needed in repair are not located contiguously.
- sub-chunk size $=$ stripe size$/k\alpha$
- For (20,16,19) code $\alpha = 1024$, $k = 16$. Therefore, for stripe sizes 64MB and 1MB, the sub-chunk sizes are 4KB, 64B

- W1 has 448GB with each node storing 28GB.
- W2 has 512GB data with each node storing 32GB.
- Theoretical reads for W1 and W2 should be 7GB and 8 GB respectively for Clay code.
- In the case of RS codes, the disk reads are 28GB and 32GB for W1 and W2 respectively.

Binary MSR Constructions

- $n\alpha$ uncoupled code symbols in this case belong to $\mathcal{Q} = \mathbb{F}_2^m$.
- For any $\underline{z} \in \mathbb{Z}_s^t$. The n symbols $\{U(x, y; \underline{z}) \mid x \in \mathbb{Z}_s, y \in \mathbb{Z}_t\}$ are a codeword of $[n, k]$ MDS code $\mathcal{C}_{\text{MDS}}$ defined over alphabet $\mathcal{Q}$.
- For any (x, y, z) such that $z_y \neq x$

  $$(U(x, y, \underline{z}), U(x, y, \underline{z}(y, x)), C(x, y, \underline{z}), C(x, y, \underline{z}(y, x)))$$

  is codeword of $[4, 2]$ MDS code, $\mathcal{C}_{\text{coup}}$ defined over alphabet $\mathcal{Q}$.
- Node repair, decode, encode operations use the decode, encode operations of binary MDS codes $\mathcal{C}_{\text{MDS}}$ and $\mathcal{C}_{\text{coup}}$. Therefore all the operations are XOR based.

PIR Capacity: Retrieving files under Replicated server setting

- Here the servers store m files (each of size s bits) and Alice wants to retrieve a file without revealing any information about file index to the servers.

- Two possible ways

  1. Efficiently amortize bit PIR protocols.

     - ⋆ 2-server file PIR can be achieved with communication complexity of $2m + 2s$ bits. [1]

  2. Develop PIR protocols specifically for large files.

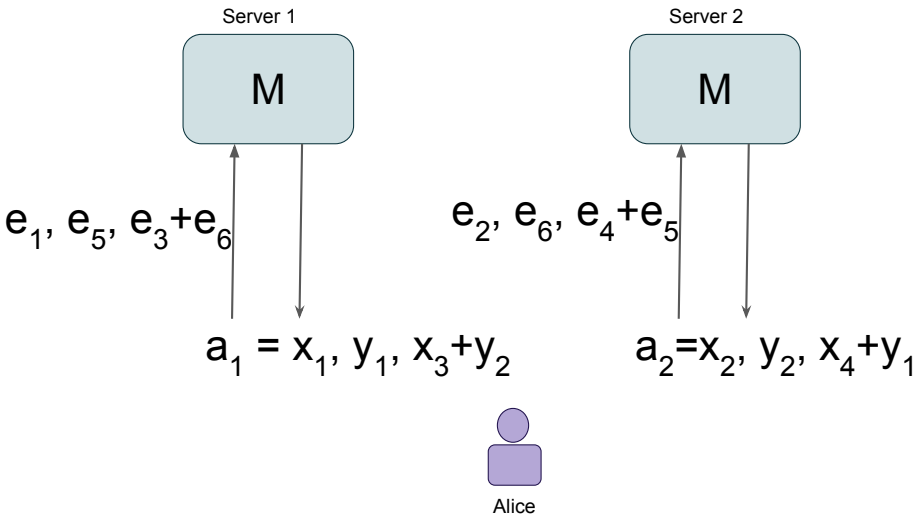
[1] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, 1998

PIR over Files : Replicated server setting

- Consider $m = 2$ files X, Y, each of size s $= 4$GB, represented as $8 \times 1G$ matrix:

$$M = \left[\begin{array}{cccc|cccc} X_1 & X_2 & X_3 & X_4 & Y_1 & Y_2 & Y_3 & Y_4 \end{array} \right]^T$$

- To retrieve file X from 2-replicated servers:



By viewing a_1 or a_2, one cannot figure out which of the 2 files is being requested.

Upload complexity is $3 \times 8 \times 2 = 48$ whereas the download is 6GB.

$48 + 1.5s < 4 + 2s$

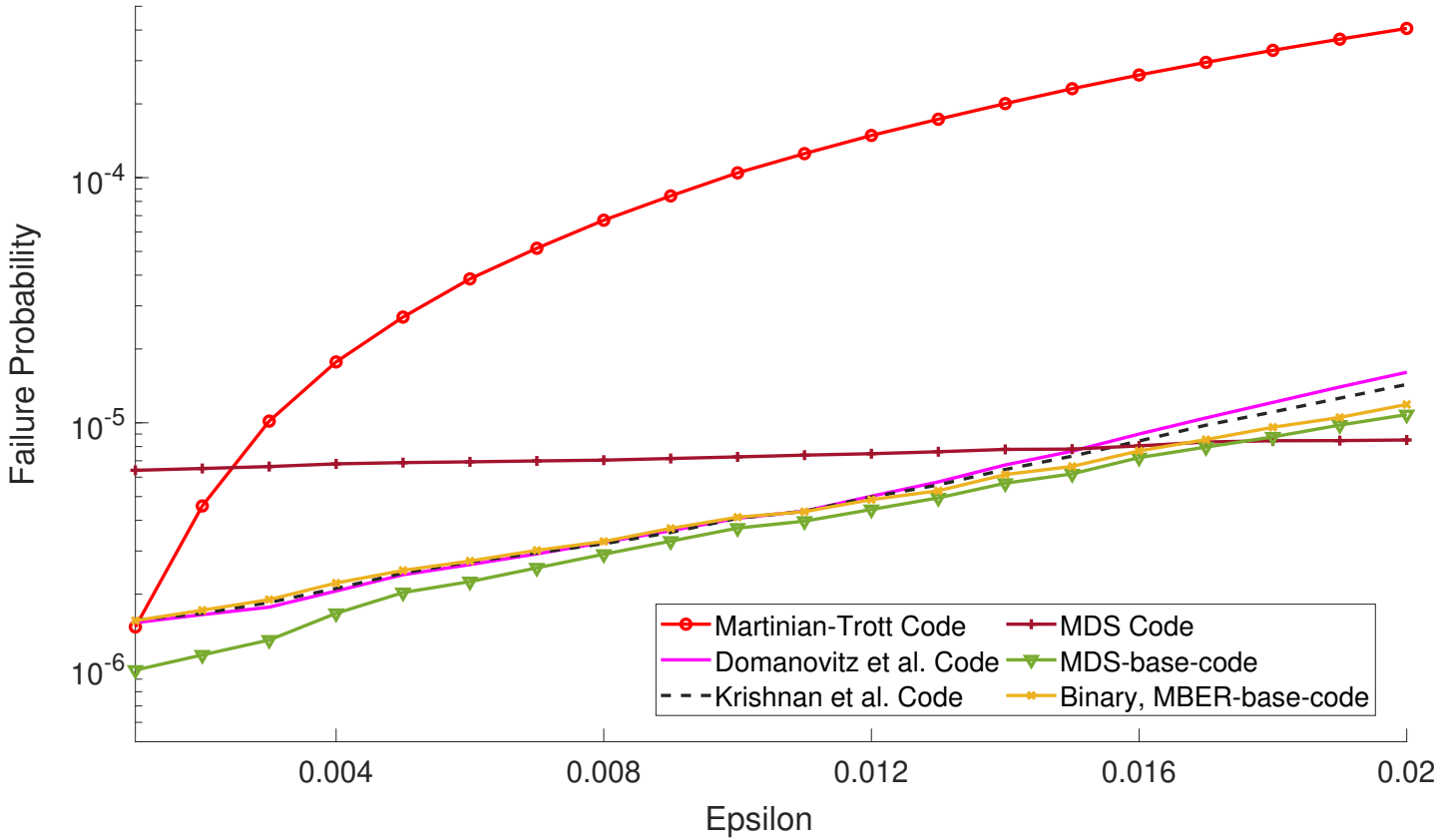
- Tradeoff upload cost to gain on download cost!!!
- Measure of performance of PIR protocol is given by communication Price of Privacy (cPoP)

$$cPoP = \frac{\#\text{ bits downloaded}}{\#\text{ bits recovered}}.$$

Coded PIR over Files : Some known protocols

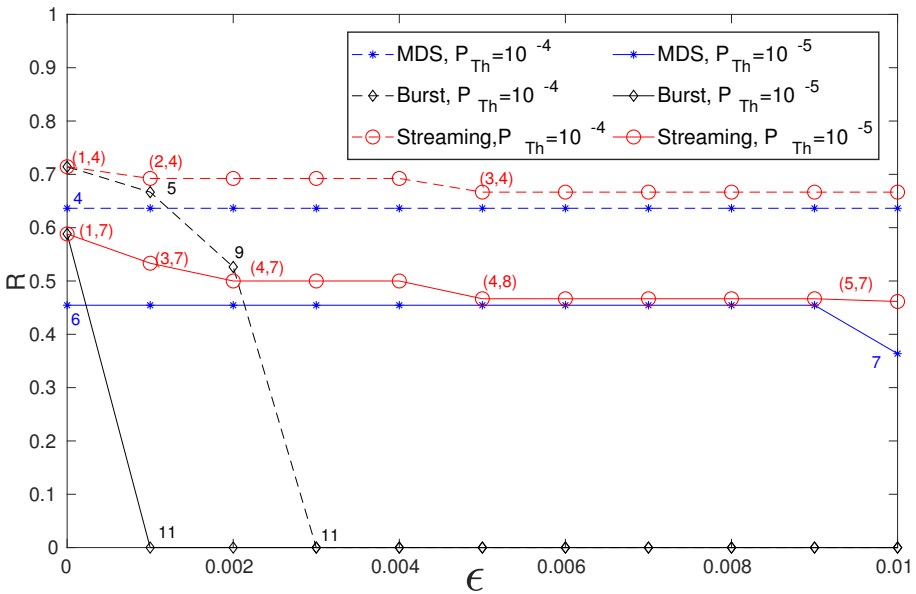
- PIR protocol specific to the code used.
- Framework to emulate replication based protocols on coded database not known.
- Notation: n servers, m files, each file divided into $k\alpha$ sub-files
- H. Sun and S. A. Jafar: n replication, $\alpha = n^m$, $cPoP = 1 + \frac{1}{n} + \frac{1}{n^2} + \cdots + \frac{1}{n^{m-1}}$ (optimal) .
- K. Banawan and S. Ulukus: (n,k) MDS code, $\alpha = kn^m$, $cPoP = 1 + R + R^2 + \cdots + R^{m-1}$; $R = \frac{k}{n}$ (optimal).
- R. Tajeddine, O. W. Gnilke and S. El Rouayheb: (n, k) MDS code, $\alpha = n - k$, cPoP$= \frac{n}{n-k}$ (not optimal)
- S. Kumar, E. Rosnes and A. G. Amat: (n, k, d_{min}) linear code, $\alpha \leq d_{min} - 1$, cPoP$\leq \frac{n}{d_{min}-1}$

Explaining the Simulation over GE channel with $\alpha = 10^{-4}, \beta = 0.6$

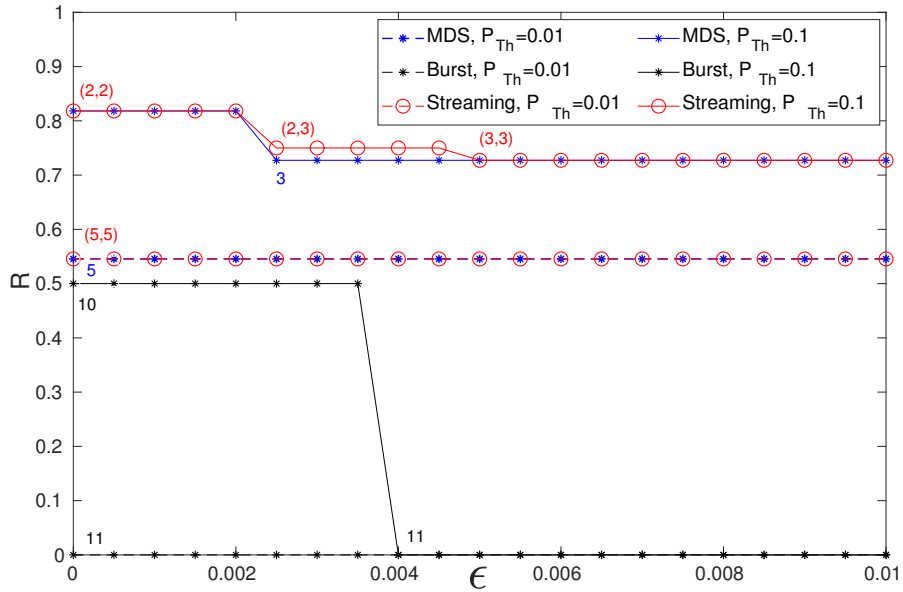


- When $\epsilon = 0$, the GE channel introduces burst erasures and as ϵ increases random erasures are seen.

Guaranteed rate under a BEP threshold



$\mathrm{GE}(\alpha = 10^{-4}, \beta = 0.5, \epsilon_0 = \epsilon, \epsilon_1 = 1)$



$\mathrm{GE}(\alpha = 0.1, \beta = 0.5, \epsilon_0 = \epsilon, \epsilon_1 = 1)$

$\tau = 10$. Best rate possible under streaming code setting, MDS setting and burst only setting when BEP is bounded by $P_{Th} = 10^{-5}$.