Hierarchical Codes: How to Make Erasure Codes Attractive for Peer-to-Peer Storage Systems

Alessandro Duminuco and Ernst Biersack
EURECOM
Sophia Antipolis, France

(Best paper award in P2P'08)

Presented by: Amir H. Payberah
amir@sics.se

What's The Problem?

What Is The Problem?

Does file backup fit the P2P model?

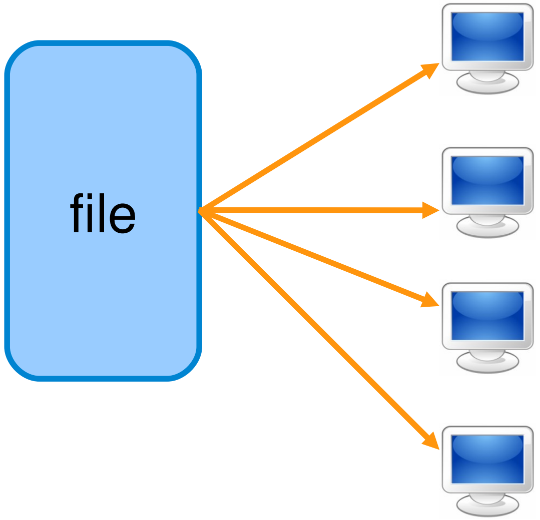
Churn and Redundancy

- The challenge in P2P model is to provide storage reliability under churn.

- The key solution is to add redundancy to the data.

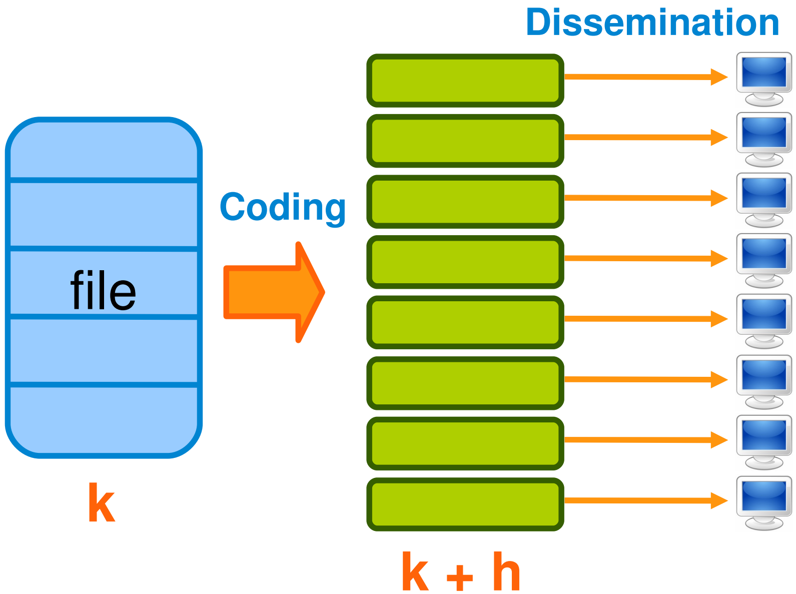
The Basic Solution: Replication

- With 4 replicas, even if 3 peers are offline we still have the file.
- Every file consumes storage for 4 times its size!!



A Better Solution: Coding

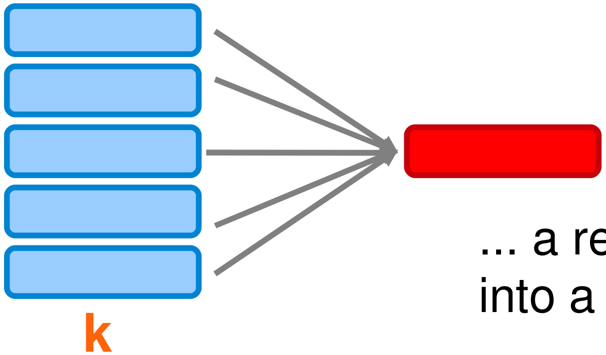
- Any k fragments are sufficient to reconstruct the file:
  - We can sustain any h losses.
- Every file consumes storage for (k+h)/k times its size:
  - If k=6 and h=3, (k+h)/k=1.5 .... Instead of 4!!



Repair Communication Cost

- Replication

- Coding

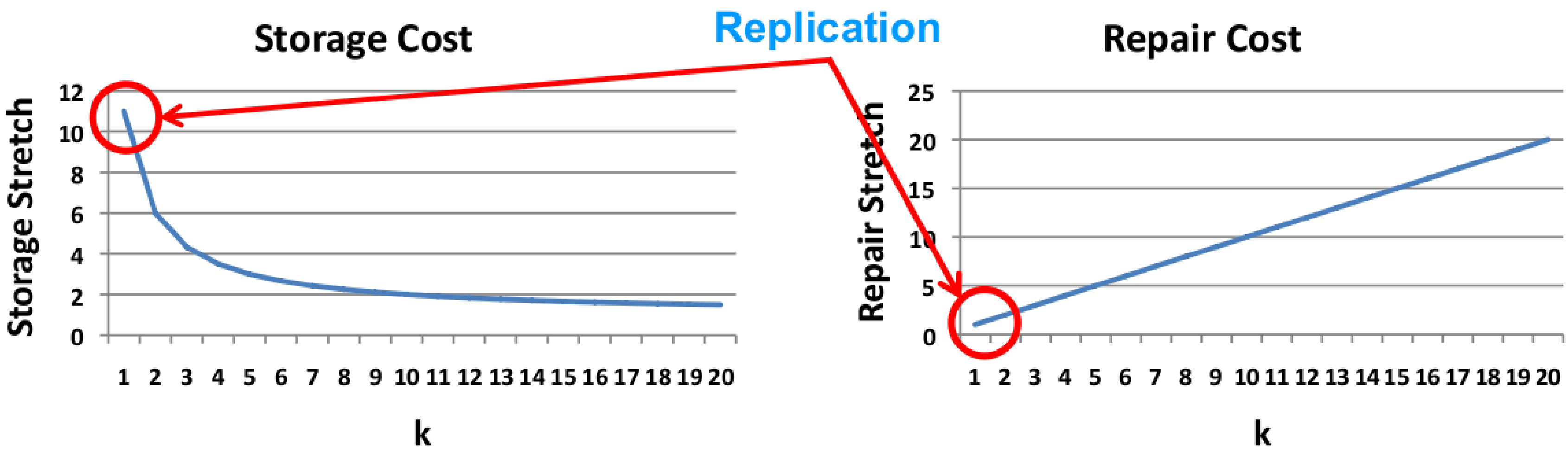


... a repair means combining k fragments into a new one.

- To create a single fragment we must transfer k fragments, i.e. the size-equivalent of the whole file!!

Storage vs Repair Cost

- If we want to sustain 10 losses:



Repair Cost makes coding unattractive.

Motivation

Can we mitigate the repair cost of coding while retaining storage efficiency?

Efficiency Metrics

- Redundancy factor
  - $\beta = |S| / |O|$
  - |S|: size of the stored data.
  - |O|: size of the original data.

- Repair degree
  - The amount of data read with respect to the amount of new redundant data created.
  - Denoted as d.

Efficiency Analysis

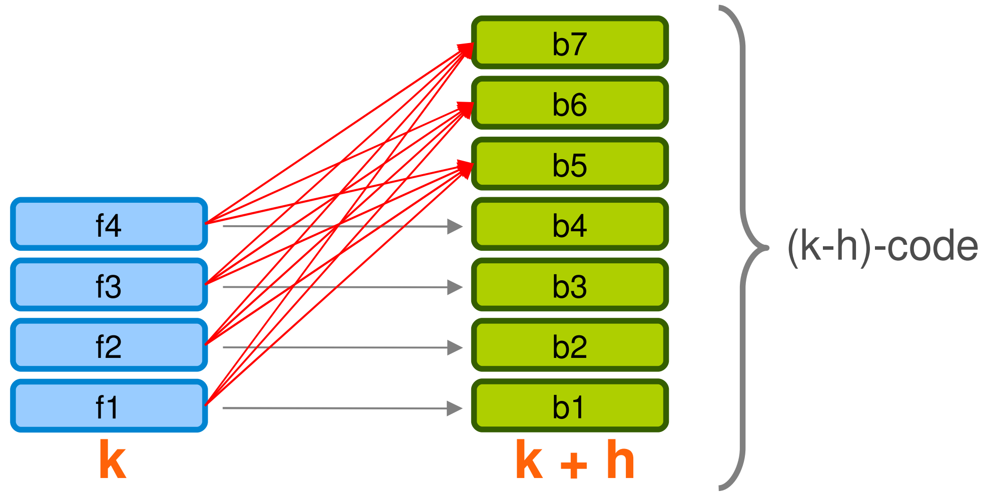
- Replication
  - $\beta = R$
  - $d = 1$
- Block replication
  - $\beta = R$
  - $d = 1$
- Erasure codes
  - $\beta = (k + h) / k$
  - $d = k$

Linear Codes

- A specific implementation of erasure codes.

- f_i: i^{th} fragment
- b_i:i^{th} fragment
- $c_{i,j}$: coefficients

$$b_i = \begin{cases} f_i & i \le k \\ \sum(c_{i,j} \times f_j) & k < i \le k + h \end{cases}$$

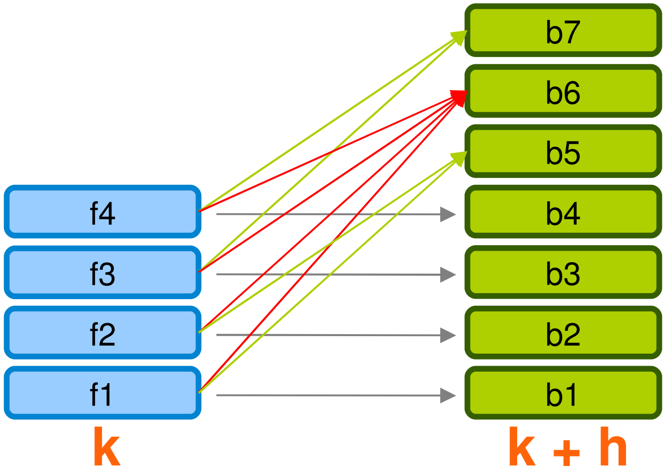


- Any 4 of these 7 fragments can reconstruct the original file if the coefficients are linearly independent. (will be back to it later)

- Repair degree d = k

Hierarchical Code

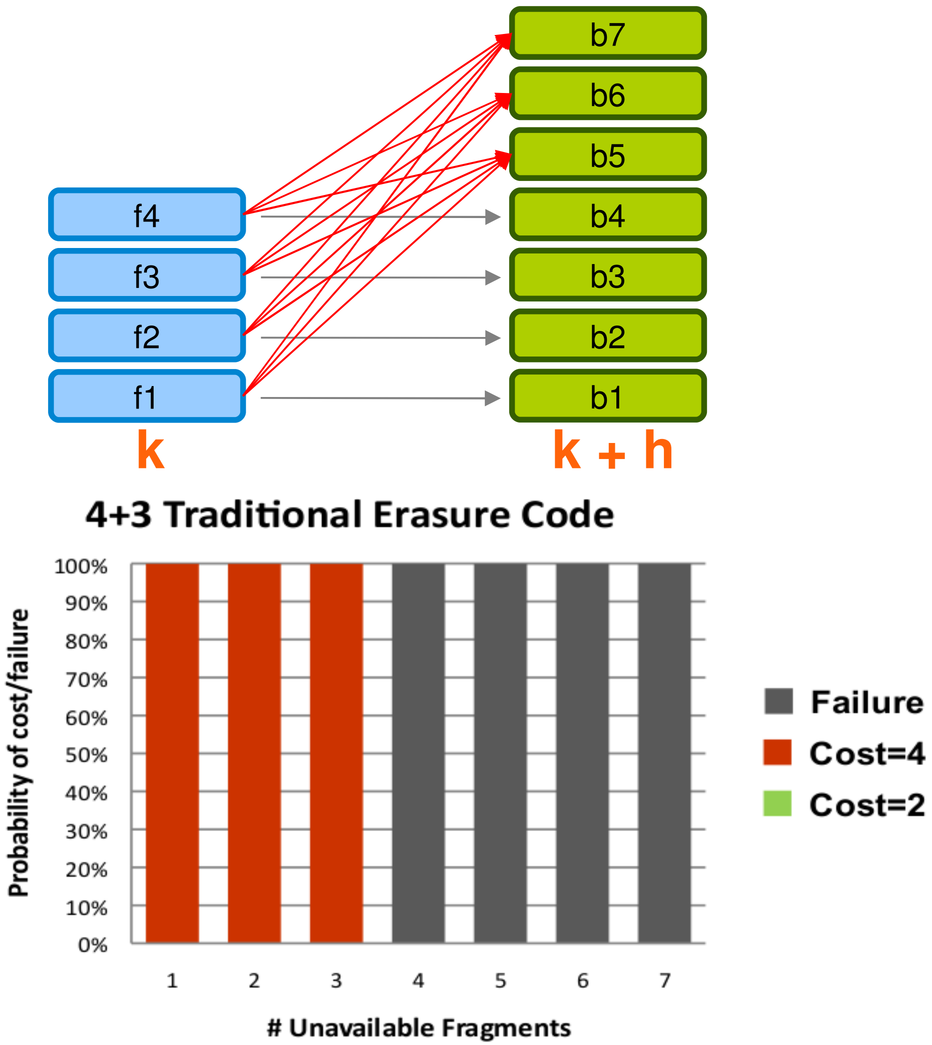
- Additional fragments can be linear combinations of a subset of the original ones.



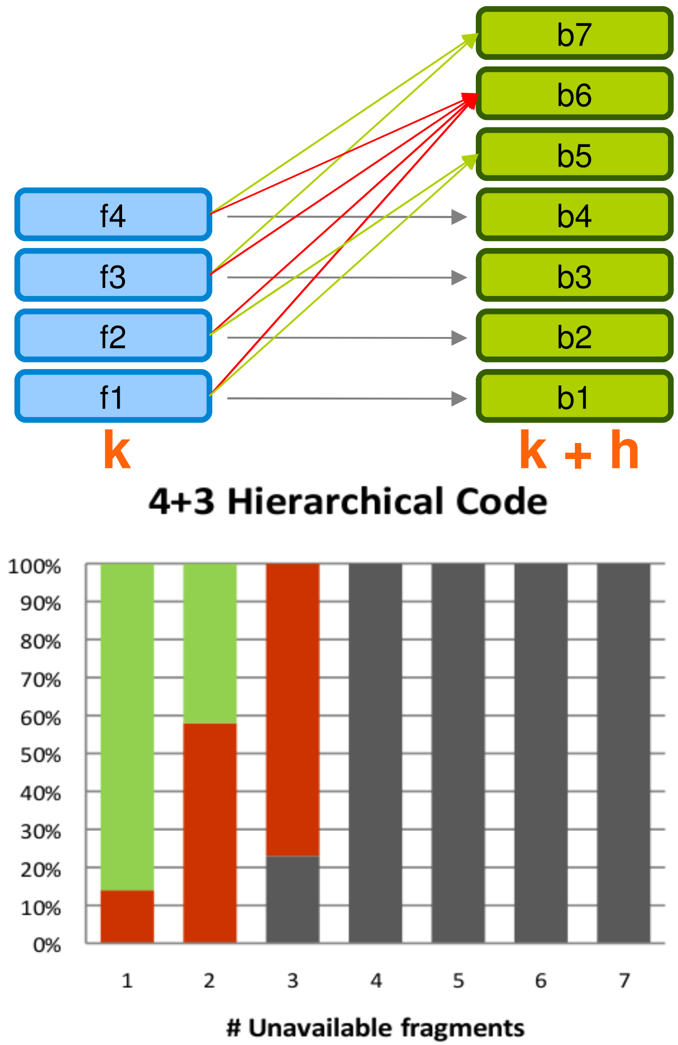
- Not all the subsets of 4 fragments are sufficient to reconstruct the file.

- The repair cost varies accordingly to the particular fragments that are available (we can have $d < k$).

Comparison



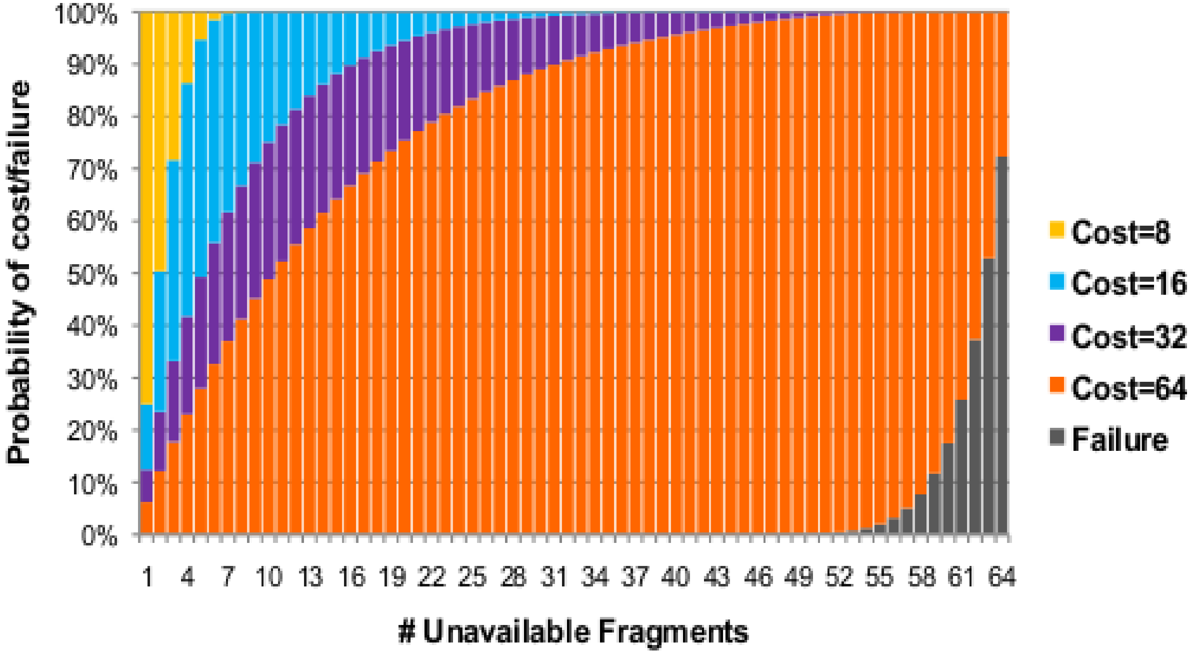
f4
f3
f2
f1
b7
b6
b5
b4
b3
b2
b1
k
k + h
4+3 Traditional Erasure Code
Probability of cost/failure
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
1
2
3
4
5
6
7
Unavailable Fragments
Failure
Cost=4
Cost=2



f4
f3
f2
f1
b7
b6
b5
b4
b3
b2
b1
k
k + h
4+3 Hierarchical Code
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
1
2
3
4
5
6
7
Unavailable fragments

Generalizing The Concept

- If we take a 64+64 traditional linear code and we apply the same idea hierarchically...
- If we set the hierarchy differently we obtain a different trade-off.

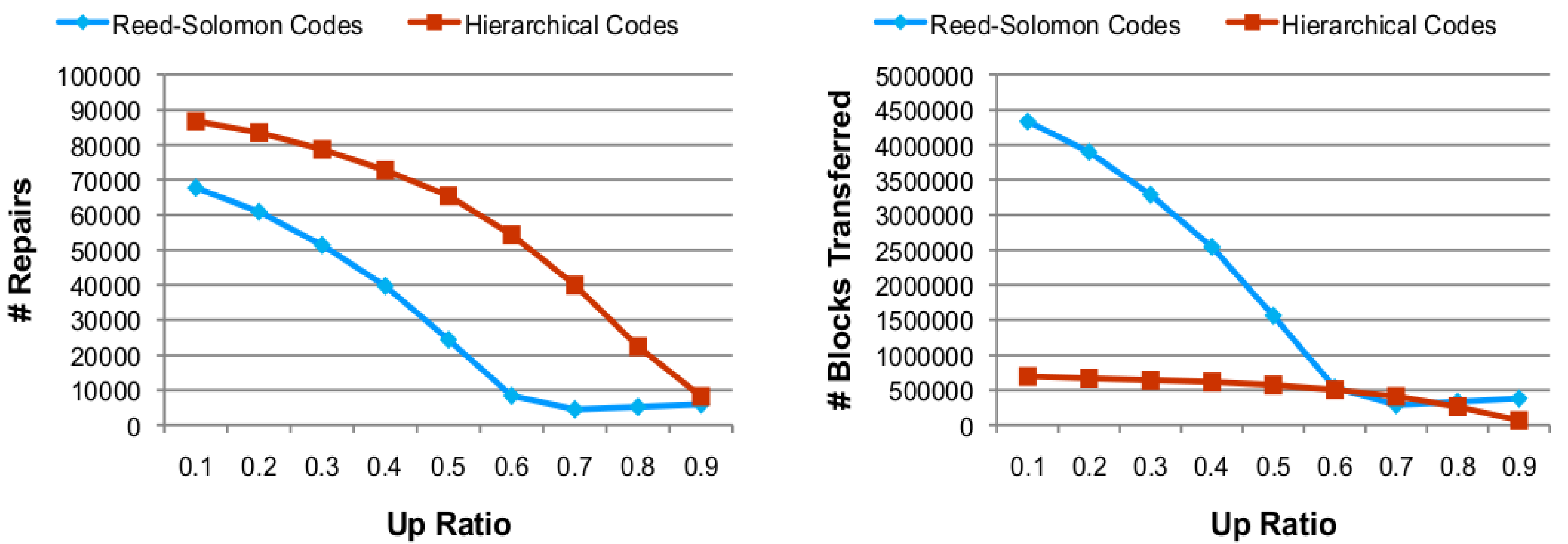


Experiments

Synthetic Data

- An event-driven simulator.
- They compared a 64+64 Reed-Solomon code (linear code) with one instance of a 64+64 Hierarchical code.
- They generated synthetic peer behavior with exponentially distributed uptimes, downtimes and lifetimes.
- As a general rule, the smaller is the up-ratio the higher the number of repairs.

Synthetic Data Results



Real Data

- PlanetLab traces consist in 669 nodes monitored for 500 days.
- KAD traces consist in the availability of about 6500 peers in the KAD network for about 5 months.

Real Data Results

- PlanetLab

	Number of Repairs	Number of blocks transferred
Reed-Solomon Code	472	30208
Hierarchical Code	637	4624

- KAD

	Number of Repairs	Number of blocks transferred
Reed-Solomon Code	765	48960
Hierarchical Code	3888	39710

Conclusion

Conclusion

- They proposed a new class of erasure codes called Hierarchical Codes.
- They aim at coupling the communication efficiency of replication with the storage efficiency of coding.
- Experiments showed that Hierarchical Codes require more repairs, but those repairs are so cheap that the resulting communication cost is smaller.

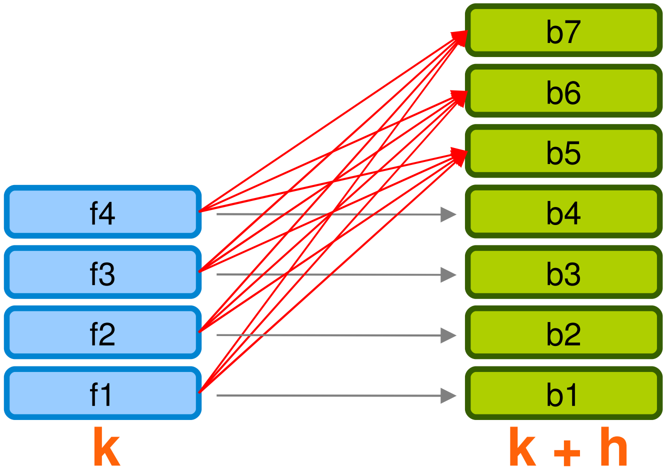
More Detail About Coding

Linear Codes

- f_i: i^{th} fragment
- b_i:i^{th} fragment
- $c_{i,j}$: coefficients

$$b_i = \begin{cases} f_i & i \leq k \\ \sum(c_{i,j} \times f_j) & k < i \leq k + h \end{cases}$$

- Any 4 of these 7 fragments can reconstruct the original file if the coefficients are linearly independent.



Linear Codes

$$B = C' F \quad \Rightarrow \quad F = S^{-1} B_S$$

- If any sub-matrix S built using k rows from C' is invertible, then the original fragments can be always reconstructed by $F = S^{-1}B_S$.
- B_S: The k-long subvector of B, corresponding to the coefficients chosen in S.
- If this property is satisfied, the code obtained is a (k,h)-code.

Coefficient Matrix

- Reed-Solomon Codes
- Random Linear Codes

Reed-Solomon

- $I_{k,\,k}$: Indentity matrix.
- $C_{h,\,k}$: Coefficient Matrix.

$$B = \begin{pmatrix} I \\ C \end{pmatrix} F = C' \, F$$

- If k = 2 and h = 3

$$B = \begin{pmatrix} I \\ C \end{pmatrix} F = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \\ c_{3,1} & c_{3,2} \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ c_{1,1}f_1 + c_{1,2}f_2 \\ c_{2,1}f_1 + c_{2,2}f_2 \\ c_{3,1}f_1 + c_{3,2}f_2 \end{pmatrix}$$

Reed-Solomon Codes

- Define the matrix C as a h × k Vandermonde matrix.
- $c_{i,j} = a_i^{j-1}$

$$V = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \dots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \dots & \alpha_2^{n-1} \\ 1 & \alpha_3 & \alpha_3^2 & \dots & \alpha_3^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_m & \alpha_m^2 & \dots & \alpha_m^{n-1} \end{bmatrix}$$

Reed-Solomon Codes

- $k = 2$
- $h = 3$
- $c_{i,j} = j^{i-1}$

$$B = \begin{pmatrix} I \\ C \end{pmatrix} F = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_1 + f_2 \\ f_1 + 2f_2 \\ f_1 + 3f_2 \end{pmatrix}$$

Reed-Solomon Codes

$$B = \begin{pmatrix} I \\ C \end{pmatrix} F = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_1 + f_2 \\ f_1 + 2f_2 \\ f_1 + 3f_2 \end{pmatrix}$$

$$S = \begin{pmatrix} 1 & 0 \\ 1 & 3 \end{pmatrix}$$

$$S^{-1} B_S = \begin{pmatrix} 1 & 0 \\ -1/3 & 1/3 \end{pmatrix} \begin{pmatrix} f_1 \\ f_1 + 3f_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = F$$

Random Linear Code

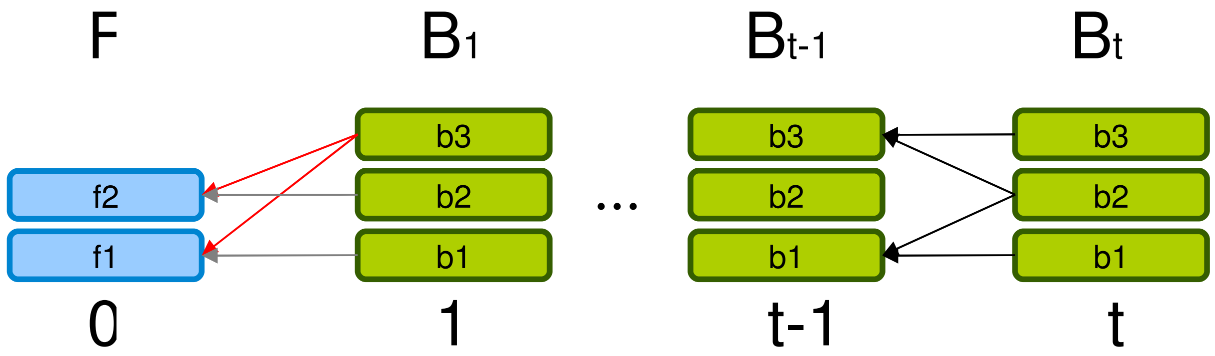
- It is shown that a $k \times k$ random matrix S in $GF(2^q)$ is invertible with a probability which depends only on the field size and will increase by the size increasing.
  - $GF(2^q)$: Galois Field, where the elements can be expressed by q-bit words.

- If $q \geq 16$, the probability can be considered practically 1.

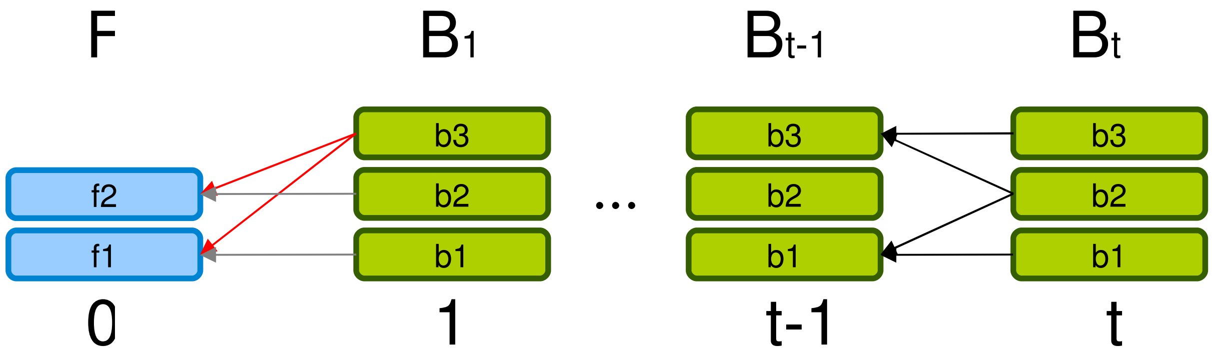
- This means that any $k \times k$ sub-matrix of C' is invertible and that the property of a (k,h)-code is provided.

Information Flow Graph (Code Graph)

- Represents the evolution of the stored data through time.



Information Flow Graph (Code Graph)

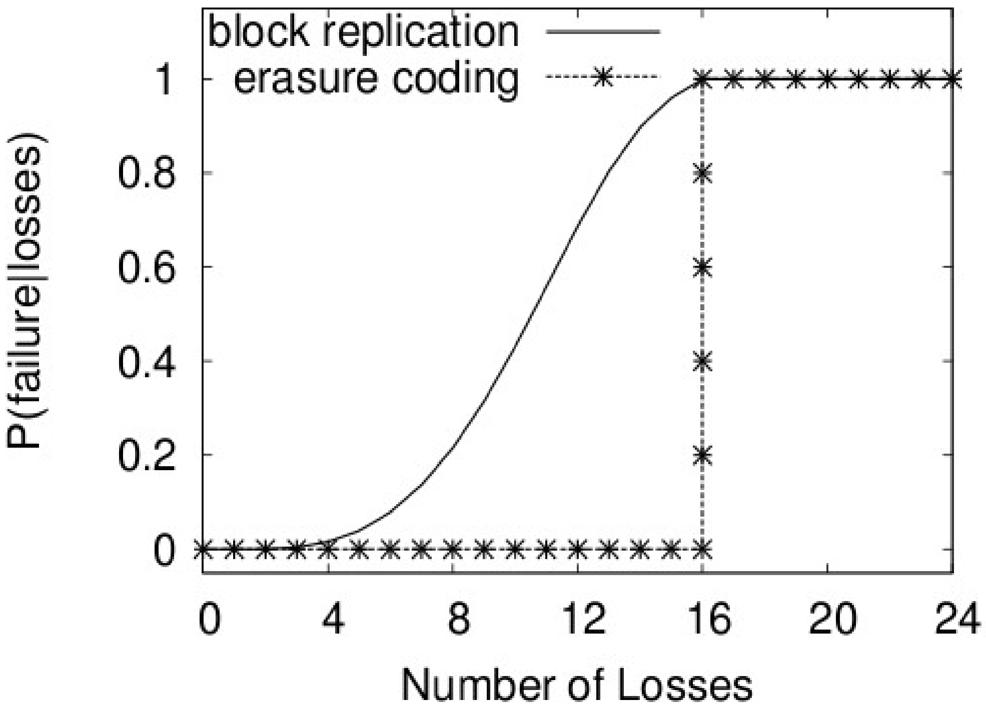


- Proposition 1: At any time t, any of all the possible selections of k nodes B_t^k is sufficient to reconstruct the original fragments only if the disjoint paths condition is provided at time step t = 1 and the repair degree $d \geq k$.

- A Random linear code provides this condition
  - By design any node in B1 is connected to all the source nodes in F.

Block Replication vs Linear Codes

- k = 8, h = 16 and R = 3



- Block replication: d = 1
- Linear codes: d = k

Question?

Is there a design space between these two limits that can be explored to find a better trade-off between storage efficiency and repair degree?

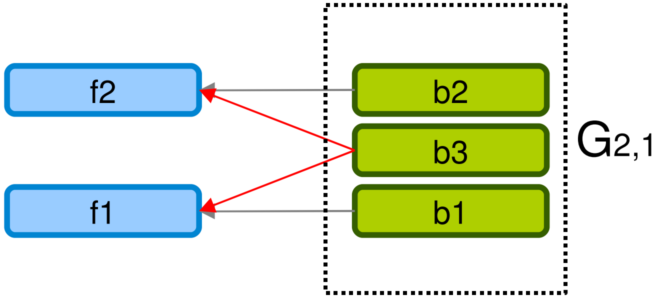
Hierarchical Codes

Hierarchical Code Graph – Step 1

- Choose k_0 and h_0 and build (k_0, h_0)-code:

$$b_i = \begin{cases} f_i & i \leq k \\ \sum (c_{i,j} \times f_j) & k < i \leq k + h \end{cases}$$

- $k_0 = 2$
- $h_0 = 1$

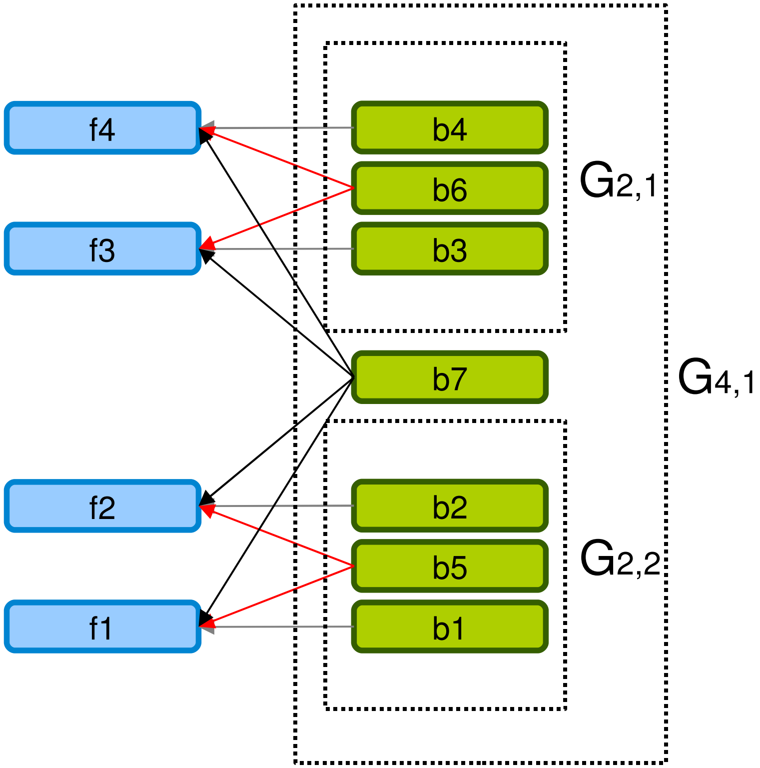


Hierarchical (2, 1)-code

- The generated group denoted as $G_{d0,1}$, where $d_0 = k_0$.

Hierarchical Code Graph – Step 2

- Choose g_1 and h_1.
- Replicate $G_{d0,1}$ for g_1 times.
  - g_1 groups denoted as $G_{d0,1}$, ..., $G_{d0,g}$.
- Then add other h_1 redundant blocks.
  - Combining all the existing $g_1 k_0$ original fragments F.
- The new group denoted as $G_{d1,1}$,
  - Hierarchical (d_1, H_1)-code,
  - $H_1 = g_1 h_0 + h_1$
  - $d_1 = g_1 k_0 = g_1 d_0$
- $g_1 = 2$
- $h_1 = 1$



Hierarchical (4, 3)-code

Hierarchical Code Graph – Step 3

- Repeat Step 2 several times.

- $H_s = g_s H_{s-1} + h_s$
- $d_s = g_s d_{s-1}$

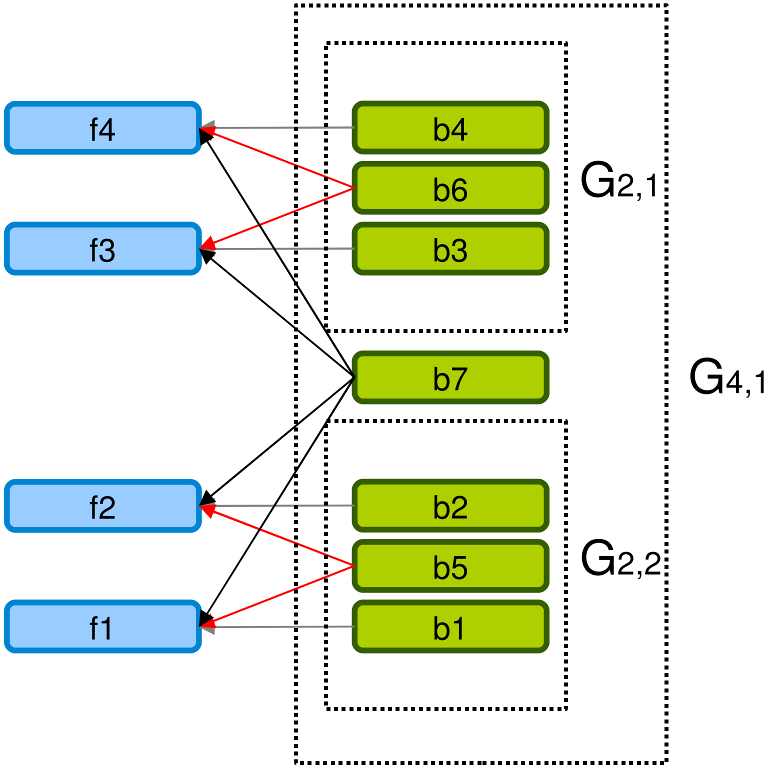
Hierarchical Code – Reliability

- Proposition 2: Consider B_k, a set of k blocks in the code graph of a hierarchical (k,h)-code.
- If the nodes in B_k are chosen fulfilling the following condition:

$$|G_{d,i} \cap B_k| \leq d, \quad \forall G_{d,i} \text{ belonging to the code}$$

- Then the nodes in B_k are sufficient to reconstruct the original fragments.

Hierarchical Code – Reliability



Hierarchical (4, 3)-code

- P(failure | I) = 0.23

Hierarchical Code – Repair Degree

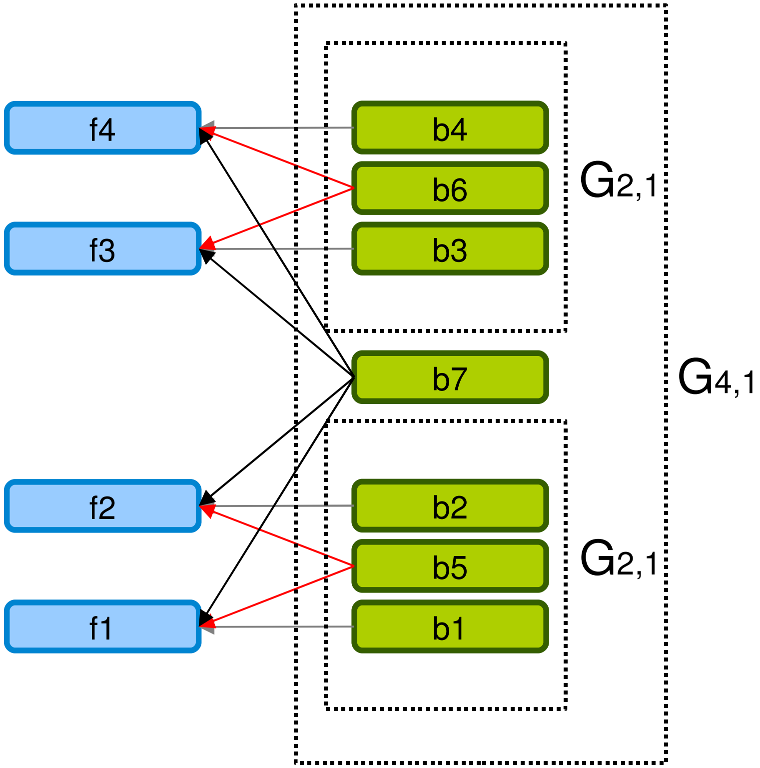
- Proposition 3: Consider a node b repaired at time step t. Denote as G(b) the hierarchy of groups that contains b and as R(b) the set of nodes in B_{t-1} that have been combined to repair b
- If If ∀t and ∀b, R(b) fulfills the following conditions:

$$|G_{d,i} \cap R(b)| \le d, \ \ \forall G_{d,i} \text{ belonging to the code}$$
$$\exists G_{d,i} \in G(b): R(b) \subseteq G_{d,i}, \ |R(b)| = d$$

- Then Then the code does not degrade, i.e. preserve the properties of the code graph expressed in Proposition 2.

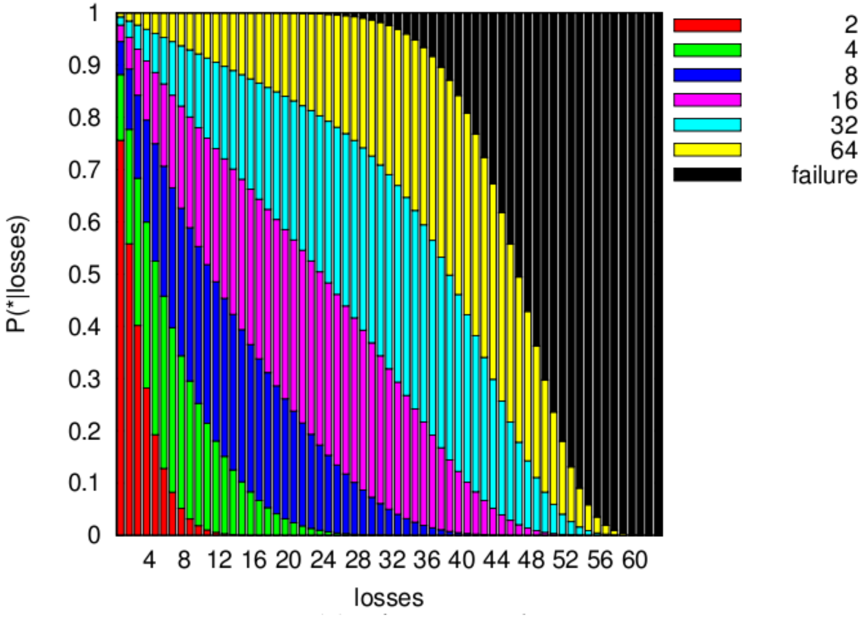
Hierarchical Code

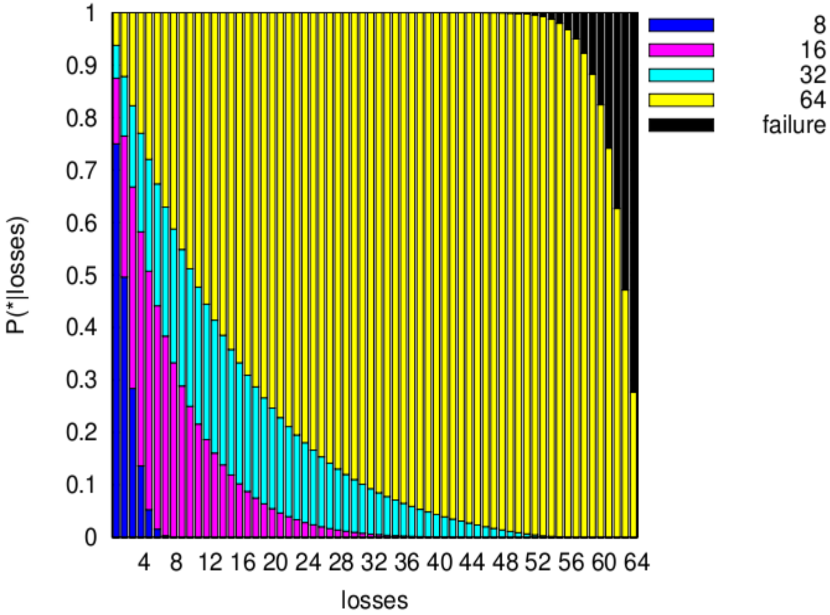
	l (losses)		
	1	2	3
$P(d=2\|l)$	0.86	0.42	0
$P(d=4\|l)$	0.14	0.58	0.77
$P(failure\|l)$	0	0	0.23



Hierarchical (4, 3)-code

Hierarchical Code





Question?