Distributed storage systems from combinatorial designs

Aditya Ramamoorthy

November 20, 2014

Department of Electrical and Computer Engineering, Iowa State University,
Joint work with Oktay Olmez (Ankara University, Turkey)

Data in all shapes and sizes!

NETFLIX

facebook

Dropbox

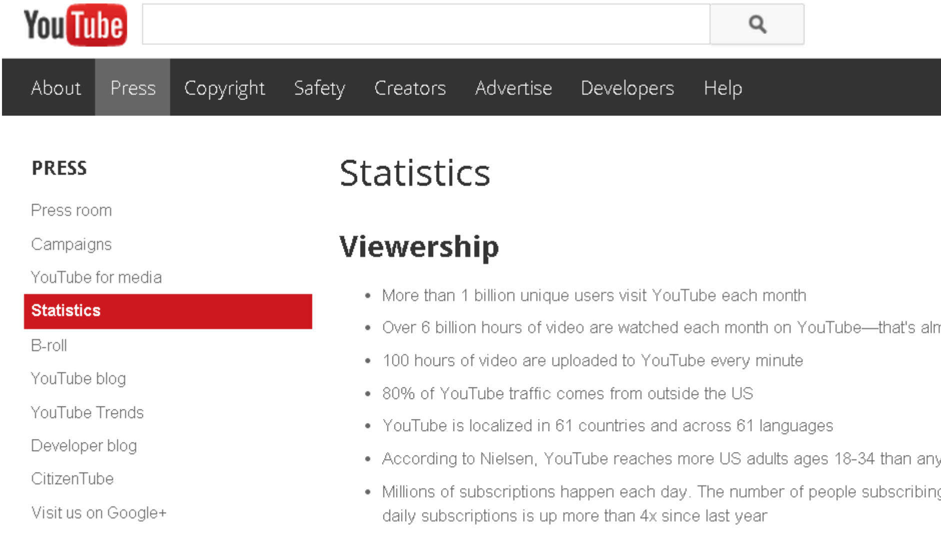
amazon web services

Microsoft Azure

hulu

YouTube

Sample statistics from Youtube



Several challenges...

- Access needs to be reliable.
  - *Indeed, server failure is the norm rather than the exception.* (Source: hadoop.apache.org)
- System needs to be efficient.
  - *Failure recovery must be seamless and be inexpensive (bandwidth, time, energy etc.).*

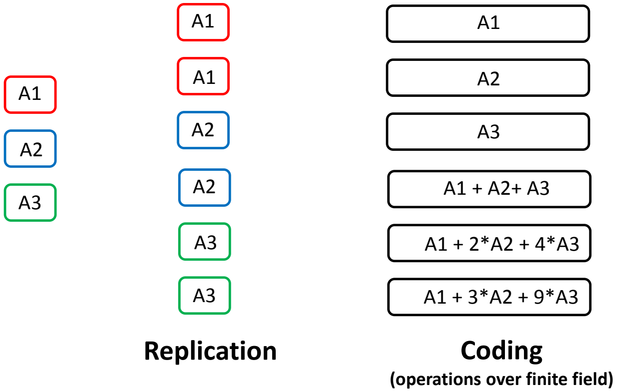
Several challenges...

- Access needs to be reliable.
  - *Indeed, server failure is the norm rather than the exception.* (Source: hadoop.apache.org)
- System needs to be efficient.
  - *Failure recovery must be seamless and be inexpensive (bandwidth, time, energy etc.).*
- Host of other issues such as security, privacy etc.

Several challenges...

- Access needs to be reliable.
  - *Indeed, server failure is the norm rather than the exception.* (Source: hadoop.apache.org)
- System needs to be efficient.
  - *Failure recovery must be seamless and be inexpensive (bandwidth, time, energy etc.).*
- Host of other issues such as security, privacy etc.
  - *Not discussed in this talk...*

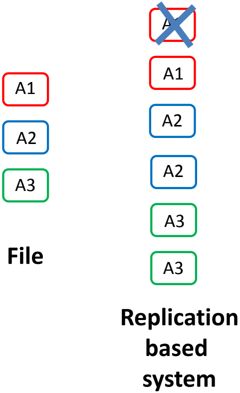
Replication vs. coding



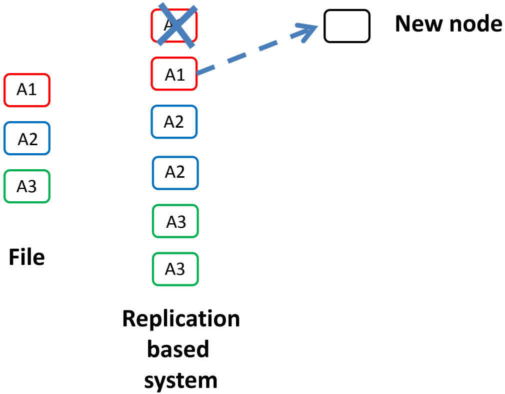
Observation

Both systems have same redundancy, but coded solution can recover from any three node failure event.

Dealing with failure in replication based systems



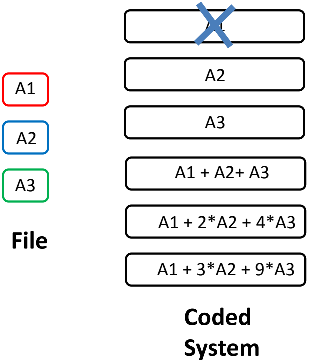
Repair in replication based systems



Observation

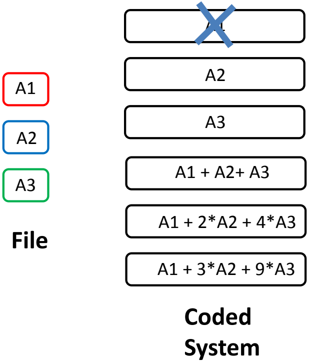
Repair simply by downloading from the existing copy!

Repair in coded systems



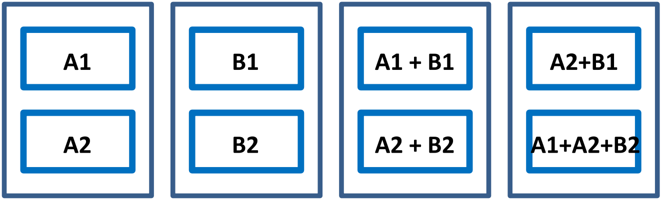
- Packet $A1$ cannot be recovered unless the file $(A1, A2, A3)$ is recovered.

Repair in coded systems



- Packet $A1$ cannot be recovered unless the file $(A1, A2, A3)$ is recovered.
- This requires connecting to three nodes and downloading one packet from each of them.

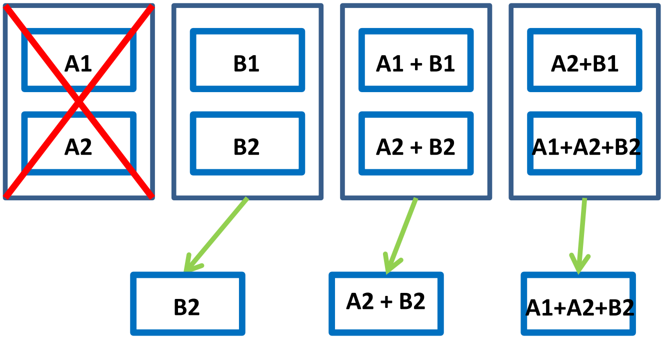
Can we do better - EVENODD Example [Blaum et al. '95]



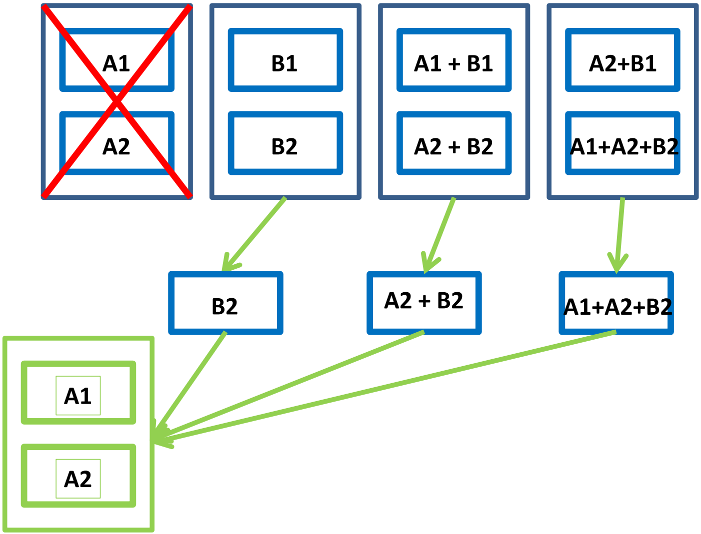
Observation

($n = 4, k = 2$) code. File consists of four packets ($A1, A2, A3, A4$). File can be reconstructed from any two nodes. Resilient to two failures.

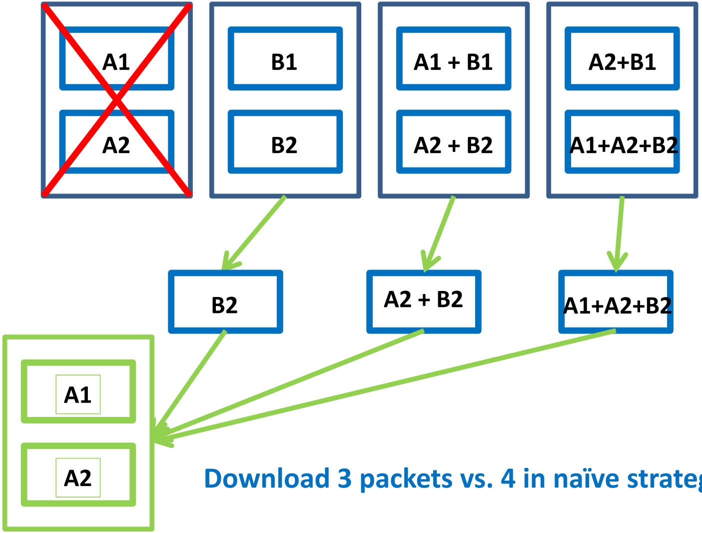
Can we do better - EVENODD Example



Can we do better - EVENODD Example



Can we do better - EVENODD Example



Download 3 packets vs. 4 in naïve strategy!

Different notions of repair efficiency

- Repair bandwidth: Attempts to minimize the amount of data downloaded for reconstructing the failed node.

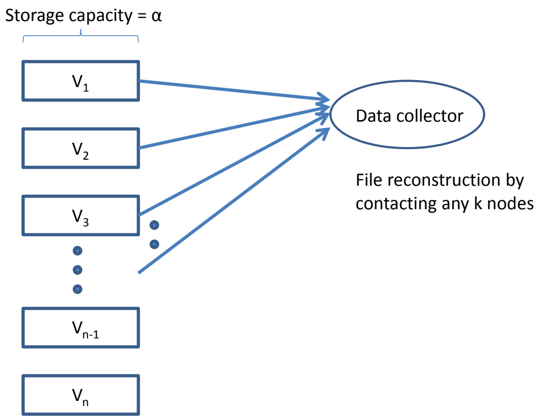
Different notions of repair efficiency

- Repair bandwidth: Attempts to minimize the amount of data downloaded for reconstructing the failed node.
- Local repair: Attempts to minimize the number of nodes contacted for recovering the node.

Different notions of repair efficiency

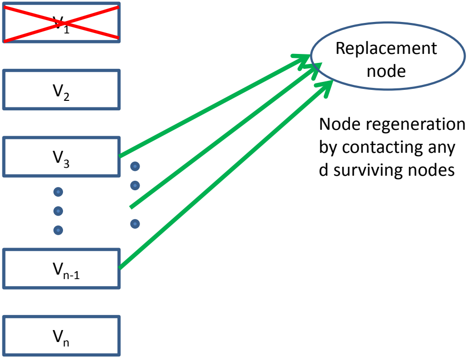
- Repair bandwidth: Attempts to minimize the amount of data downloaded for reconstructing the failed node.
- Local repair: Attempts to minimize the number of nodes contacted for recovering the node.
- There are probably other metrics as well in practice, but these appear to be tractable for code design.

(n, k, d)- Distributed storage system [Dimakis et al. 10]



- File of size $\mathcal{M}$ packets or symbols stored on n nodes.
- Each node stores α symbols.
- Any user can reconstruct the file by contacting any k nodes. (MDS property)

(n, k, d)- Distributed storage system [Dimakis et al. 10]



- A failed node can be reconstructed by contacting any d $(d \geq k)$ surviving nodes and downloading β packets from each.
  - d - repair degree, β - normalized repair bandwidth.
- Storage capacity vs. repair bandwidth tradeoff was characterized for the case of *functional repair*.

(n, k, d)- Distributed storage system with exact repair

- Exact copy of the failed node needs to be produced.

(n, k, d)- Distributed storage system with exact repair

- Exact copy of the failed node needs to be produced.
- Minimum storage regenerating (MSR) point: Store exactly $\mathcal{M}/k$ packets per node, i.e., storage capacity of node is minimum.
  - Constructions from [Cadambe et al. 2013 & others].

(n, k, d)- Distributed storage system with exact repair

- Exact copy of the failed node needs to be produced.
- Minimum storage regenerating (MSR) point: Store exactly $\mathcal{M}/k$ packets per node, i.e., storage capacity of node is minimum.
  - Constructions from [Cadambe et al. 2013 & others].
- Minimum bandwidth regenerating (MBR) point: Exactly α packets are downloaded for node regeneration. Equals storage capacity of a node.
  - Constructions from [Rashmi et al. 2011 & others].
- We focus on MBR constructions in this talk.

Easy repair & reliable distributed storage systems

- Advantage of replication based systems is easy reconstruction; drawback is storage inefficiency.
- Advantage of coded systems is optimum storage vs. repair bandwidth tradeoff; drawback is complicated reconstruction.

Easy repair & reliable distributed storage systems

- Advantage of replication based systems is easy reconstruction; drawback is storage inefficiency.
- Advantage of coded systems is optimum storage vs. repair bandwidth tradeoff; drawback is complicated reconstruction.
- This work - attempt to combine best of both worlds ...

Systems with exact and uncoded repair [El Rouayheb and Ramchandran '10]

- Exact repair constructions typically use coding across the source symbols.
  - Read-write bandwidth of machines is often a bottleneck in system operation.
  - Coding across potentially large ($\approx$ GB) packets can be memory intensive.
  - Decoding coded packets can cause an increased repair time [Jiekak et al. '12].

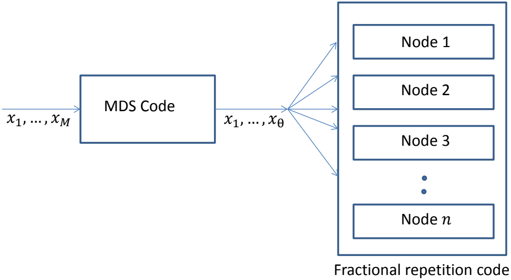
Systems with exact and uncoded repair [El Rouayheb and Ramchandran '10]

- Exact repair constructions typically use coding across the source symbols.
  - Read-write bandwidth of machines is often a bottleneck in system operation.
  - Coding across potentially large ($\approx$ GB) packets can be memory intensive.
  - Decoding coded packets can cause an increased repair time [Jiekak et al. '12].

Definition (Exact and uncoded repair)

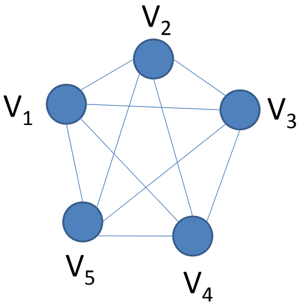
- Exact regeneration by simply downloading symbols from the surviving nodes.
- Operate at the MBR point.
- Table-based repair - new node contacts a *specific* set of surviving nodes.

System Architecture



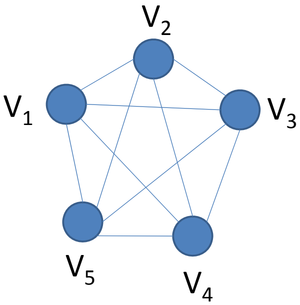
- Outer MDS code.
- Inner fractional repetition (FR) code - specifies placement of symbols on storage nodes.
    - File reconstruction if enough symbols are obtained from any k nodes.
    - Failure recovery depends on FR code properties.

System example - complete graph on 5 nodes, $d \geq k$



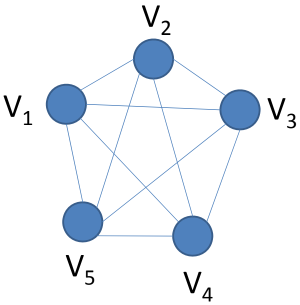
- File $(x_1, \ldots, x_9) \in \mathbb{F}_q^9$, $\mathcal{M} = 9$. Use $(10, 9)$ MDS code to get coded symbols $(y_1, \ldots, y_{10})$.
- Number of storage nodes $n = 5$, number of symbols $\theta = 10$.

System example - complete graph on 5 nodes, $d \geq k$



- File $(x_1, \ldots, x_9) \in \mathbb{F}_q^9$, $\mathcal{M} = 9$. Use $(10, 9)$ MDS code to get coded symbols $(y_1, \ldots, y_{10})$.
- Number of storage nodes $n = 5$, number of symbols $\theta = 10$.
- Label edges of the complete graph.

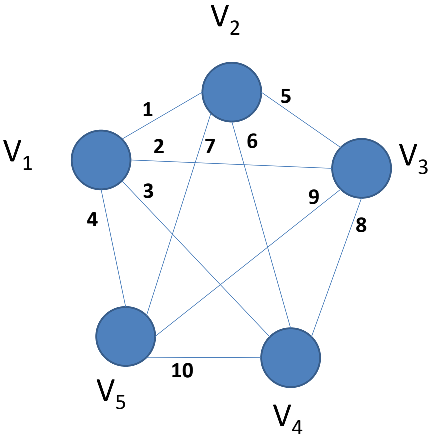
System example - complete graph on 5 nodes, $d \geq k$



- File $(x_1, \dots, x_9) \in \mathbb{F}_q^9$, $\mathcal{M} = 9$. Use $(10, 9)$ MDS code to get coded symbols $(y_1, \dots, y_{10})$.
- Number of storage nodes $n = 5$, number of symbols $\theta = 10$.
- Label edges of the complete graph.
- Storage nodes store incident symbols.

System example - complete graph on 5 nodes, $d \geq k$

V_1	1 2 3 4
V_2	1 5 6 7
V_3	2 5 8 9
V_4	3 6 8 10
V_5	4 7 9 10



System example - complete graph on 5 nodes, $d \geq k$ Analyzing file size

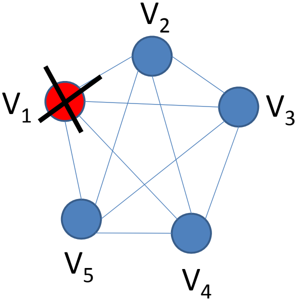
V_1	1 2 3 4
V_2	1 5 6 7
V_3	2 5 8 9
V_4	3 6 8 10
V_5	4 7 9 10

- $n = 5$ nodes, $\theta = 10$ symbols.
- Storage nodes are 4-sized subsets.
  Using inclusion-exclusion principle

$$|A_1 \cup A_2 \cup A_3| = \sum_i |A_i| - \sum_{i<j} |A_i \cap A_j| + |\cap_i A_i|$$
$$= 3 \times 4 - \binom{3}{2} + 0 = 9.$$

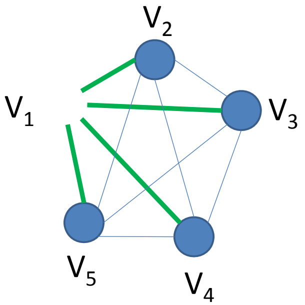
  Thus, $k = 3$.
- Repair degree $d = 4$.

Failure analysis



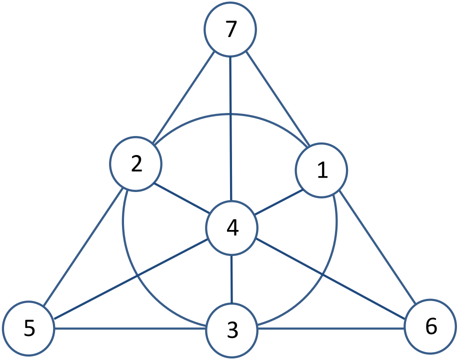
- Suppose node V_1 fails.

Failure analysis



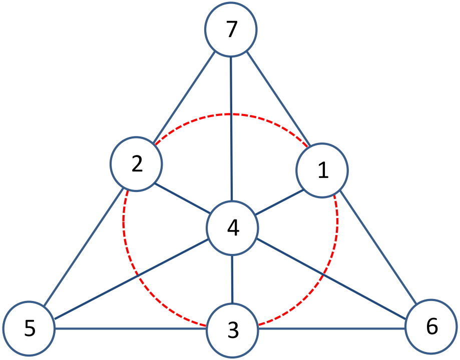
- Suppose node V_1 fails.
- One symbol from all the other nodes is needed for recovery.
- Need to contact at least k nodes.

FR codes from combinatorial designs - Fano plane

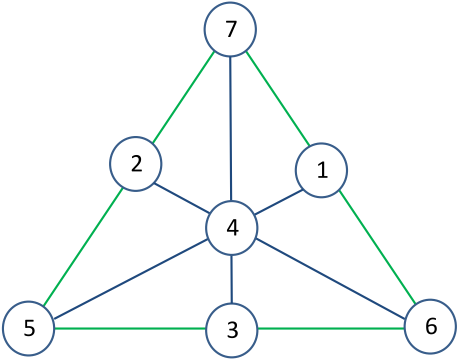


- File $(x_1, \ldots, x_6) \in \mathbb{F}_q^6$, $\mathcal{M} = 6$. Use $(7, 6)$ MDS code to get coded symbols $(y_1, \ldots, y_7)$.
- Number of storage nodes $n = 7$.
- Nodes correspond to lines in Fano plane.

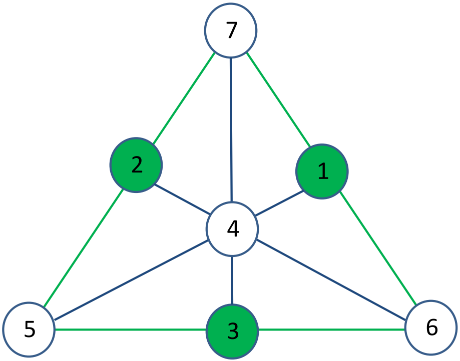
FR codes from combinatorial designs - Fano plane



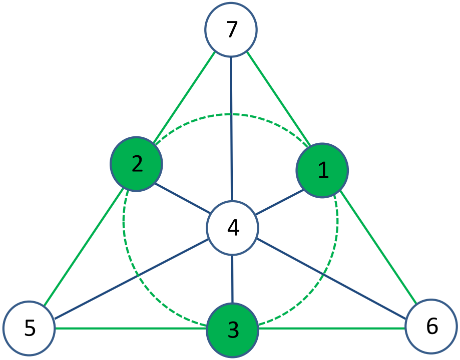
FR codes from combinatorial designs - Fano plane



FR codes from combinatorial designs - Fano plane



FR codes from combinatorial designs - Fano plane



FR code from Fano plane
Analyzing file size

V_1	1 2 3
V_2	1 4 5
V_3	1 6 7
V_4	2 4 6
V_5	2 5 7
V_6	3 5 6
V_7	3 4 7

- Nodes are 3-sized subsets. Using inclusion-exclusion principle

$$|A_1 \cup A_2 \cup A_3| = \sum_i |A_i| - \sum_{i<j} |A_i \cap A_j| + |\cap_i A_i|$$

- Depending on choice of $A_i, i = 1, \ldots, 3$, three-way intersection can either be zero or 1. Minimum value is $3 \times 3 - \binom{3}{2} = 6$. Hence, $k = 3$.
- Failure recovery by contacting $d = 3$ nodes.

Key questions in FR code design

Key questions in FR code design

- Can we construct FR codes that are flexible in the number of failures that they tolerate?
  - Need flexible combinatorial designs: formalized in our work by resolvability.

Key questions in FR code design

- Can we construct FR codes that are flexible in the number of failures that they tolerate?
  - Need flexible combinatorial designs: formalized in our work by resolvability.
- For a given FR code, can we determine the maximum file size that can be supported?
  - Hard problem for a general combinatorial design. Need to find the minimum number of symbols covered over all k-sized subsets of the storage nodes; inclusion-exclusion analysis may not always be possible (though bounds can be obtained).
  - FR codes with the same parameters $(n, k, d, \theta, \alpha)$ can have different file sizes.
  - We determine file size for our constructions for certain parameter ranges.

Key questions in FR code design

- How to calculate system metrics such as minimum distance?

Definition

The minimum distance of a DSS denoted d_{min} is defined to be the size of the smallest subset of storage nodes whose failure guarantees that the file is not recoverable from the surviving nodes under any possible recovery mechanism.

Contributions of our work - I [Olmez & R. 2012]

- Construct a large class of codes from resolvable designs where failure resilience of system can be varied in a simple manner (Prior constructions typically lack this flexibility).
  - Simple implementation of repair table.
- Construct FR codes that cannot be constructed using Steiner systems
  - Answers an open question raised in [El Rouayheb-Ramchandran '10].
- Determine the maximum supported file size for several parameter ranges.
  - Prior work mostly provides lower bounds.

Example of a resolvable FR with $\rho = 2$ - Row-Column construction

$$A = \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

Example of a resolvable FR with $\rho = 2$ - Row-Column construction

$$A = \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

$\boxed{1\ \ 2\ \ 3}$ $\boxed{4\ \ 5\ \ 6}$ $\boxed{7\ \ 8\ \ 9}$

Example of a resolvable FR with $\rho = 2$ - Row-Column construction

$$A = \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

$\boxed{1 \quad 2 \quad 3}$ $\boxed{4 \quad 5 \quad 6}$ $\boxed{7 \quad 8 \quad 9}$

$\boxed{1 \quad 4 \quad 7}$ $\boxed{2 \quad 5 \quad 8}$ $\boxed{3 \quad 6 \quad 9}$

Example of Parallel Classes

$$A = \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

Example of Parallel Classes

$$A = \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

1 2 3	4 5 6	7 8 9	Parallel class 1
1 4 7	2 5 8	3 6 9	Parallel class 2

Resolvable fractional repetition code

Definition

Let $\mathcal{C} = (\Omega, V)$ where $V = \{V_1, \ldots, V_n\}$ be a FR code. A subset $P \subset V$ is said to be a parallel class if

- $V_i \in P$ and $V_j \in P$ with $i \neq j$ we have $V_i \cap V_j = \emptyset$, and
- $\cup_{\{j : V_j \in P\}} V_j = \Omega$.

- A partition of V into r parallel classes is called a resolution.
- If there exists at least one resolution then the code is called a resolvable fractional repetition code.

Example construction from 2-D subspaces of $\mathbb{F}_3^3$

There are thirteen two-dimensional subspaces of $\mathbb{F}_3^3$ which are the solutions to homogeneous linear equations over $\mathbb{F}_3$ in three variables.

Example construction from 2-D subspaces of $\mathbb{F}_3^3$

There are thirteen two-dimensional subspaces of $\mathbb{F}_3^3$ which are the solutions to homogeneous linear equations over $\mathbb{F}_3$ in three variables.

- Equation: $x_1 = 0$

Example construction from 2-D subspaces of $\mathbb{F}_3^3$

There are thirteen two-dimensional subspaces of $\mathbb{F}_3^3$ which are the solutions to homogeneous linear equations over $\mathbb{F}_3$ in three variables.

- Equation: $x_1 = 0$
- Subspace: $\{000, 001, 002, 010, 020, 011, 012, 021, 022\}$

Example construction from 2-D subspaces of $\mathbb{F}_3^3$

There are thirteen two-dimensional subspaces of $\mathbb{F}_3^3$ which are the solutions to homogeneous linear equations over $\mathbb{F}_3$ in three variables.

- Equation: $x_1 = 0$
- Subspace: $\{000, 001, 002, 010, 020, 011, 012, 021, 022\}$
- Equation: $x_1 + 2x_2 + 2x_3 = 0$

Example construction from 2-D subspaces of $\mathbb{F}_3^3$

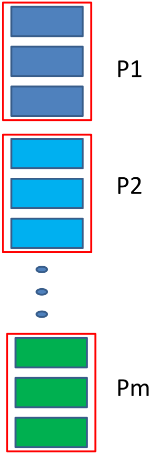
There are thirteen two-dimensional subspaces of $\mathbb{F}_3^3$ which are the solutions to homogeneous linear equations over $\mathbb{F}_3$ in three variables.

- Equation: $x_1 = 0$
- Subspace: $\{000, 001, 002, 010, 020, 011, 012, 021, 022\}$
- Equation: $x_1 + 2x_2 + 2x_3 = 0$
- Subspace: $\{000, 012, 021, 110, 101, 122, 220, 202, 211\}$

The other blocks are additive cosets of these 13 representatives. For example,

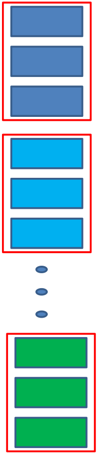
$$B_1 = \{000, 001, 002, 010, 020, 011, 012, 021, 022\}$$
$$B_2 = \{100, 101, 102, 110, 120, 111, 112, 121, 122\}$$
$$B_3 = \{200, 201, 202, 210, 220, 211, 212, 221, 222\}$$

Observations



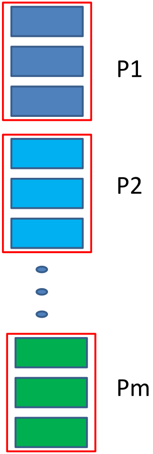
- $\{B_1, B_2, B_3\}$ covers 27 symbols - is a parallel class!

Observations



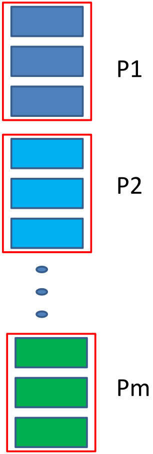
- $\{B_1, B_2, B_3\}$ covers 27 symbols - is a parallel class!
- There are a total of 13 parallel classes.

Observations



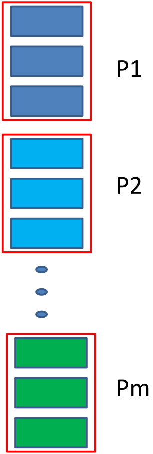
- $\{B_1, B_2, B_3\}$ covers 27 symbols - is a parallel class!
- There are a total of 13 parallel classes.
- Two nodes from different parallel classes have exactly 3 symbols in common.

Observations

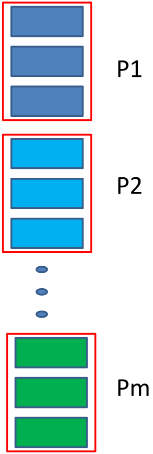


- $\{B_1, B_2, B_3\}$ covers 27 symbols - is a parallel class!
- There are a total of 13 parallel classes.
- Two nodes from different parallel classes have exactly 3 symbols in common.
- Each symbol is repeated $\rho = 13$ times.

Observations

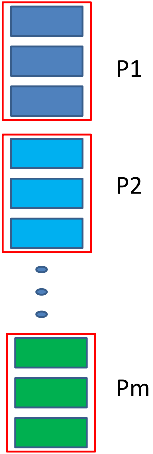


Observations



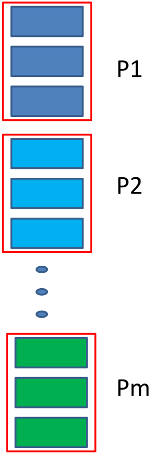
- Failure resilience can be varied from 1 to 12 failures! - Significant flexibility as compared to Steiner systems considered in [El Rouayheb-Ramchandran '10].

Observations



- Failure resilience can be varied from 1 to 12 failures! - Significant flexibility as compared to Steiner systems considered in [El Rouayheb-Ramchandran '10].
- Simply choose an appropriate number of parallel classes.

Observations



- Failure resilience can be varied from 1 to 12 failures! - Significant flexibility as compared to Steiner systems considered in [El Rouayheb-Ramchandran '10].
- Simply choose an appropriate number of parallel classes.
- For failure recovery simply contact the intact parallel class.

General Construction [Olmez & R. 2012]

Construction

Given an affine resolvable design with parameters $(n, \theta, \alpha, \rho) = \left(\frac{q^{m+1}-1}{q-1}, q^m, q^{m-1}, \frac{q^m-1}{q-1}\right)$ *with blocks* $B_1, B_2, \cdots, B_n$*, an FR code* $\mathcal{C}$ *can be obtained by taking* $\mathcal{C} = \{B_1, B_2, \cdots, B_n\}$.

General Construction [Olmez & R. 2012]

Construction

Given an affine resolvable design with parameters $(n, \theta, \alpha, \rho) = \left(\frac{q^{m+1}-1}{q-1}, q^m, q^{m-1}, \frac{q^m-1}{q-1}\right)$ *with blocks* $B_1, B_2, \cdots, B_n$*, an FR code* $\mathcal{C}$ *can be obtained by taking* $\mathcal{C} = \{B_1, B_2, \cdots, B_n\}$.

Corollary

The above construction yields an FR code with $\beta = \dfrac{\alpha^2}{\theta}$.

General Construction [Olmez & R. 2012]

Construction

Given an affine resolvable design with parameters $(n, \theta, \alpha, \rho) = \left(\frac{q^{m+1}-1}{q-1}, q^m, q^{m-1}, \frac{q^m-1}{q-1}\right)$ *with blocks* $B_1, B_2, \cdots, B_n$*, an FR code* $\mathcal{C}$ *can be obtained by taking* $\mathcal{C} = \{B_1, B_2, \cdots, B_n\}$*.*

Corollary

The above construction yields an FR code with $\beta = \dfrac{\alpha^2}{\theta}$*.*

- Ability to obtain codes with higher normalized repair bandwidth β. These parameters cannot be obtained by trivially treating each symbol in a smaller code as consisting of a larger number of symbols.

Implications of result for $q = 4, m = 5, \delta = 4$,

- Obtain a FR code with $\theta = 1024$ symbols, storage capacity $\alpha = 256$ symbols, normalized repair bandwidth $\beta = 64$.
- Failure resilience can be varied from 1 to 340!
- Prior constructions lack this flexibility.

File size analysis [Olmez & R. 2012]

Theorem

For $q > m$ and $m \geq k$, we can choose the parallel classes such that the file size $\mathcal{M} = q^m\left(1 - \left(1 - \frac{1}{q}\right)^k\right)$.

- File size analysis for FR codes is challenging as one needs to compute the minimum cardinality of the union of all k-sized storage nodes.
- However, careful analysis of the algebraic properties of the design can often help.

Constructions from mutually orthogonal Latin squares (MOLS) [Olmez & R. 2012]

$$A = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{array}$$

$$L_1 = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{array}$$

$$L_2 = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \\ 2 & 1 & 4 & 3 \end{array}$$

- L_1 and L_2 are mutually orthogonal.
- Choose blocks as elements of A corresponding to locations in L_i.

$$P^{L_1} = \{\{1,6,11,16\},\{2,5,12,15\}, \{3,8,9,14\},\{4,7,10,13\}\}$$

  - Forms a parallel class.

$$P^{\text{rows}} = \{\{1,2,3,4\},\{5,6,7,8\},\{9,10,11,12\},\{13,14,15,16\}\}$$

$$P^{\text{cols}} = \{\{1,5,9,13\},\{2,6,10,14\},\{3,7,11,15\},\{4,8,12,16\}\}$$

$$P^{L_1} = \{\{1,6,11,16\},\{2,5,12,15\},\{3,8,9,14\},\{4,7,10,13\}\}$$

$$P^{L_2} = \{\{1,7,12,14\},\{2,8,11,13\},\{3,5,10,16\},\{4,6,9,15\}\}$$

- For $N = p^s$, we can construct $N - 1$ MOLS of size $N \times N$.

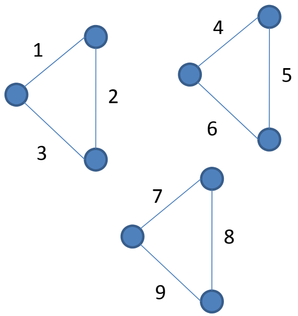
- For $N = p^s$, we can construct $N - 1$ MOLS of size $N \times N$.

- If $N \neq 2, 6$, constructions of *two* MOLS are known [Bose-Shrikhande-Parker '60].

Implications of result

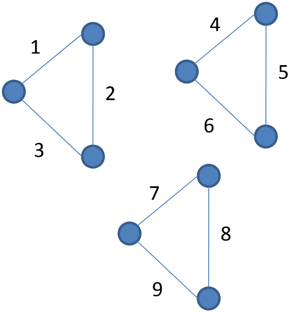
- We can construct a FR code starting with two MOLS of order 10 using [Bose-Shrikhande-Parker '60].
- However, Steiner system with storage capacity $\alpha = 10$ and number of symbols $\theta = 100$ does not exist.
  - Equivalent to the existence of a projective plane of order 10 which is known not to exist [Lam et al. '89].
  - Answers open question posed in [El Rouayheb-Ramchandran '10]

Local Repair Example, $d < k$



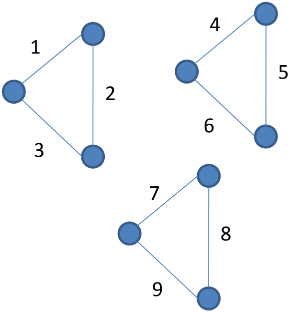
- File $(x_1, \dots, x_5) \in \mathbb{F}_q^9$, $\mathcal{M} = 5$. Use $(9, 5)$ MDS code to get coded symbols $(y_1, \dots, y_9)$.
- Number of storage nodes $n = 9$.
- Nodes store incident edge labels.

Local Repair Example, $d < k$



- Failure recovery by contacting surviving nodes in the same column, $d = 2$.
- Any four nodes cover $\mathcal{M} = 5$ symbols, hence $k = 4$.

Local Repair Example, $d < k$



- Failure recovery by contacting surviving nodes in the same column, $d = 2$.
- Any four nodes cover $\mathcal{M} = 5$ symbols, hence $k = 4$.
- Repair degree $d < k$ ...
- Notion of local repair [Gopalan et al. '12, Papailopolous et al. '13, Oggier et al. '13]

Contributions of our work - II [Olmez & R. 2013]

- Constructions of locally recoverable FR codes.
  - Local recovery from single failure - from high girth graphs.
  - Local recovery from multiple failures - Collection of local FR codes. Global code inherits properties of the local one.
- Derive minimum distance bound for local, exact and uncoded repair. Our codes meet this bound for specific parameters.

Locally Recoverable FR codes from high-girth graphs [Olmez & R. 2013]

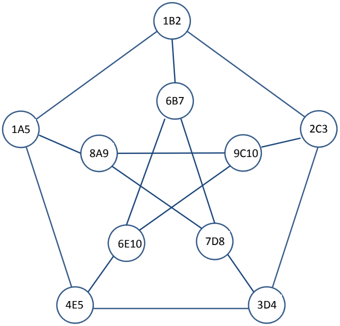
Local recovery from single failure.

An (s, g)-graph, denoted Γ: vertex degree s, girth g.

(i) Index the edges from 1 to $\frac{ns}{2}$.

(ii) Each vertex $\equiv$ storage node; stores the symbols incident on it.

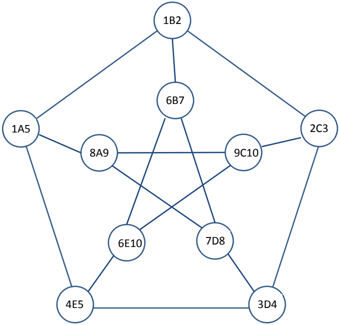
Petersen Graph - degree 3, girth 5



- Parameters $n = 10, k = 5, \alpha = 3, \rho = 2, d = 3$ and $\mathcal{M} = 10$.
- Can be shown that construction meets the minimum distance bound.

$$d_{\min} \leq n - \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil - \left\lceil \frac{\mathcal{M}}{d\alpha} \right\rceil + 2$$

Petersen Graph - degree 3, girth 5



- Parameters $n = 10, k = 5, \alpha = 3, \rho = 2, d = 3$ and $\mathcal{M} = 10$.
- Can be shown that construction meets the minimum distance bound.

$$d_{\min} \leq n - \left\lceil \frac{\mathcal{M}}{\alpha} \right\rceil - \left\lceil \frac{\mathcal{M}}{d\alpha} \right\rceil + 2$$

General result...

Theorem

Let $\Gamma = (V, E)$ be a (s, g)-graph with $|V| = n$ and $s > 2$. If $g \geq k = as + b$ such that $s > b \geq a + 1$, then $\mathcal{C}$ obtained from Γ is optimal with respect to the minimum distance bound when the file size $\mathcal{M} = k(s-1)$.

Construction from collection of local FR codes

Pick FR code (Ω, V) with parameters n - number of nodes, θ - number of symbols, α - storage capacity, ρ- repetition degree, such that

- Any $\Delta+1$ nodes in V cover θ symbols.
  - Need to aim for a Δ that is somewhat low.
- Intersection size $|V_i \cap V_j|$ either equals β or 0.
  - Allows for symmetric download.

Construction from collection of local FR codes

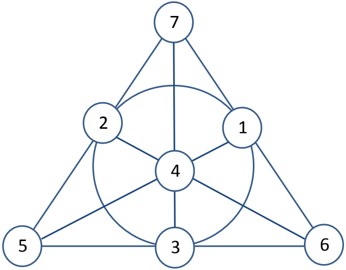
Pick FR code (Ω, V) with parameters n - number of nodes, θ - number of symbols, α - storage capacity, ρ- repetition degree, such that

- Any $\Delta+1$ nodes in V cover θ symbols.
  - Need to aim for a Δ that is somewhat low.
- Intersection size $|V_i \cap V_j|$ either equals β or 0.
  - Allows for symmetric download.

Construct $\bar{\mathcal{C}}$ by considering the disjoint union of $l(> 1)$ copies of $\mathcal{C}$. Thus, $\bar{\mathcal{C}}$ has parameters $(ln, l\theta, \alpha, \beta)$.

Construction Example: Fano plane as a local FR code

V_1	1 2 3
V_2	1 4 5
V_3	1 6 7
V_4	2 4 6
V_5	2 5 7
V_6	3 5 6
V_7	3 4 7



- Parameters $(\theta, n, \alpha, \rho, \beta) = (7, 7, 3, 3, 1)$. Resilient up to two failures.
- Any $\Delta + 1 = 5$ nodes cover all 7 symbols.
- Any 4 nodes covers at least 6 (Corradi's lemma).

Construction Example

$X_1X_2X_4$	$X_2X_3X_5$	$X_3X_4X_6$	$X_4X_5X_7$	$X_5X_6X_1$	$X_6X_7X_2$	$X_7X_1X_3$
$Y_1Y_2Y_4$	$Y_2Y_3Y_5$	$Y_3Y_4Y_6$	$Y_4Y_5Y_7$	$Y_5Y_6Y_1$	$Y_6Y_7Y_2$	$Y_7Y_1Y_3$
$Z_1Z_2Z_4$	$Z_2Z_3Z_5$	$Z_3Z_4Z_6$	$Z_4Z_5Z_7$	$Z_5Z_6Z_1$	$Z_6Z_7Z_2$	$Z_7Z_1Z_3$
$T_1T_2T_4$	$T_2T_3T_5$	$T_3T_4T_6$	$T_4T_5T_7$	$T_5T_6T_1$	$T_6T_7T_2$	$T_7T_1T_3$

- 4 copies of Fano plane on: X_1^7, Y_1^7, Z_1^7 and T_1^7.
  - $n = 28, \theta = 28$, repair degree $= 3$.
- Any set of $k = 15$ nodes cover at least 17 symbols, hence $\mathcal{M} = 17$.
  - Code resilient to 13 failures.
  - Meets the minimum distance bound for locally recoverable FR codes that consist of local structures that are also FR codes.

General result [Olmez & R. 2013]

Theorem

Suppose that the parameters of the local FR code satisfy $(\rho - 1)\alpha\theta - (\theta + \alpha)(\Delta - 1)\beta \geq 0$. Let the file size be $\mathcal{M} = t\theta + \alpha$ for some $1 \leq t < l$. Then $\bar{\mathcal{C}}$ is minimum distance optimal.

- Condition allows us to estimate file size $\mathcal{M}$ using Corradi's lemma.
- Several local FR codes satisfy the condition.
  - Affine resolvable FR codes.
  - Projective plane based FR codes.
  - Complete graphs, cycle graphs etc.

Conclusions

Conclusions

- Present a large class of resolvable FR codes. Allow the system designer to vary the repetition degree within a large range in a simple manner.

Conclusions

- Present a large class of resolvable FR codes. Allow the system designer to vary the repetition degree within a large range in a simple manner.
- We answer a question posed in prior work [El Rouayheb and Ramchandran '10] about the existence of codes that are not derivable from Steiner systems.

Conclusions

- Present a large class of resolvable FR codes. Allow the system designer to vary the repetition degree within a large range in a simple manner.
- We answer a question posed in prior work [El Rouayheb and Ramchandran '10] about the existence of codes that are not derivable from Steiner systems.
- The systems under consideration require table-based repair. Resolvable nature of the code, makes the implementation of the table very simple.

Olmez & R., "Fractional repetition codes with flexible repair from combinatorial designs", preprint 2014 (on arxiv).

Conference papers at Allerton 2012, NetCod 2013 and Asilomar 2013.

Conclusions

Conclusions

- Our locally repairable FR codes meet the minimum distance bound for certain file size values.

Conclusions

- Our locally repairable FR codes meet the minimum distance bound for certain file size values.
- We also derive a minimum distance bound that is tighter in the case of codes with exact and uncoded repair.

Olmez & R., "Fractional repetition codes with flexible repair from combinatorial designs", 2014 (on arxiv).

Conference papers at Allerton 2012, NetCod 2013 and Asilomar 2013.