

# REPAIRING MULTIPLE FAILURES WITH COORDINATED AND ADAPTIVE REGENERATING CODES

Nicolas Le Scouarnec (Technicolor)

Anne-Marie Kermarrec (INRIA) et Gilles Straub (Technicolor)



NetCod  
July 2011,  
Beijing



[www.leaderstudio.net](http://www.leaderstudio.net)

# Distributed Storage Systems

---

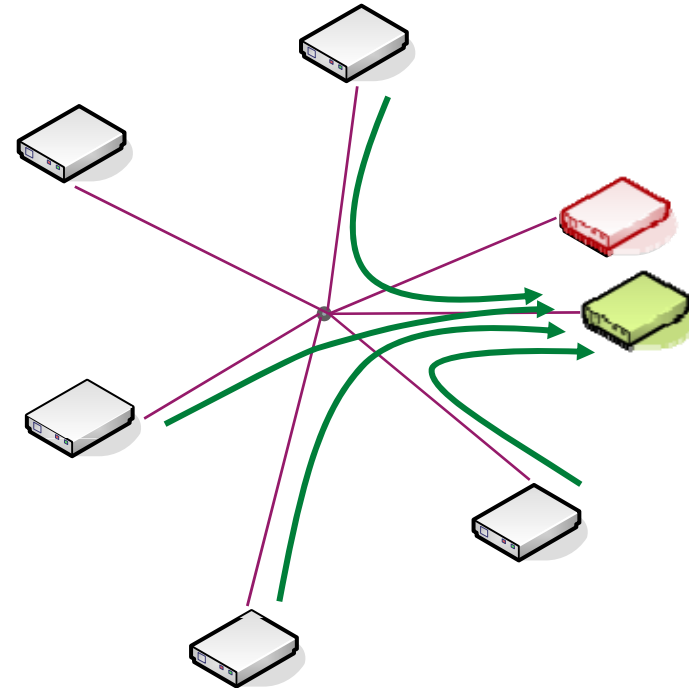
Aggregate storage space

Store file securely (redundancy)

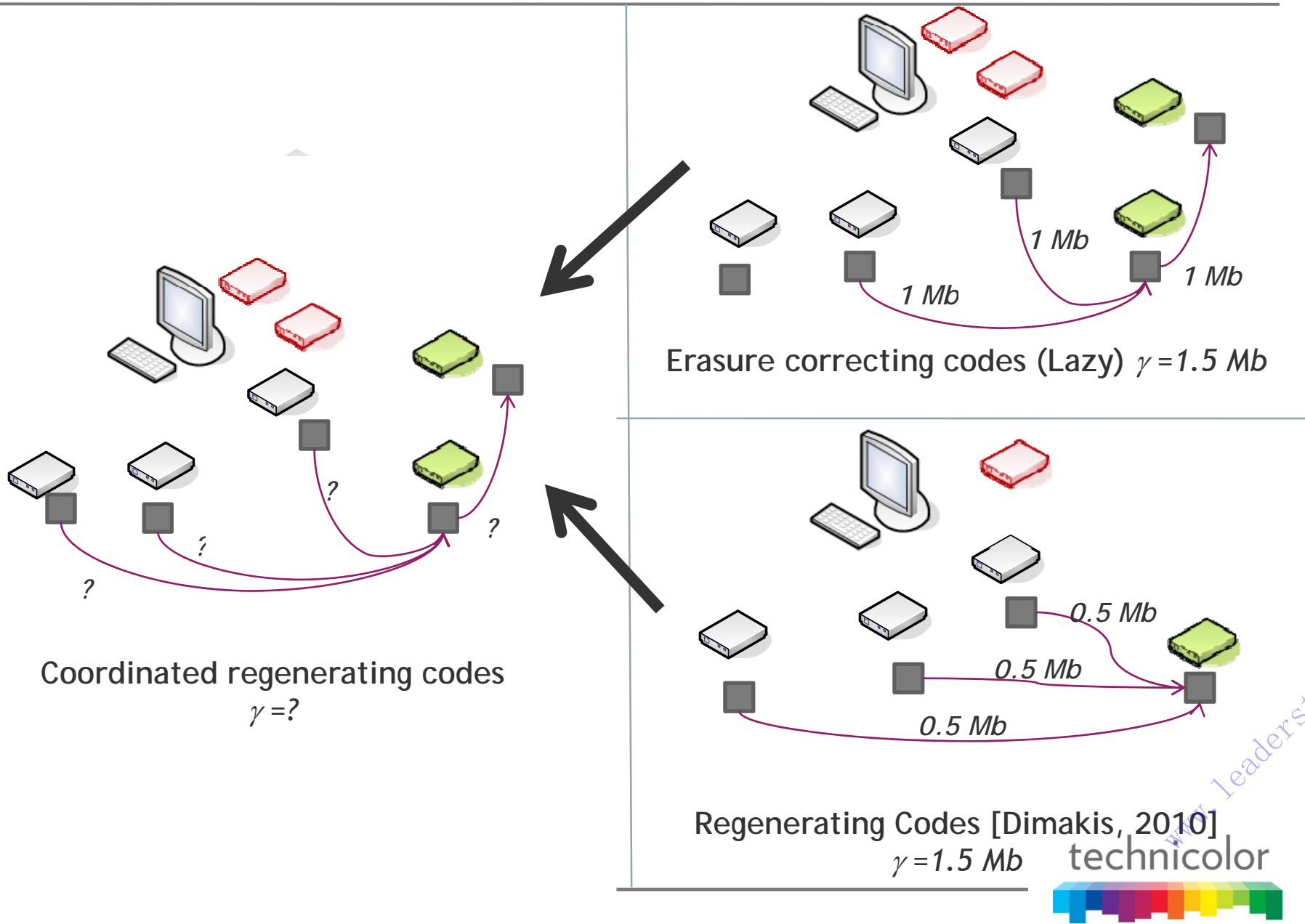
Long lifetime (self heal)

Limiting factors

- Storage (Code based redundancy)
- Bandwidth (Efficient repair)



# Repairing a Distributed Storage System



# Simultaneous repairs for regenerating codes

---

## Limits of regenerating codes

- Single failures
- Static system (parameters set for eternity)

## Open questions

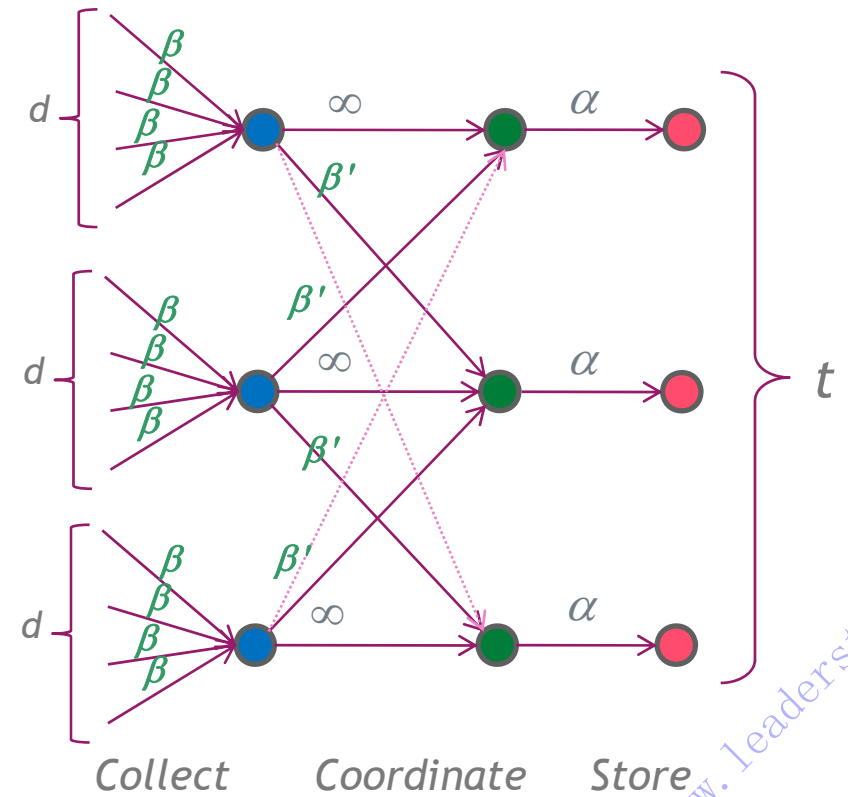
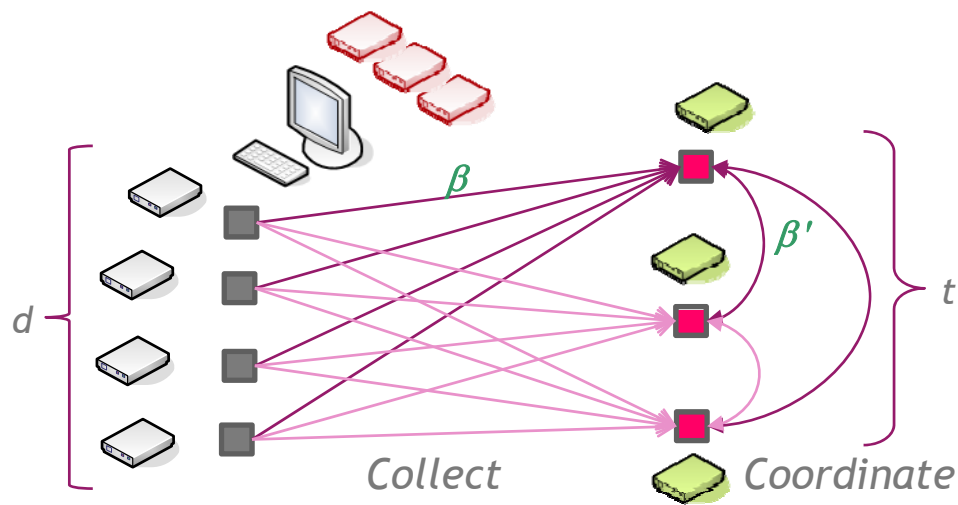
- Optimal amounts of information for multiple repairs ?
- Can we reduce costs by deliberately delaying repairs ?
- Can we have codes that adapt to dynamic settings ?

# Repair algorithm

Repair cost

$$\gamma = d\beta + (t-1)\beta'$$

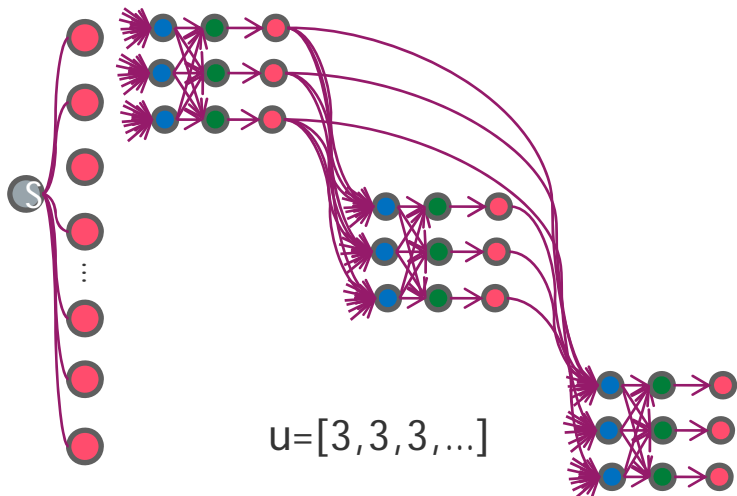
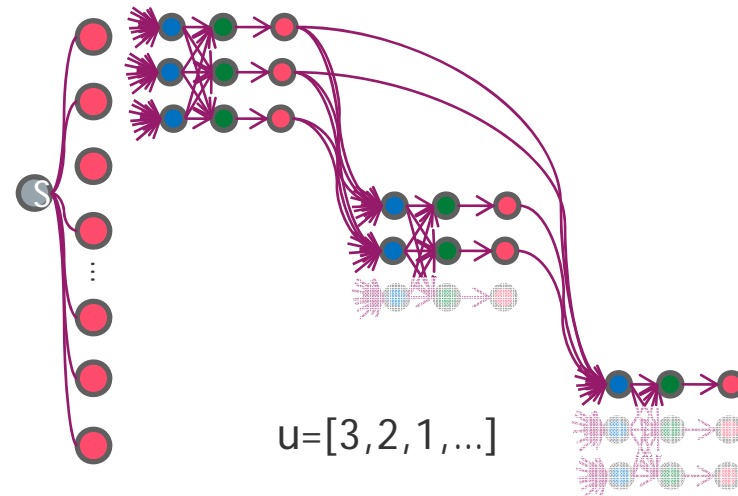
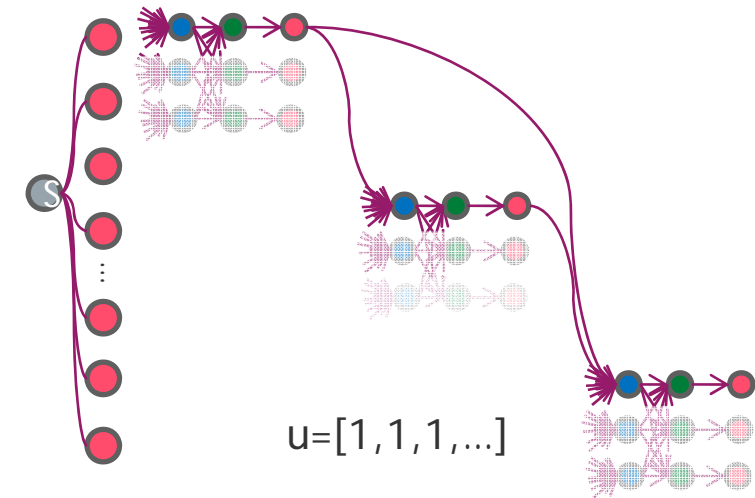
d	Number of live devices contacted during repairs
k	Number of devices needed to recover
t	Number of devices being repaired



www.leaderstudio.net

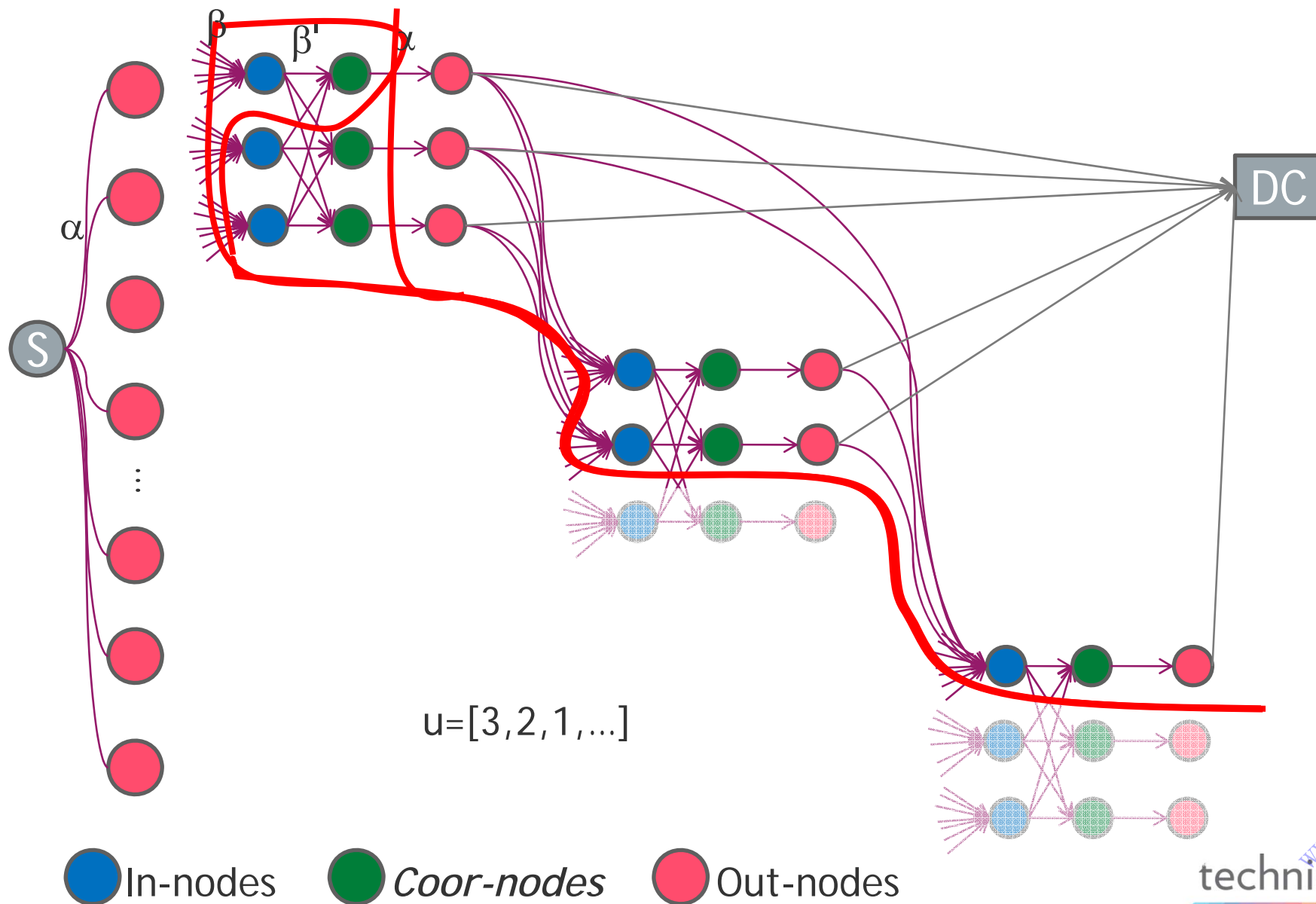


# Scenarios



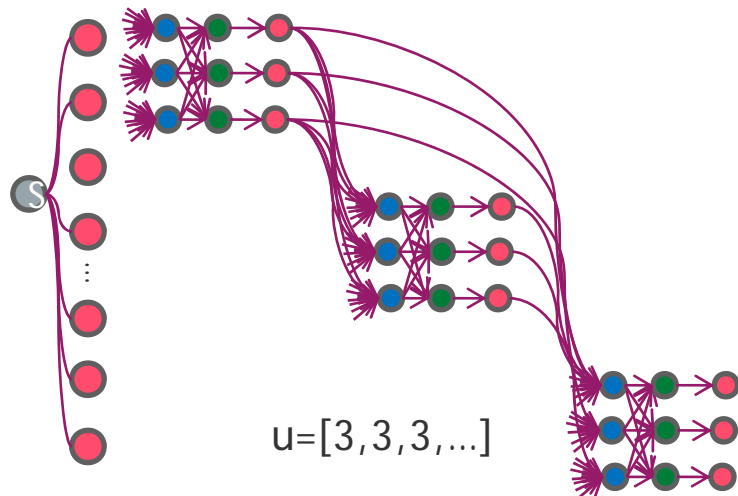
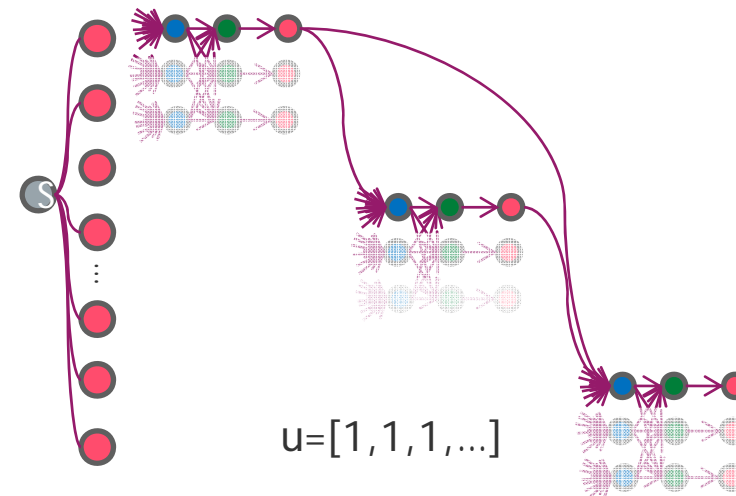
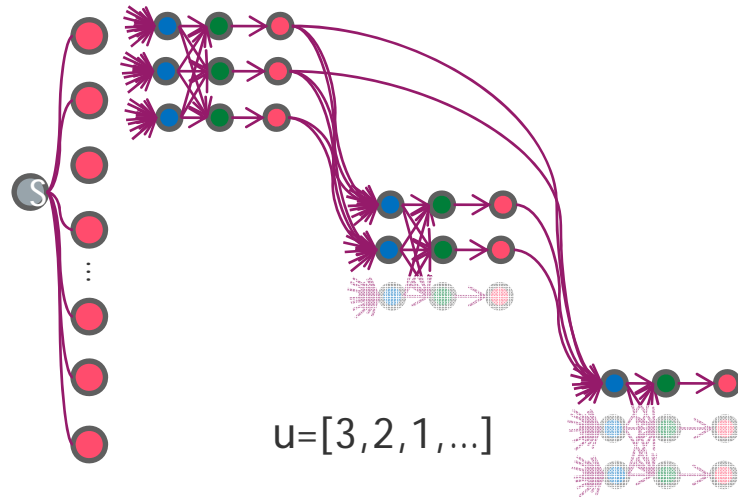
[www.leaderstudio.net](http://www.leaderstudio.net)

# Finding the minimum cut



● In-nodes   
 ● *Coord-nodes*   
 ● Out-nodes

# Finding the worst scenario



## Closed forms for MSCR and MBCR

- Consider only constraints coming from  $u=[1,1,1,\dots]$  and  $u=[t,t,t,\dots]$
- Show that values are correct for all  $u$

## Closed form for Interior Points

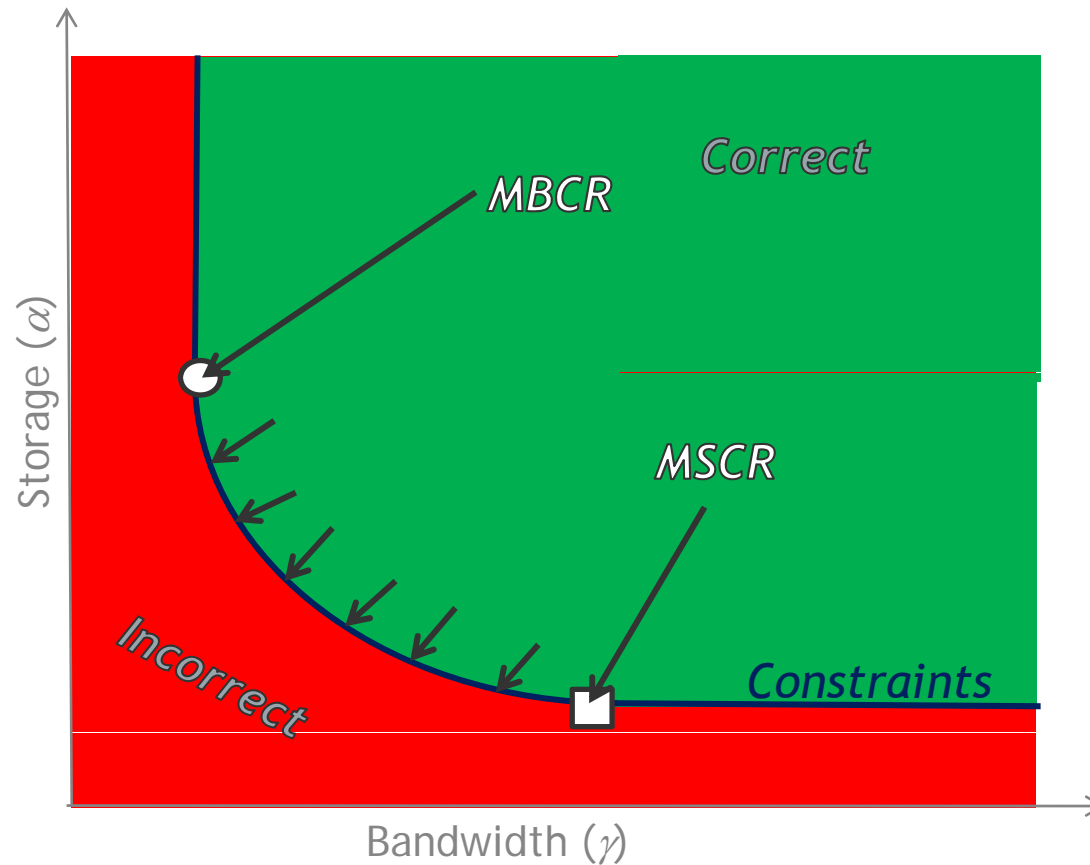
- Numerical values obtained by optimisation over all  $u$
- Open question,  $u=[1,1,1,\dots]$  and  $u=[t,t,t,\dots]$  are not always the worst ones

[www.leaderstudio.net](http://www.leaderstudio.net)



# How to set amounts of information ( $\alpha$ , $\beta$ , $\beta'$ ) ?

---



[www.leaderstudio.net](http://www.leaderstudio.net)



# MSCR: Minimum Storage Coordinated Regenerating Codes

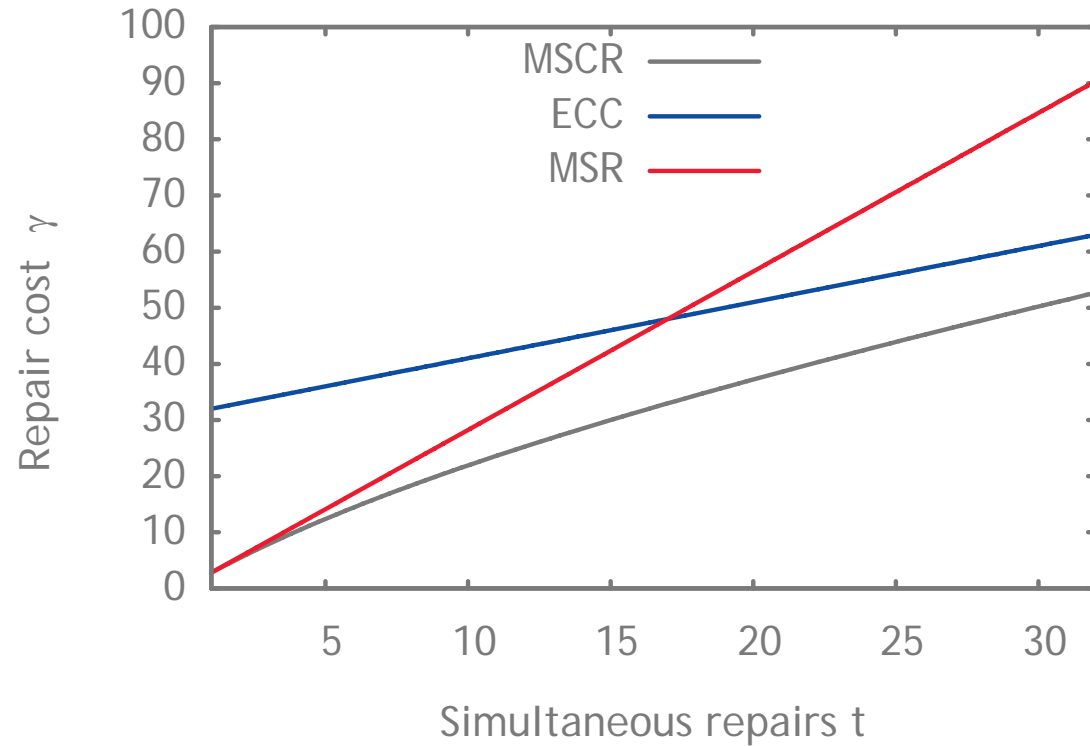
## System

- $k=32$  ,  $d=48$ ,  $t=1..32$

## Advantages

- A single repair scheme for all  $t$

## Optimal



$$\alpha = \frac{M}{k} \quad \beta = \frac{M}{k} \frac{1}{d-k+t} \quad \beta' = \frac{M}{k} \frac{1}{d-k+t}$$

Similar results by Hu et al. in JSAC 2010, with  $d=n-t$

# MBCR: Minimum Bandwidth Coordinated Regenerating Codes

## System

- $k=32$  ,  $d=48$ ,  $t=1..32$

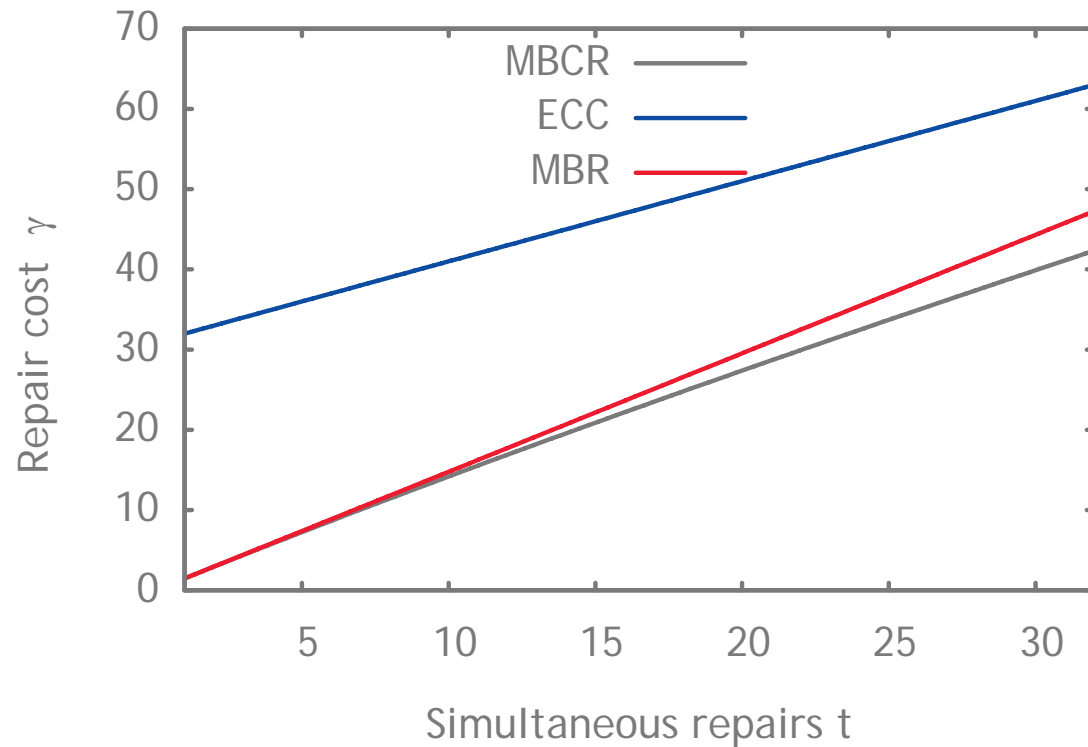
## Advantages

- A single repair scheme for all  $t$

## Optimal

$$\alpha = \frac{M}{k} \frac{2d - t + 1}{2d - k + t}$$

$$\beta = \frac{M}{k} \frac{2}{2d - k + t} \quad \beta' = \frac{M}{k} \frac{1}{2d - k + t}$$



www.leaderstudio.net



# Deliberately delaying repairs ?

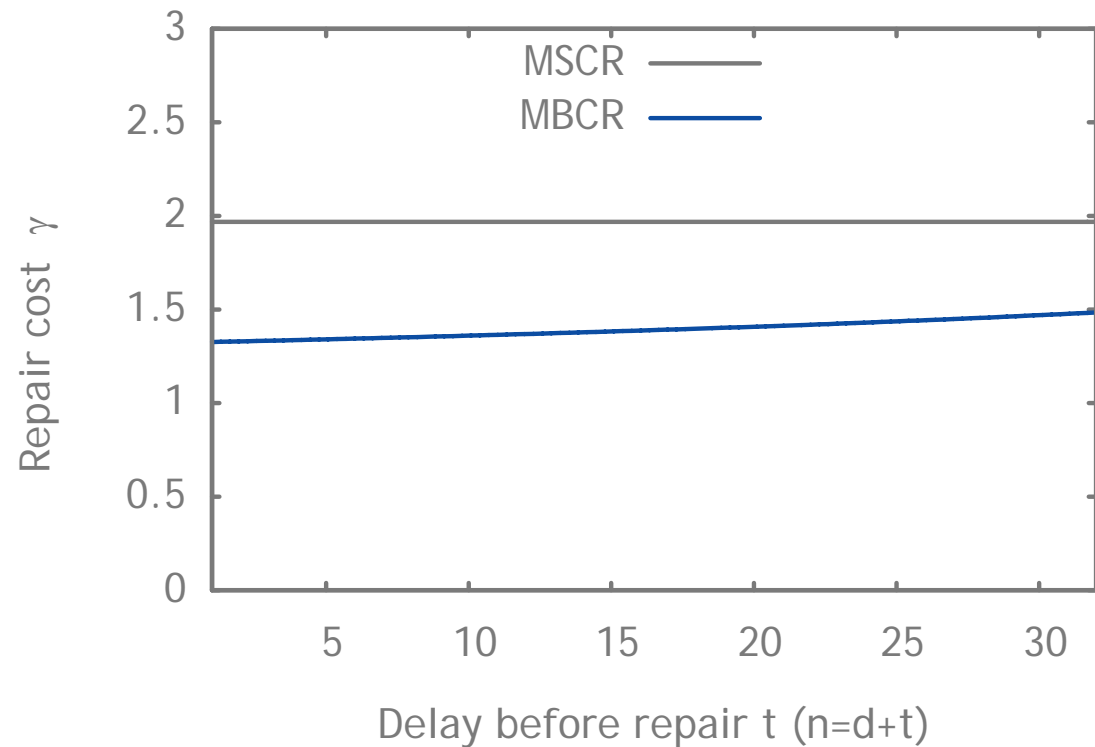
## Contradiction

- The longer we wait, the lower  $d$  (cost increase)
- The longer we wait, the higher  $t$  (cost decrease)

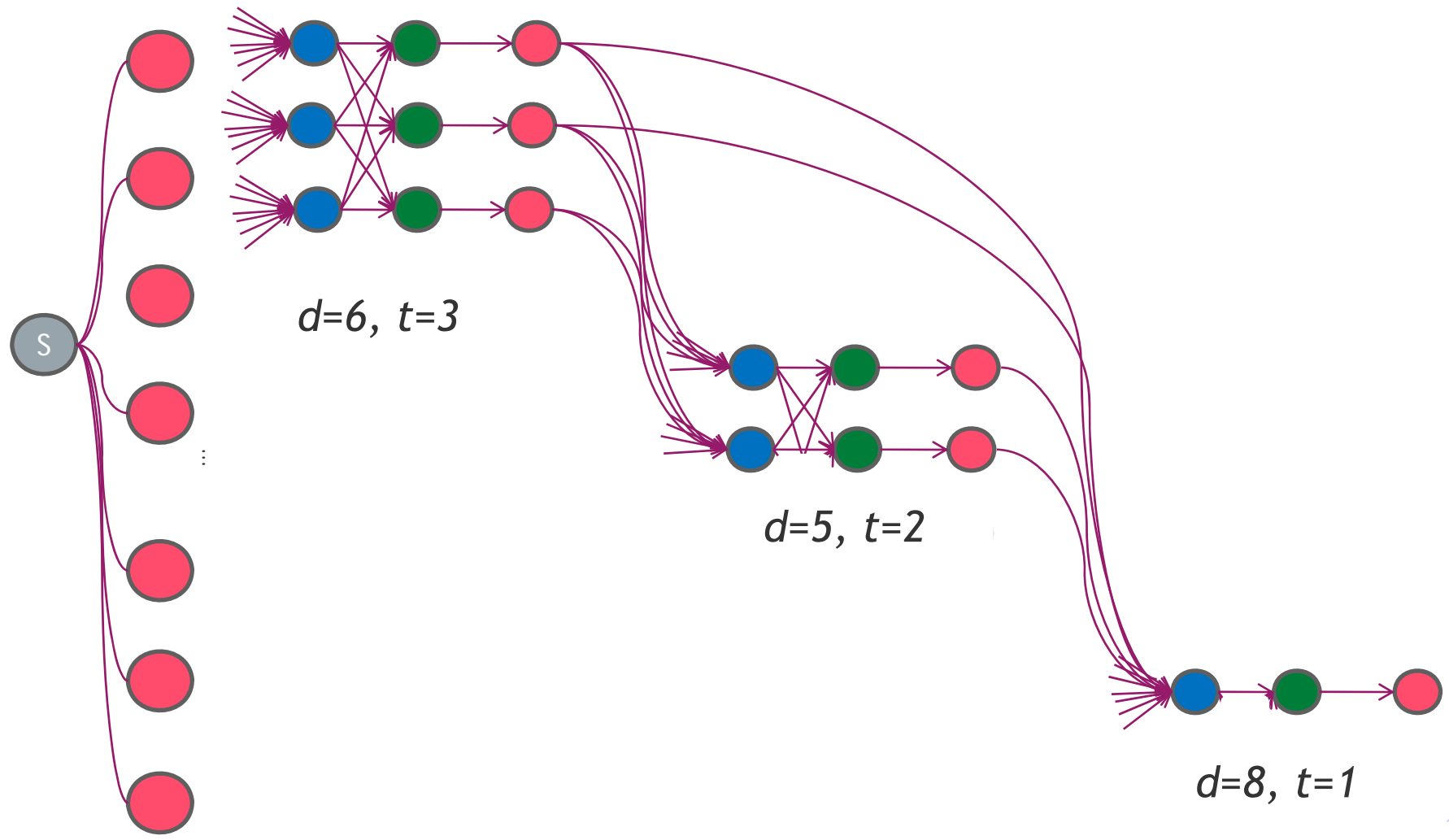
Does it make sense to deliberately delay ?

- MSCR => any
- MBCR => no

*Classical* regenerating codes still the best



# Adaptive Regenerating Codes



*Not allowed in classical regenerating codes*

[www.leaderstudio.net](http://www.leaderstudio.net)



# Adaptive Regenerating Codes

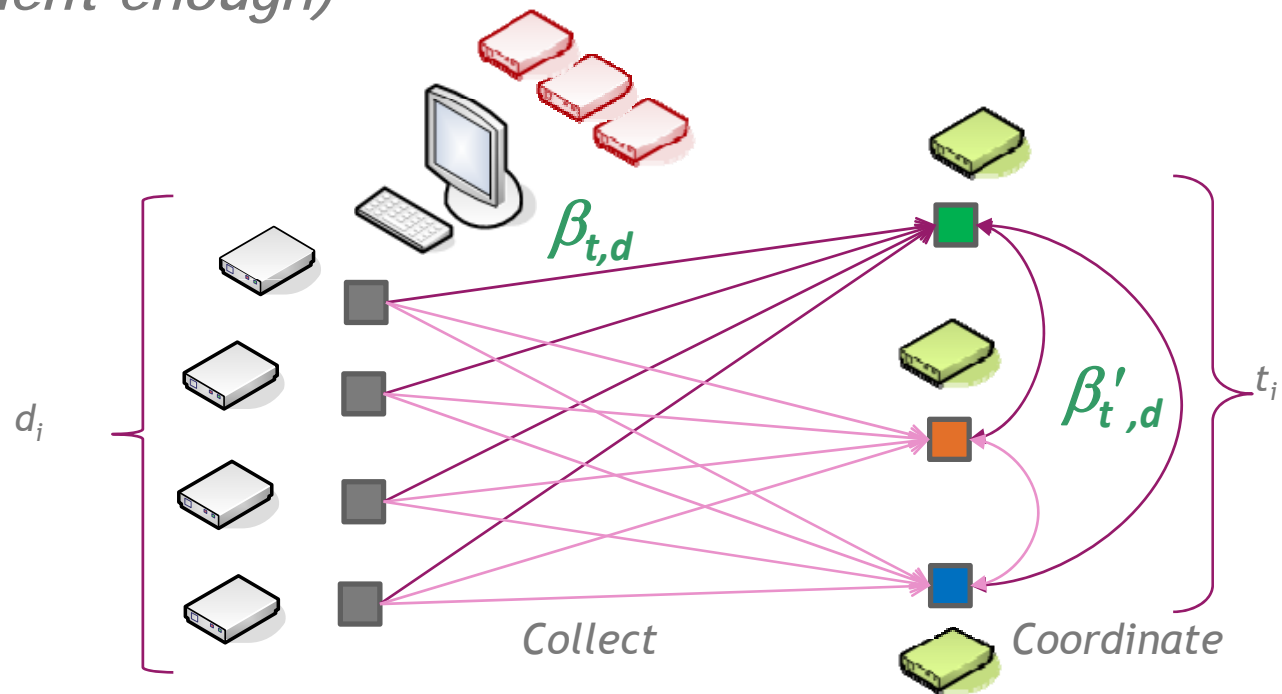
*Allowing  $t$  and  $d$  to change during the lifetime of the system*

*Only possible at the Minimum Storage point (otherwise repairs are not independent enough)*

$$\alpha = \frac{M}{k}$$

$$\beta_{t,d} = \frac{M}{k} \frac{1}{d-k+t}$$

$$\beta'_{t,d} = \frac{M}{k} \frac{1}{d-k+t}$$



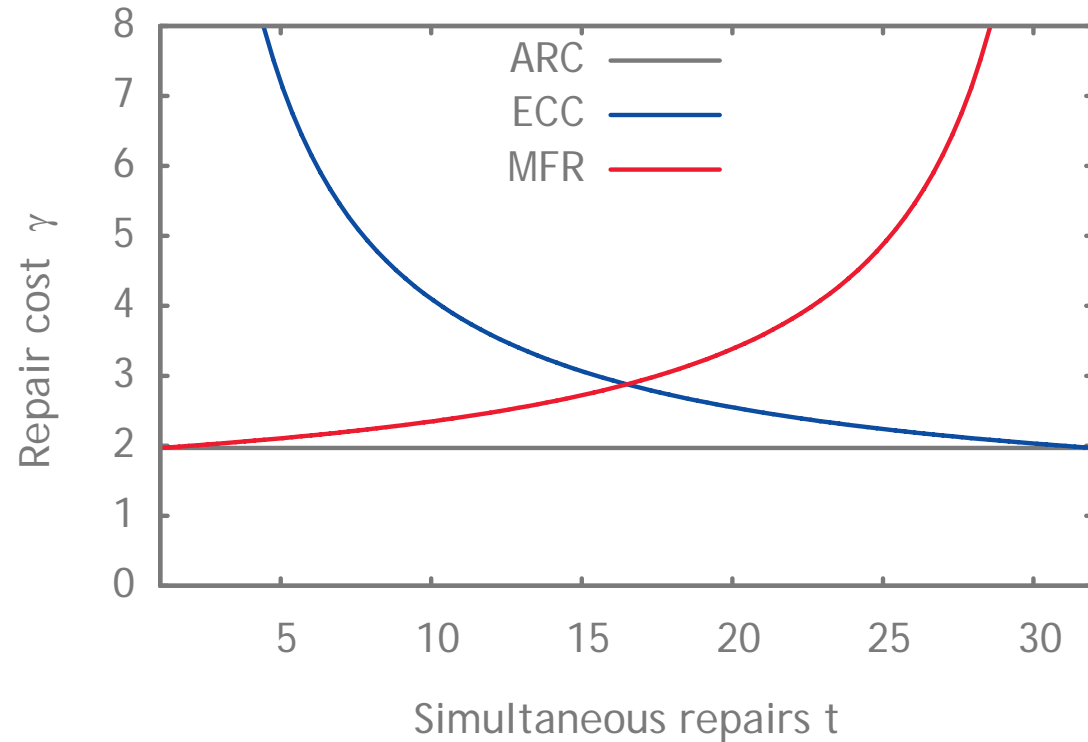
# Adaptive Regenerating Codes

## System

- Constant size  $n=d+t$
- $n=64$  ,  $k=32$

## Advantages of ARC

- Lower repair cost
- Easier implementation thanks to constant cost



# Conclusion

---

Optimal amounts of information for multiple repairs ?

**MSCR and MBCR**

Can we reduce costs by deliberately delaying repairs ?

**No**

Can we have codes that adapt to dynamic settings ?

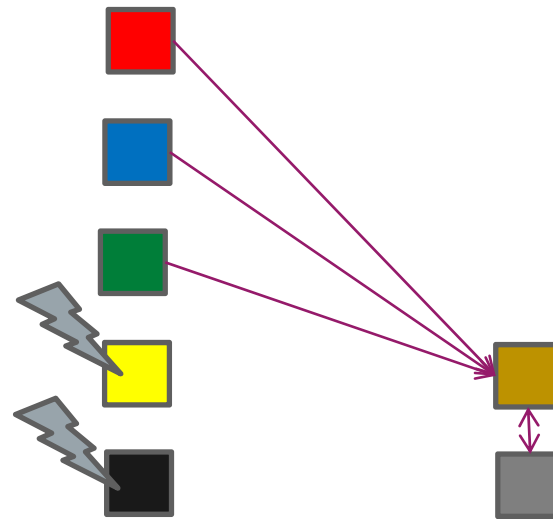
**Yes**

# Exact Regenerating Codes

---

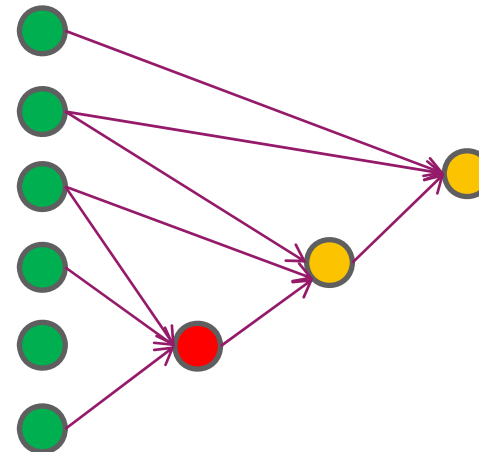
More interesting for

- Access without decoding (systematic codes)
- Access to a subset of the file (systematic codes)
- Implicit structure (no headers)

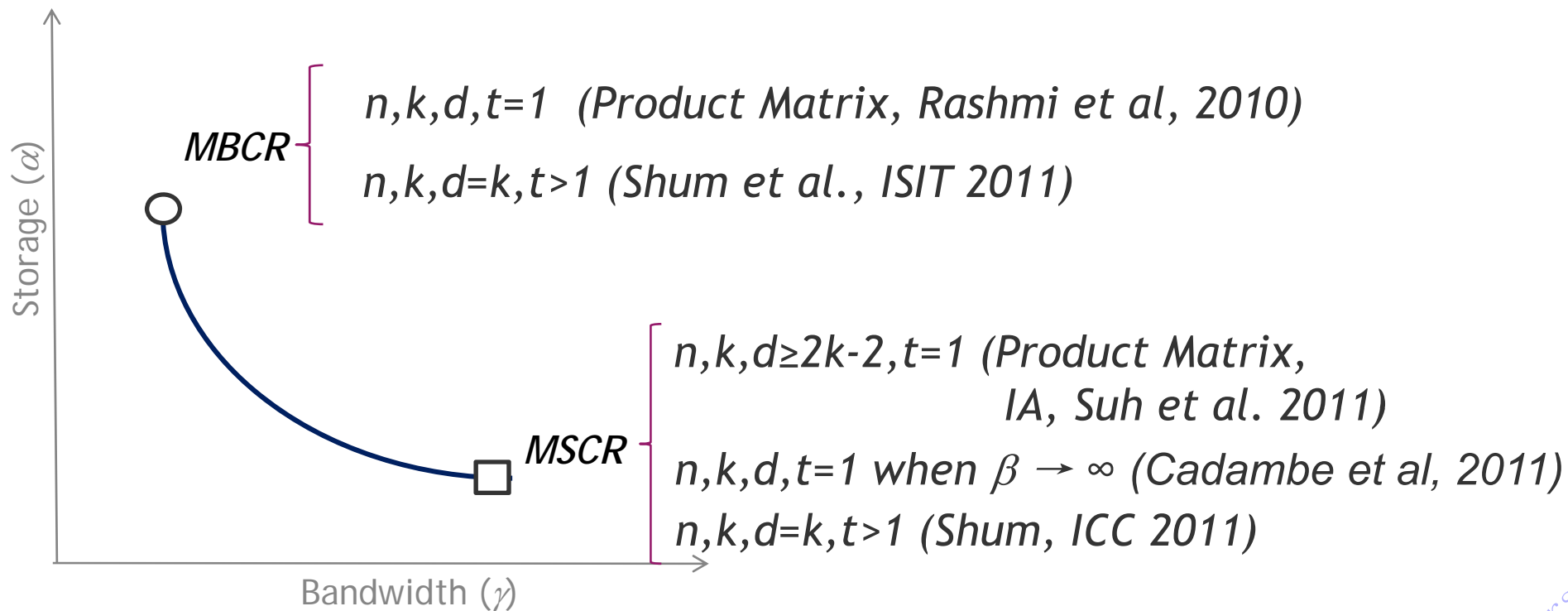


Better security for practical system

- Not sensible to pollution attacks
- Can use regular Secure Hashing/Signature
- Can prevent data destruction at repairs



# Exact Repair of Regenerating Codes



# REPAIRING MULTIPLE FAILURES WITH COORDINATED AND ADAPTIVE REGENERATING CODES

Nicolas Le Scouarnec (Technicolor)

Anne-Marie Kermarrec (INRIA) et Gilles Straub (Technicolor)



NetCod  
July 2011,  
Beijing

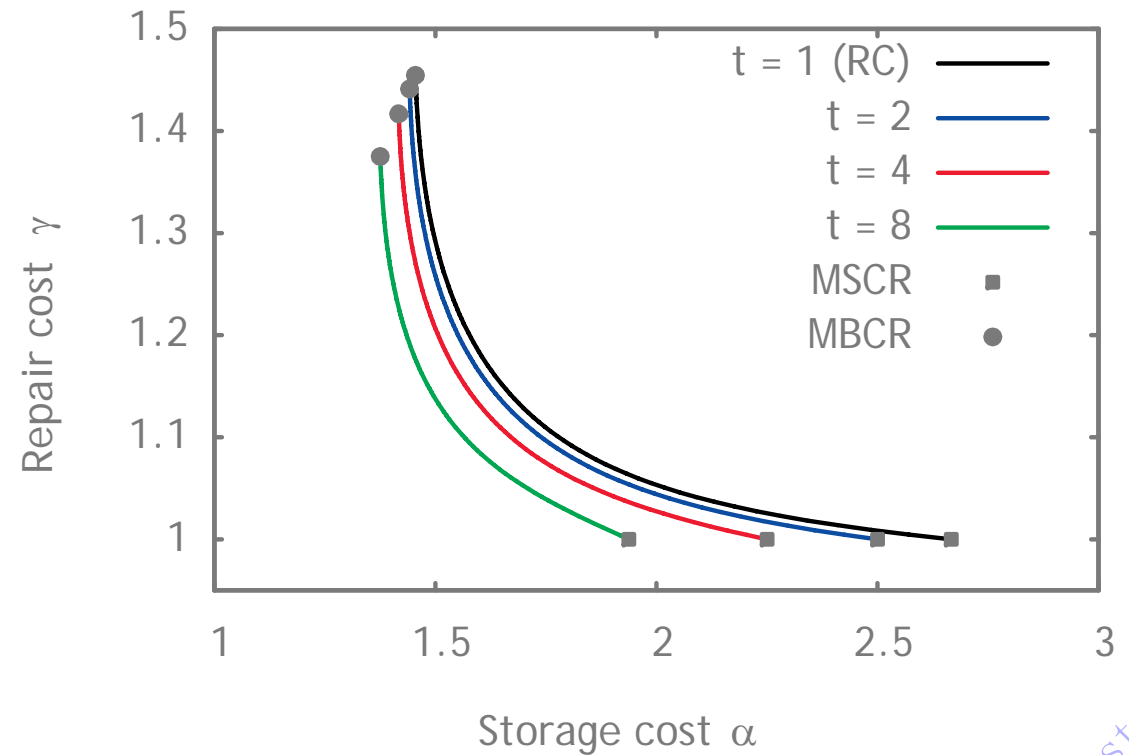


[www.leaderstudio.net](http://www.leaderstudio.net)

# General case

Minimize  $(\alpha, \gamma)$

Go beyond existing limit through coordination



$d=24, k=16, M=16$



# Storage and repair costs

Some figures

- Store a file of 32 MB
- Split in block of 1 MB
- Tolerate up to 10 failures

d=36	Live device for repairing
k=32	Needed devices for recovering
t=4	Devices being repaired

	Scheme	Storage cost (total)	Repair cost (per block)
	Replication	320 MB	1 MB
	ECC	42 MB	32 MB
	ECC (lazy)	42 MB	8.8 MB
Regenerating Codes	MSR	42 MB	7.2 MB
	MBR	76 MB	1.8 MB
	Our MSCR	42 MB	4.9 MB
Coordinated Regenerating Codes	Our MBCR	71 MB	1.7 MB

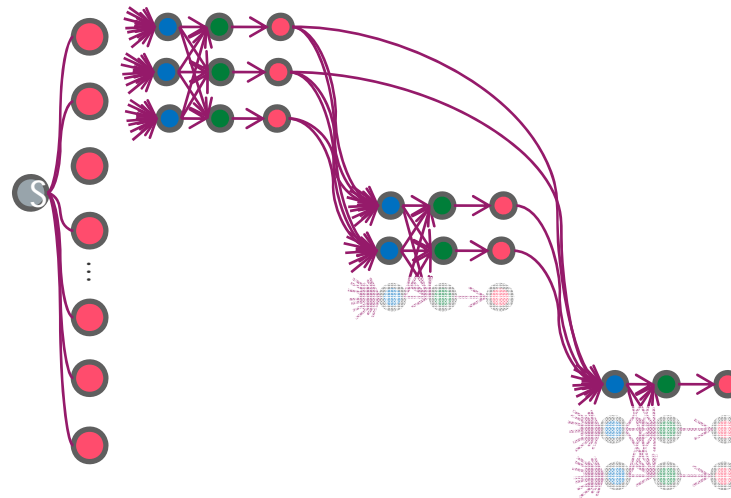
www.leaderstudio.net



# Constraints

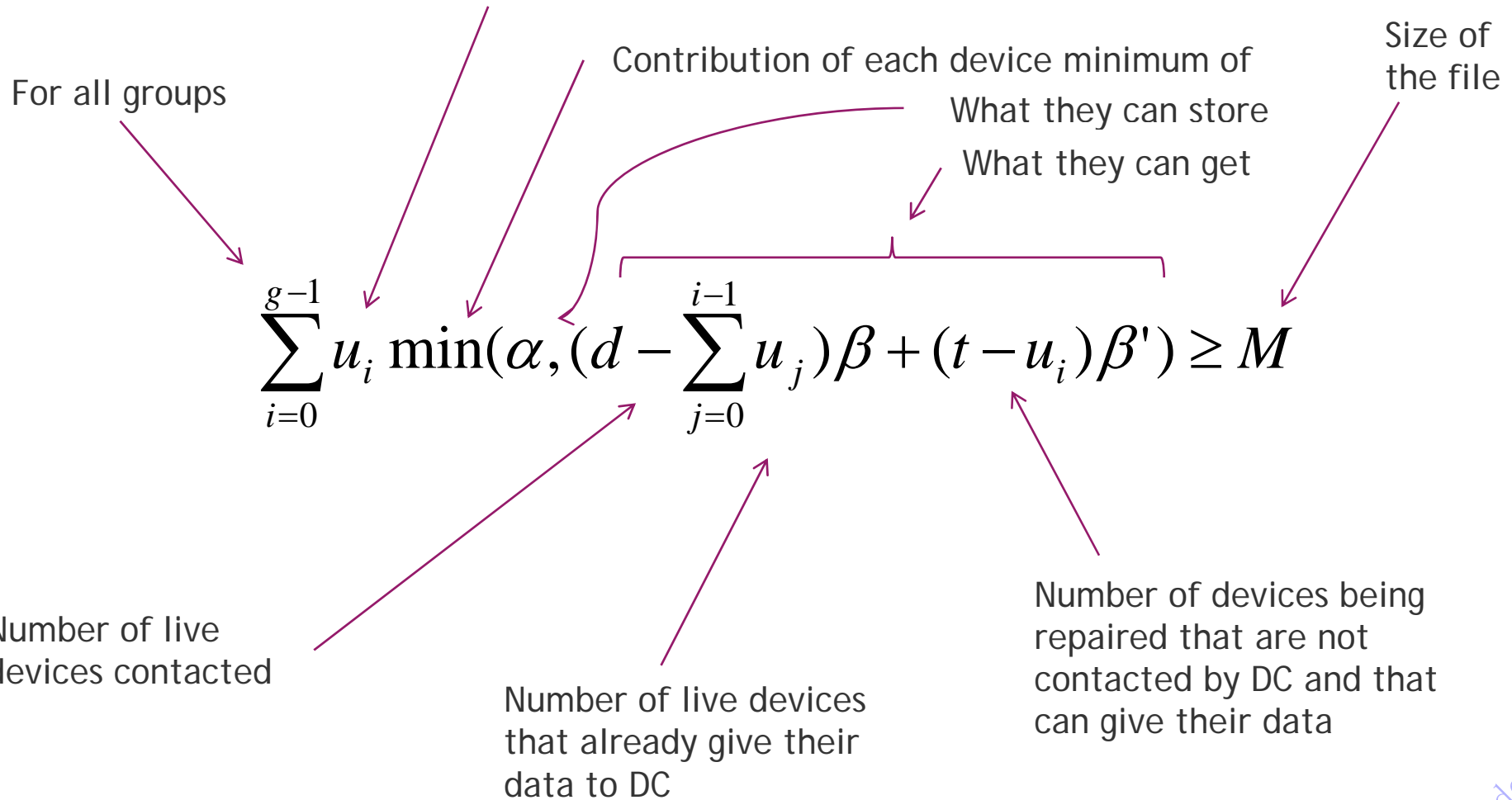
---

$$\sum_{i=0}^{g-1} u_i \min\left\{\alpha, \left(d - \sum_{j=0}^{i-1} u_j\right)\beta + (t - u_i)\beta'\right\}$$

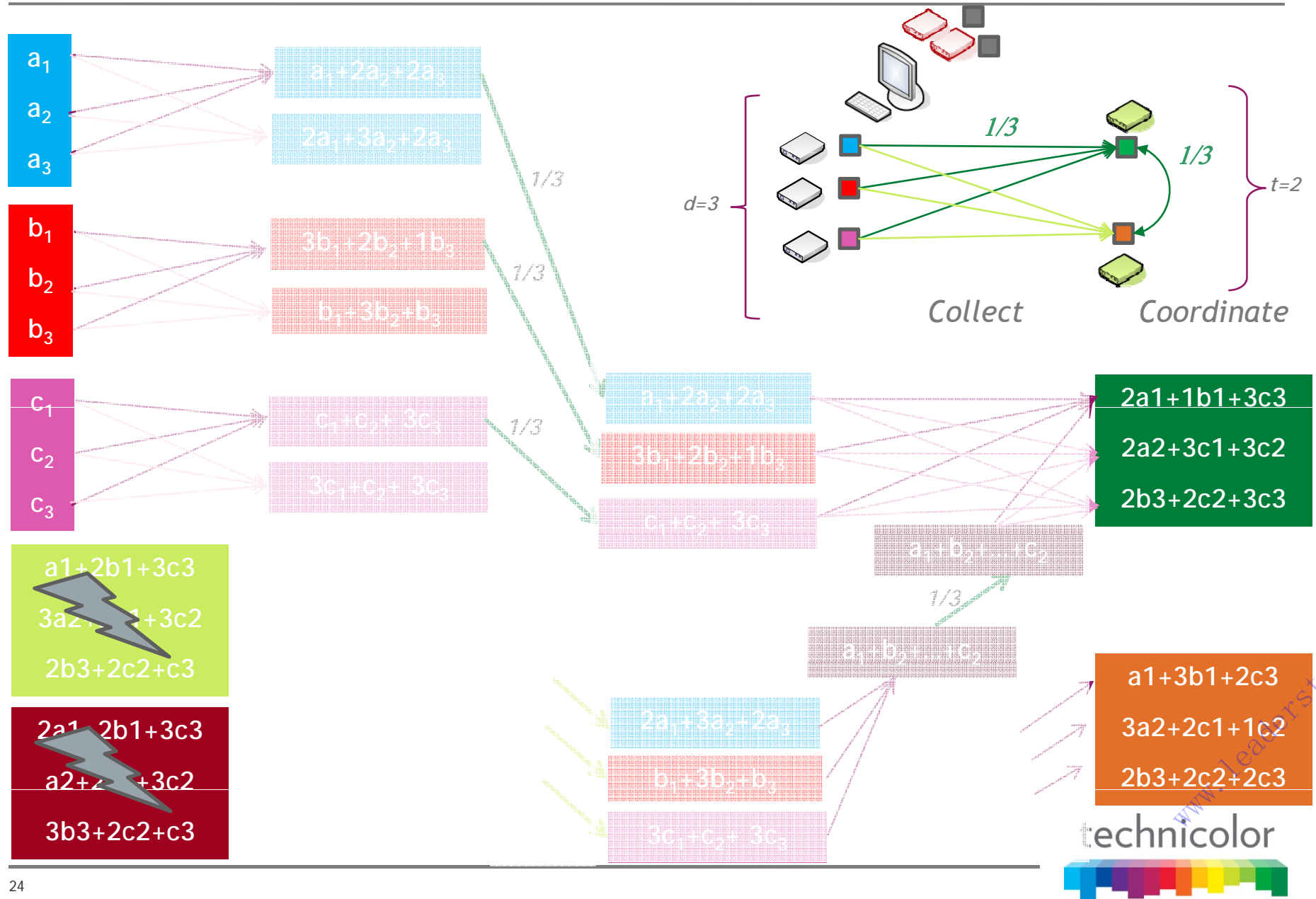


# Constraints

Number of devices contacted by DC per group



# Perspectives: Regenerating codes implementation



# Perspective: Regenerating codes implementation

