Network Coding for Distributed Storage Systems*

Presented by
Jayant Apte
ASPITRG
7/9/13 & 7/11/13

*Dimakis, A.G.; Godfrey, P.B.; Wu, Y.; Wainwright, M.J.; Ramchandran, K. "Network Coding for Distributed Storage Systems", Information Theory, IEEE Transactions on, On page(s): 4539 – 4551 Volume: 56, Issue: 9, Sept. 2010

Outline

- Part 1
  - Single Source Multi-cast Linear Network Coding
- Part 2
  - The repair problem
  - Reduction of repair problem to single source multicast network
  - Family of single source multi-cast networks arising from the reduction
  - A lower bound on min-cuts(i.e. An upper bound on max-flow and hence coding capacity of network)
  - Minimization of storage bandwidth subject to this lower bound

Some background on single source multi-cast network coding

- Max-Flow-Min-Cut Theorem for DAGs
  - Condition for existance of flow rate r between single source node s and single sink node t
- Equivalent condition for existance of flow rate r via network coding(after some algebra)*
- Extension to single source s and multiple sinks $t \in \mathcal{T}$*

*Koetter, R.; Medard, M., "An algebraic approach to network coding," Networking, IEEE/ACM Transactions on , vol.11, no.5, pp.782,795, Oct. 2003

Some background on single source multi-cast network coding

- Max-Flow-Min-Cut Theorem for DAGs
  - Condition for existance of flow rate r between single source node s and single sink node t
- Equivalent condition for existance of flow rate r via network coding(after some algebra)*
- Extension to single source s and multiple sinks $t \in \mathcal{T}$*

*Koetter, R.; Medard, M., "An algebraic approach to network coding," Networking, IEEE/ACM Transactions on , vol.11, no.5, pp.782,795, Oct. 2003

Max-Flow-Min-Cut Theorem

- A DAG is a pair (V, E) where $e \in E$ are ordered 2-tuples of vertices in V and
  $\nexists\{v_1, \ldots, v_n\} \quad s.t. \quad (v_1, v_2), \ldots, (v_{n-1}, v_n), (v_n, v_1) \in E$

- V is vertex set and E is the edge set.

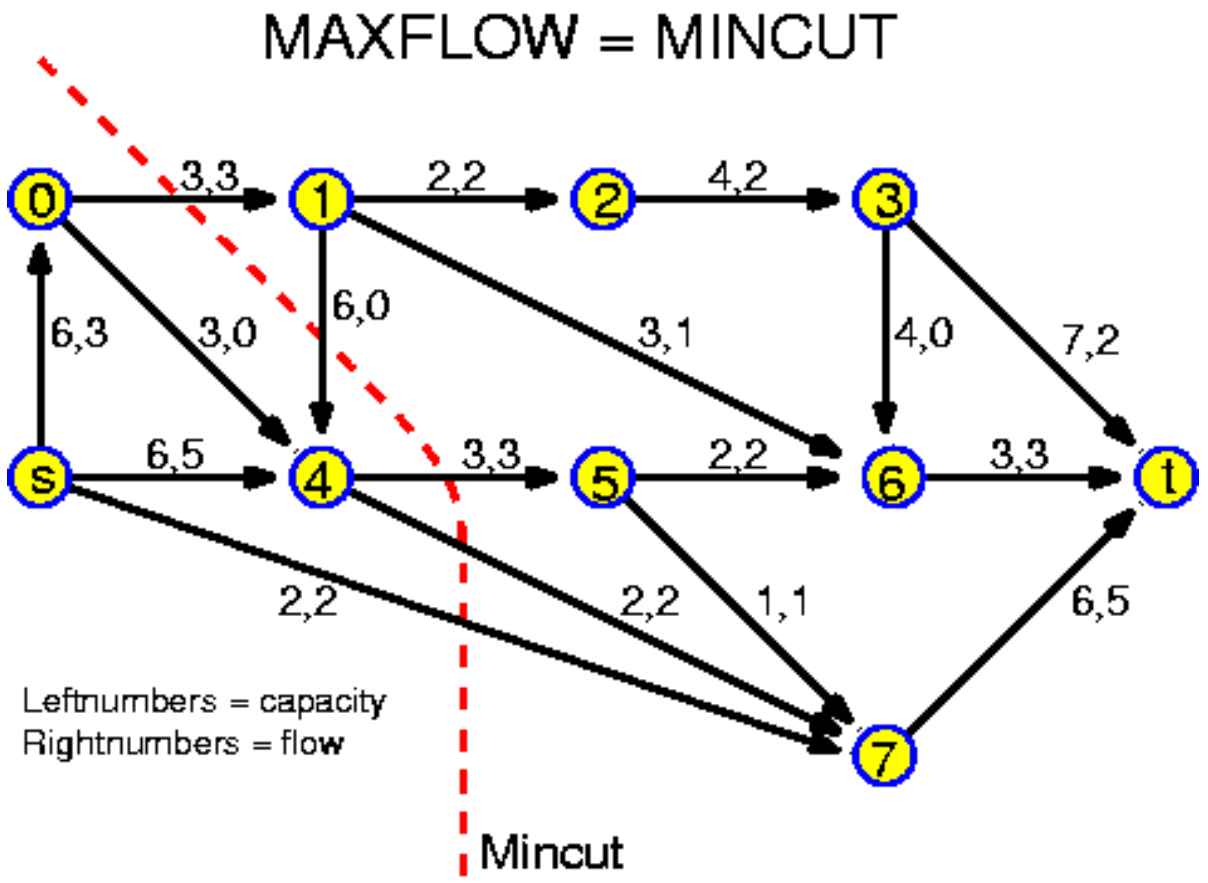
- Capacity:= $c : e \in E \mapsto R^{\geq 0}$

- (valid)Flow assignment:= $f : e \in E \mapsto R^{\geq 0}, f(e) \leq c(e)$
  $\forall e \in E$ and satisfies *flow conservation*

- Value of flow between $s, t \in V$ for a given valid flow assignment is

$$\sum_{\substack{v \in V \\ s.t.(v,t) \in E}} f((v, t))$$

Max-Flow-Min-Cut Theorem



Max-Flow-Min-Cut Theorem

- Independently proven in 1956 by Ford and Fulkerson AND by Elias, Feinstein and Claude Shannon
- **Theorem 1:** For an edge-capacitated DAG, following conditions are equivalent:
  - (C1) There exists a flow rate r between s and t
  - (C2) The value of the minimum cut between s and t is at least r
- Special case of LP duality(strong duality)

Some background on single source multi-cast network coding

- Max-Flow-Min-Cut Theorem for DAGs
  - Condition for existance of flow rate r between single source node s and single sink node t
- Equivalent condition for existance of flow rate r via network coding(after some algebra)*
- Extension to single source s and multiple sinks $t \in \mathcal{T}$*

*Koetter, R.; Medard, M., "An algebraic approach to network coding," Networking, IEEE/ACM Transactions on , vol.11, no.5, pp.782,795, Oct. 2003

Basic Network Model

- DAG $\mathcal{G} = (\mathcal{N}, \mathcal{A})$
- Arcs are lossless, delay-free communication links.
- for $l \in \mathcal{A}$, denote its origin node as $o(l)$ and destination node as $d(l)$
- $\mathcal{I}(v)$ and $\mathcal{O}(v)$ are sets of input arcs and output arcs respectively of node v
- r source random variables $X_1 \ldots X_r$ originating at $s \in \mathcal{N}$

Basic Network Model

- *Input random variables* of node v are $Y_l, l \in \mathcal{I}(v)$
- *Output random variables* are of node v are $Y_k, k \in \mathcal{O}(v)$
- Input random variables for s are the are the source random variables
- Output random variables for $t \in \mathcal{T}$ are $Z_{t,1}, \ldots, Z_{t,r}$ where $\mathcal{T} \subset \mathcal{N}$ is the set of sink nodes
- Each node forms its output random variables as a *scalar linear function* of its input random variables
- All the above random variables take binary values but can be viewed as elements of some finite field $\mathbb{F}_q, q = 2^m$ by grouping the bits

Local coding coefficients

- SLNC for arc l is given as:

$$Y_l = \begin{cases} \sum_i a_{i,l} X_i & if \quad o(l) = s \\ \sum_{k \in \mathcal{I}(o(l))} f_k Y_k & otherwise \end{cases}$$

- Sink nodes form their output symbols as: $Z_{t,i} = \sum_{k \in \mathcal{I}(t)} b_{t,i,k} Y_k$

- Stack coding coefficients as $(a_{i,l} : 1 \leq i \leq r, l \in \mathcal{A})$ and $(f_{k,l} : l, k \in \mathcal{A})$ into vectors $\boldsymbol{a}$ and $\boldsymbol{f}$ respectively and decoding coefficients $(b_{t,i,k} : t \in \mathcal{T}, 1 \leq i \leq r, k \in \mathcal{A})$ in vector $\boldsymbol{b}$

Global coding coefficients

- Since all the nodes are forming only scalar linear combinations of input variables we can represent arc variables as linear combination of source variables: $Y_l = \sum_{i=1}^{r} c_{i,l} X_i$
- $c_{i,l} \in \mathbb{F}_q$ are functions of $(\boldsymbol{a}, \boldsymbol{f})$
- $\boldsymbol{c}_l = [c_{1,l} \dots c_{r,l}] \in \mathbb{F}_q^r$ is called *global coding vector* of arc l
- Output variables $Z_{i,t}$ can be similarly expressed as linear combination of source variables
- Stack output variables of $t \in \mathcal{T}$ into vector $\boldsymbol{z}_t = [Z_{t,1} \dots Z_{t,r}]$

Matrix formulation

- $\boldsymbol{A} = (a_{i,l})$ is $r \times |\mathcal{A}|$ matrix where $a_{i,l}$ are coefficients with which which source RVs are combined on source's output arcs
- $\boldsymbol{F} = (f_{k,l})$ is $|\mathcal{A}| \times |\mathcal{A}|$ matrix whose non-zero entries $f_{k,l}$ are coefficients used by $d(k)$ for combining symbols Y_k to form symbol Y_l
- $\boldsymbol{B}_t = (b_{t,i,k})$ is an $r \times |\mathcal{A}|$ matrix whose nonzero entries $b_{t,i,k}$ are the coefficients with which sink t combines symbols Y_k on its input arcs to form output symbols $Z_{t,i}$

The transfer matrix

Theorem 2(Koetter and Médard): Let a network be given with matrices $\boldsymbol{A}$,$\boldsymbol{F}$,$\boldsymbol{B}$. The transfer matrix of the network for sink t is given as:

$$\boldsymbol{M}_t = \boldsymbol{A}(\boldsymbol{I} - \boldsymbol{F})^{-1}\boldsymbol{B}_t^T$$

where $\boldsymbol{I}$ is the identity matrix

Proof of Theorem 2

- The transfer matrix between output variables of s and input variables of t is given by the series $\boldsymbol{I} + \boldsymbol{F} + \boldsymbol{F}^2 + \ldots$
- Since network is acyclic, $\boldsymbol{F}$ is nilpotent i.e. $\exists \hat{n} < \infty$ s.t. $\boldsymbol{F}^{\hat{n}} = \boldsymbol{0}$
- $\boldsymbol{I} + \boldsymbol{F} + \boldsymbol{F}^2 + \ldots = (\boldsymbol{I} - \boldsymbol{F})^{-1}$
- $\therefore \boldsymbol{M}_t = \boldsymbol{A}(\boldsymbol{I} - \boldsymbol{F})^{-1}\boldsymbol{B}_t^T$

Theorem 3(Koetter and Médard): When $|\mathcal{T}| = 1$, the condition

- (C3) The determinant of the transfer matrix is non-zero over the ring of polynomials $\mathbb{F}_2[\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{b}]$

is equivalent to (C1) and (C2) of theorem 1

Lemma 1: Let f be a non-zero polynomial in variables $x_1, x_2, \ldots, x_n$ over $\mathbb{F}_2$ and let d be the maximum degree of f w.r.t. any variable. Then $\exists$ values for $x_1, x_2, \ldots, _n$ in $\mathbb{F}_{2^m}^n$ s.t. $f(x_1, x_2, \ldots, x_n) \neq 0$ for any m s.t. $2^m > d$

Proof:

- Let f be polynomial in $x_2, \ldots, x_n$ with coefficients from $\mathbb{F}_2[x_1]$ i.e. coefficients are polynomials in x_1
- These coefficients have degree atmost d and hence $x_1^{2^m} - x_1$ doesn't divide them. (If it did, we could write the coefficients in form $q(x_1)(x_1^{2^m} - x_1)$ and any x_1 chosen from $\mathbb{F}_{2^m}$ would render $f = 0$)
- $\therefore \exists \alpha \in \mathbb{F}_{2^m}$ s.t. $f \neq 0$ when $x_1 = \alpha$
- Now we can choose values for $x_2, \ldots, x_n$ by induction on variables

Proof of Theorem 3

- (C1)$\Rightarrow$ (C3)
  - C1 holds
    $\Rightarrow \exists r$ arc disjoint paths from s to t
    $\Rightarrow \boldsymbol{M}_t = \boldsymbol{I}$
    $\Rightarrow C3$
- (C3)$\Rightarrow$ (C1)
  - C3 holds
    $\Rightarrow \exists$ values of $\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{b}$ in some large finite field
    s.t. $det(\boldsymbol{M}_t) \neq 0$
    $\Rightarrow \tilde{\boldsymbol{B}}_t = (\boldsymbol{M}_t^T)^{-1}\boldsymbol{B}_t$ satisfies $\boldsymbol{A}(\boldsymbol{I}-\boldsymbol{F})^{-1}\tilde{\boldsymbol{B}}_t^T = \boldsymbol{I}$
    $\Rightarrow (\boldsymbol{A}, \boldsymbol{F}, (\tilde{B}))$ is solution to network coding problem that has $\boldsymbol{M}_t = \boldsymbol{I}$
    $\Rightarrow \exists r$ arc disjoint paths between s and t
    $\Rightarrow C1$

Some background on single source multi-cast network coding

- Max-Flow-Min-Cut Theorem for DAGs
  - Condition for existance of flow rate r between single source node s and single sink node t

  Equivalent condition for existance of flow rate r via network coding(after some algebra)*
- Extension to single source s and multiple sinks $t \in \mathcal{T}$*

*Koetter, R.; Medard, M., "An algebraic approach to network coding," Networking, IEEE/ACM Transactions on , vol.11, no.5, pp.782,795, Oct. 2003

Extension to multicast

Theorem 4(Fundamental theorem of multicast network coding): Consider an acyclic delay-free multicast problem where r source processes originating at a source node s are demanded by a set $\mathcal{T}$ of sinks. Then a network coding solution exists iff $\exists$ for each $t \in \mathcal{T}$ a flow of rate r between s and t

Proof:

$\forall$t$\in \mathcal{T}$ $\exists$ flow rate r between s and t

$\Leftrightarrow \forall t \in \mathcal{T}, \quad det(\boldsymbol{M}_t) \neq 0$ over ring of polynomials $\mathbb{F}_2[\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{b}]$

$\Leftrightarrow \prod_{t \in \mathcal{T}} det(\boldsymbol{M}_t) \neq 0$ over ring of polynomials $\mathbb{F}_2[\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{b}]$

$\Leftrightarrow \exists(\boldsymbol{a}, \boldsymbol{f}, \boldsymbol{b})$ in some large finite field s.t. $\prod_{t \in \mathcal{T}} det(\boldsymbol{M}_t) \neq 0$

Each sink then multiplies its output vector $\boldsymbol{z}_t$ by $(\boldsymbol{M}_t)^{-1}$

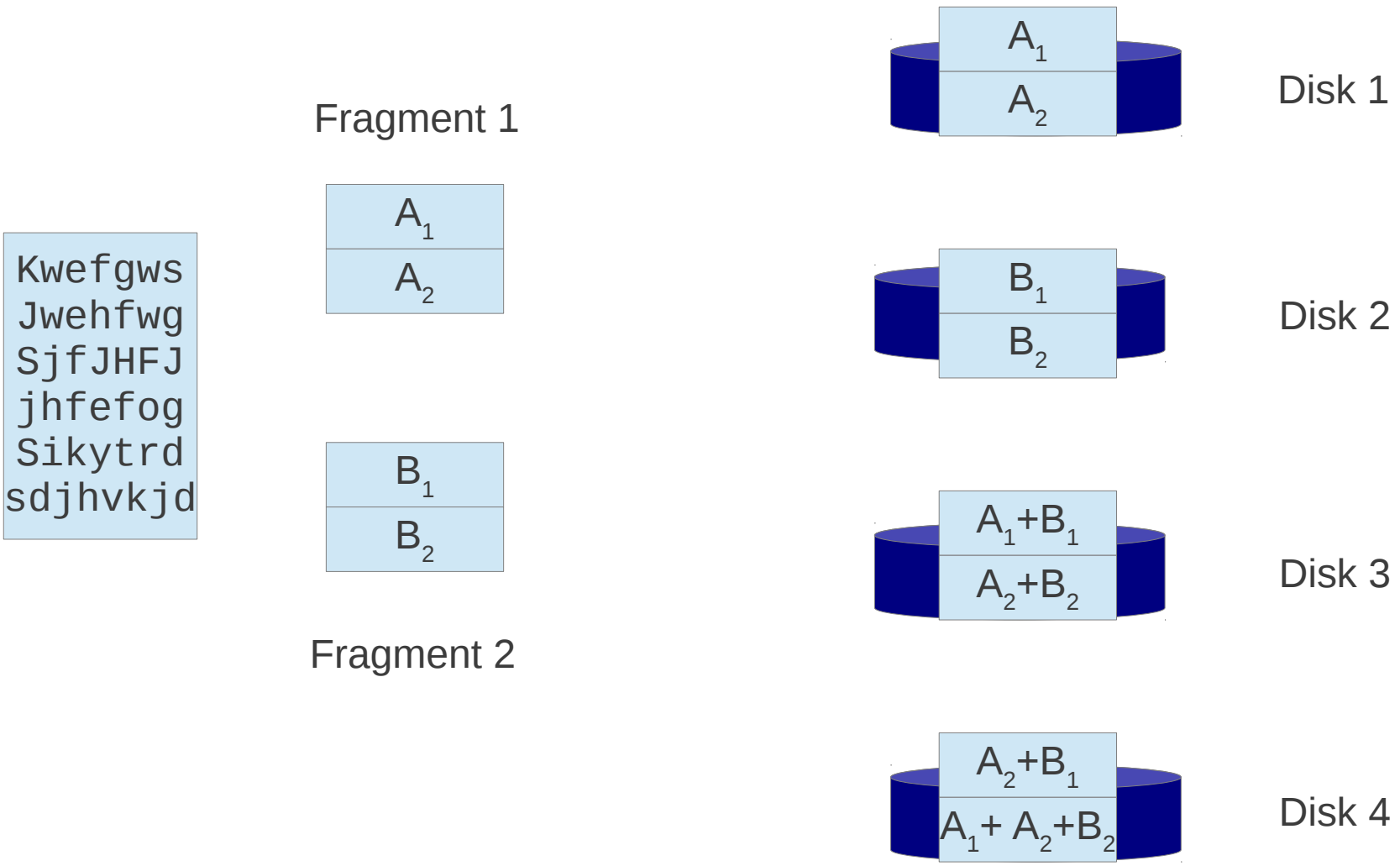
Part 2- Outline

- Introduction
- The repair problem
- Reduction of repair problem to single source multicast network
- Family of single source multi-cast networks arising from the reduction
- A lower bound on min-cuts(i.e. An upper bound on max-flow and hence coding capacity of network)
- Minimization of storage bandwidth subject to this lower bound

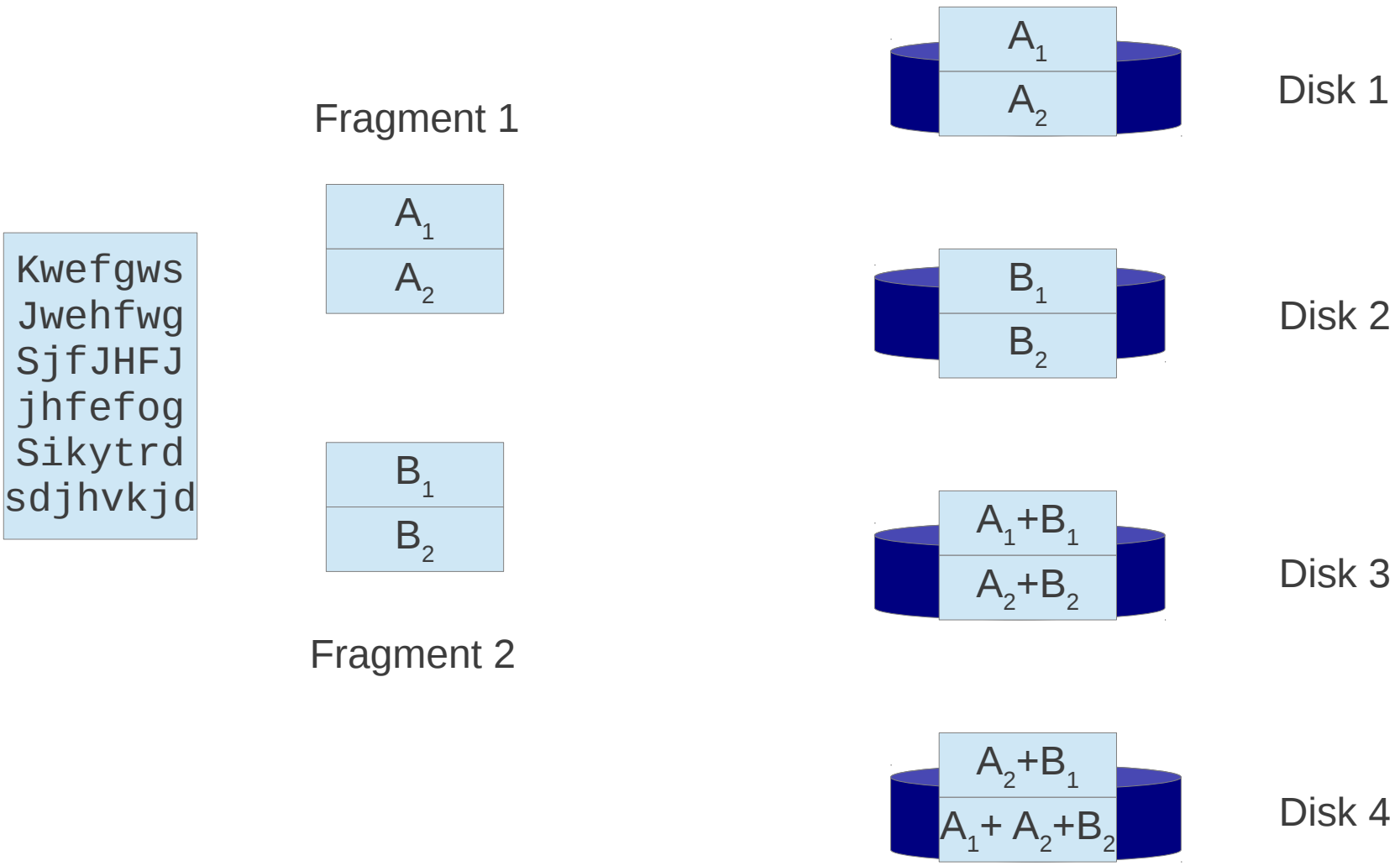
Distributed storage

- We are living in an internet age
- Demand for large scale data storage has increased significantly
- Social networks, file and video sharing require seamless storage, access and security for massive amounts of data
- Storage mediums(viz. hard-drives) are individually unreliable
- Hence we introduce redundancy via the use of erasure codes to improve reliability

A storage code((4,2) MDS)



A storage code((4,2) MDS)



Part 2- Outline

- Introduction
- The repair problem
- Reduction of repair problem to single source multicast network
- Family of single source multi-cast networks arising from the reduction
- A lower bound on min-cuts(i.e. An upper bound on max-flow and hence coding capacity of network)
- Minimization of storage bandwidth subject to this lower bound

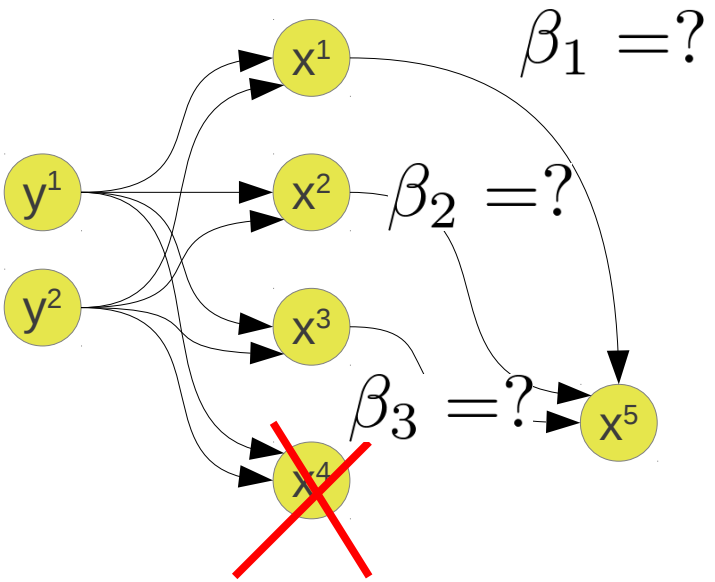
Problem Definition

- Storage nodes are *distributed* and *connected in a network*
- Together they represent some storage code(MDS or approximate MDS like LDPC)
- The issue of repairing a node arises when a storage node of the system fails
- The still functioning nodes are called *active nodes*
- A newcomer node called *repair node* must *connect to a subset of active* nodes, obtain information from them and r*econstruct the storage code i.e, repair the code*
- The objective is to minimize amount of information transferred in this process

Notation

- Storage node: A node/disk that stores data
- Repair node: A node that replaces failed storage node
- $\mathcal{M}$=Size of file we want to store
- k = fragments into which file is divided
- $\mathcal{M}/k$ = Size of each file fragment
- (n,k): Size of MDS code used
- d = size of subset of active nodes that a newcomer is allowed to connect to
- α = Storage capacity of each node, also called *storage bandwidth*
- $\beta_i, i \in A$ = Amount of information conveyed by i^{th} active node to newcomer node ,
  i.e repair bandwidth for that active node
- $\gamma = \sum_i \beta_i$ = total *repair bandwidth*

The repair problem



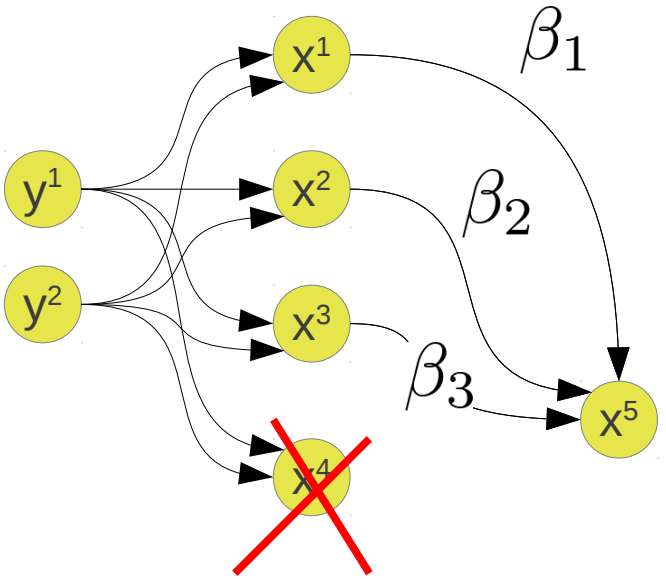
Example: A (4,2) MDS code
(β = repair bandwidth per node)

The repair problem

- Data object (2Mb) is divided into two fragments: y^1,y^2 (1 Mb each)
- 4 encoded fragments generated: x^1,x^2,x^3,x^4 (1 Mb each)
- x^4 fails, x^5, the newcomer needs to communicate with existing nodes and create a new encoded packet
- Any two out of x^1,x^2,x^3,x^5 must suffice to recover original data object

The repair problem

- What(and how much) should x^1,x^2,x^3 communicate to x^5 such that $\beta_i s$ are minimized?



Example 1: A (4,2) MDS code

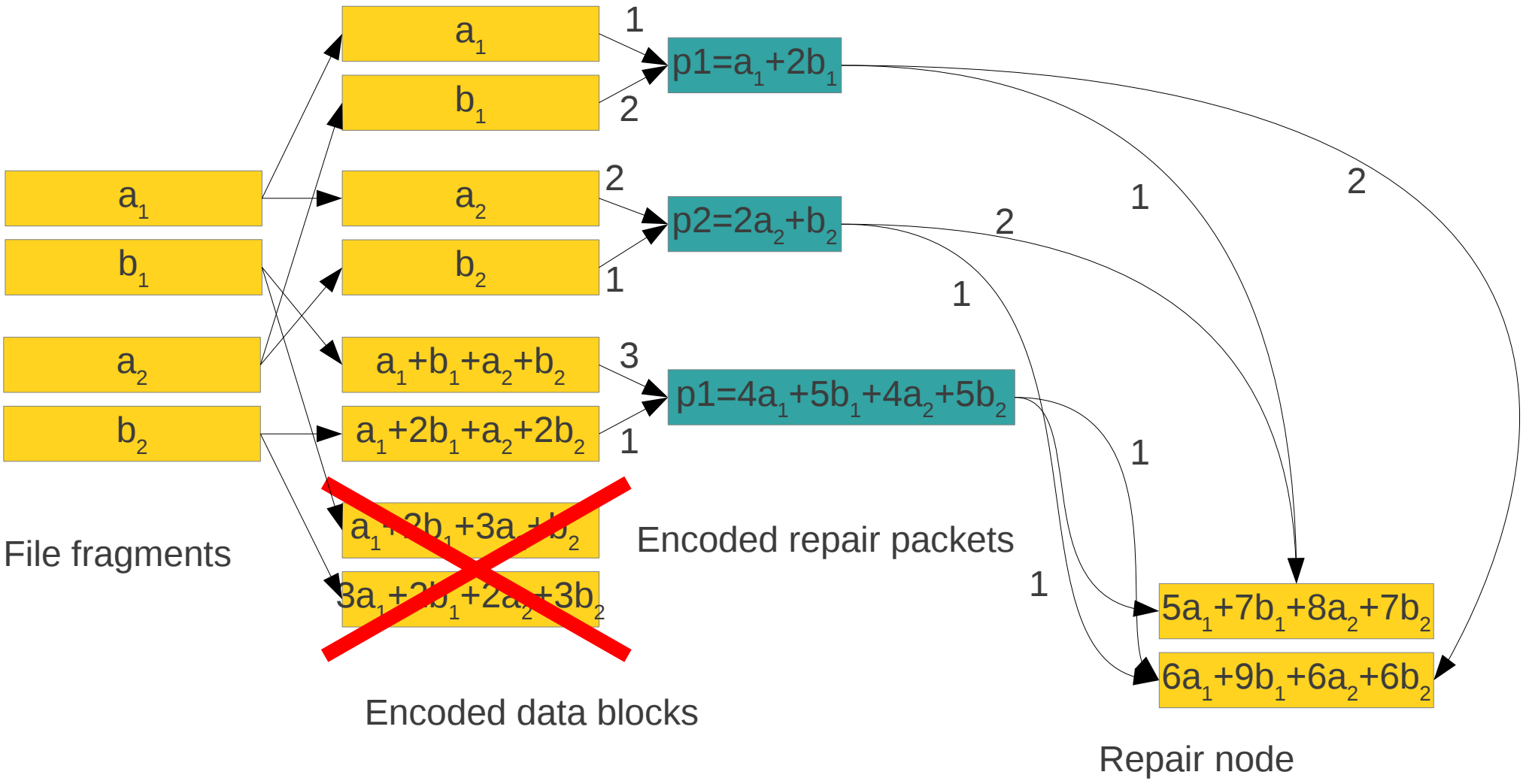
Variants of the repair problem

- Exact Repair: Failed blocks are exactly regenerated i.e. newcomer node must reconstruct exact replica of encoded block in the failed node
- Functional Repair: Newly generated data block need not be exact replica of encoded block on the failed node
- Exact repair of the systematic part: Only repair the systematic part exactly so there is always a un-coded copy of original file available

Variants of the repair problem

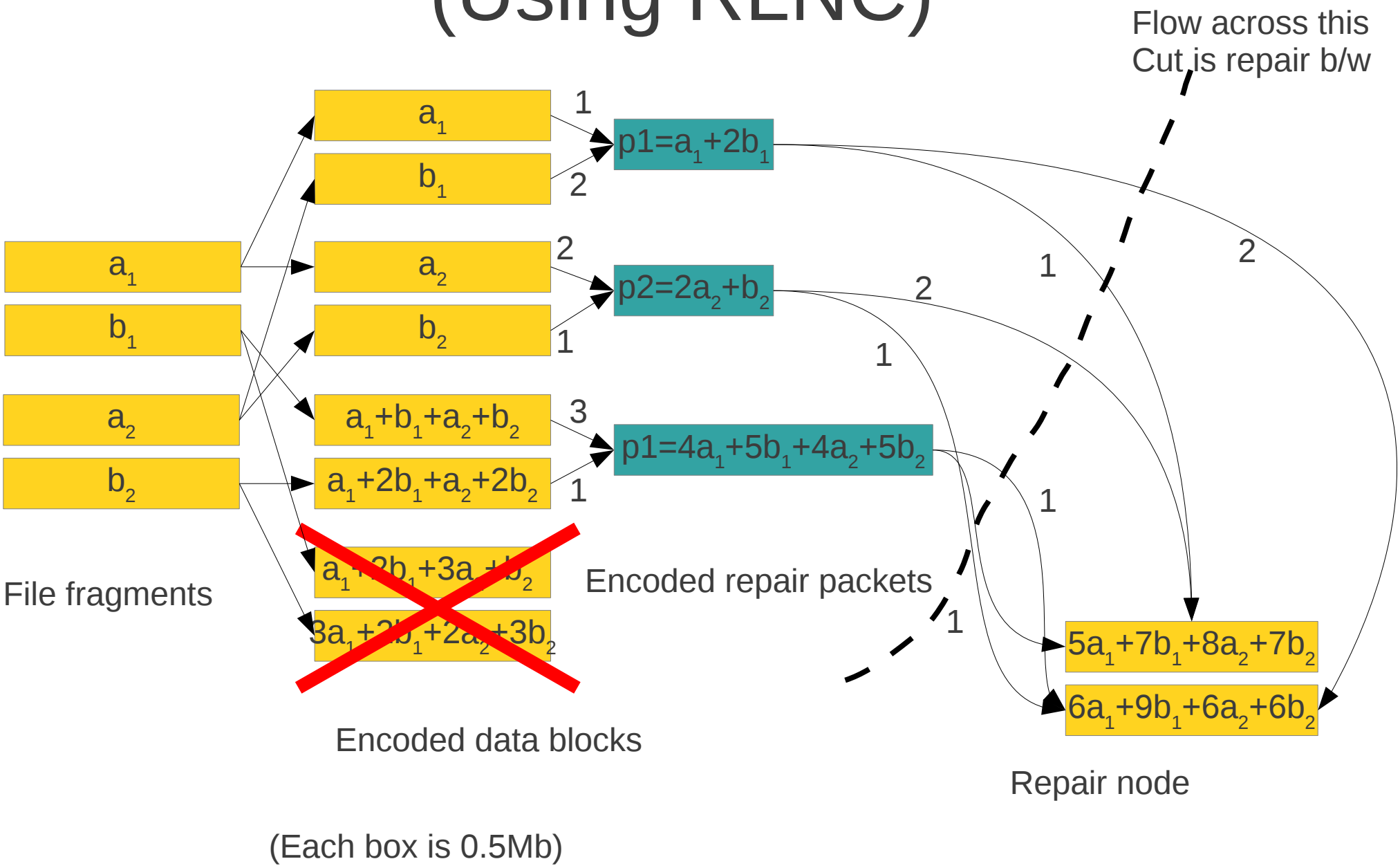
- Exact Repair: Failed blocks are exactly regenerated i.e. newcomer node must reconstruct exact replica of encoded block in the failed node
- Functional Repair: Newly generated data block need not be exact replica of encoded block on the failed node
- Exact repair of the systematic part: Only repair the systematic part exactly so there is always a un-coded copy of original file available

Functional repair example (Using RLNC)

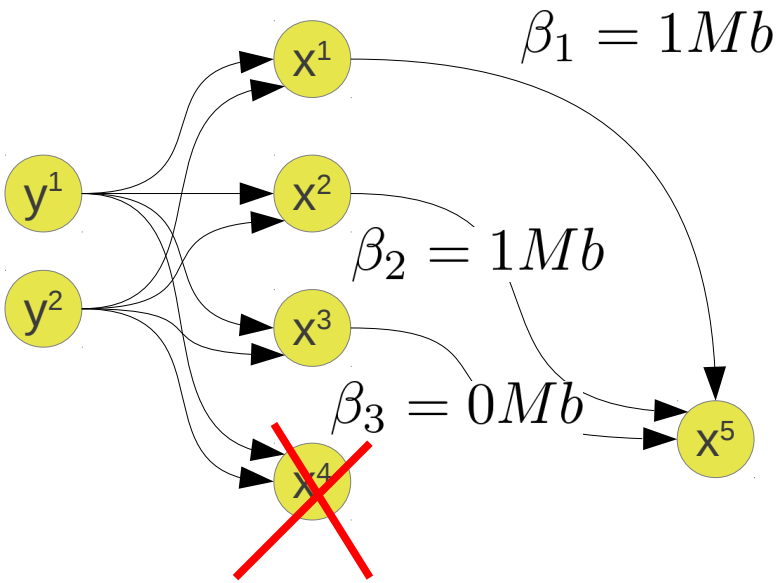


(Each box is 0.5Mb)

Functional repair example (Using RLNC)

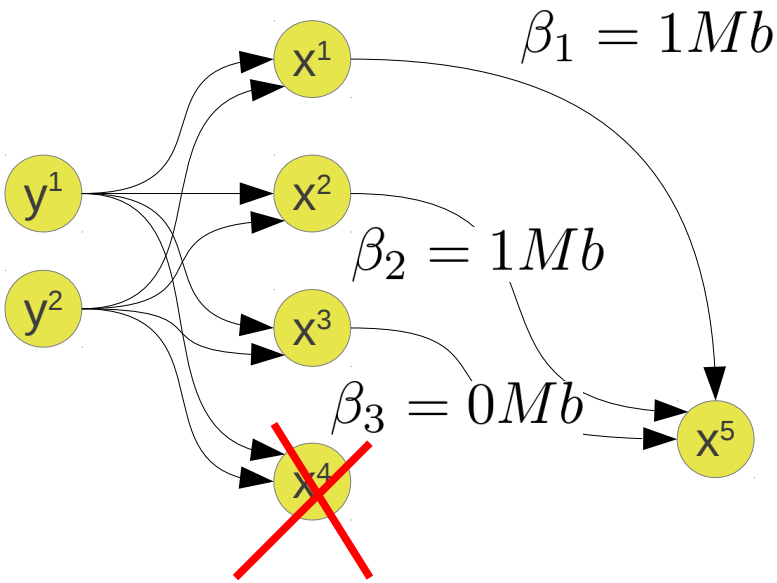


An attempt at solution



Example 1: A (4,2) MDS code

An attempt at solution



x5 Recovers original data object and creates a new independent linear combination

Example 1: A (4,2) MDS code

Can we do better than this?

Can we do better than this?

YES!

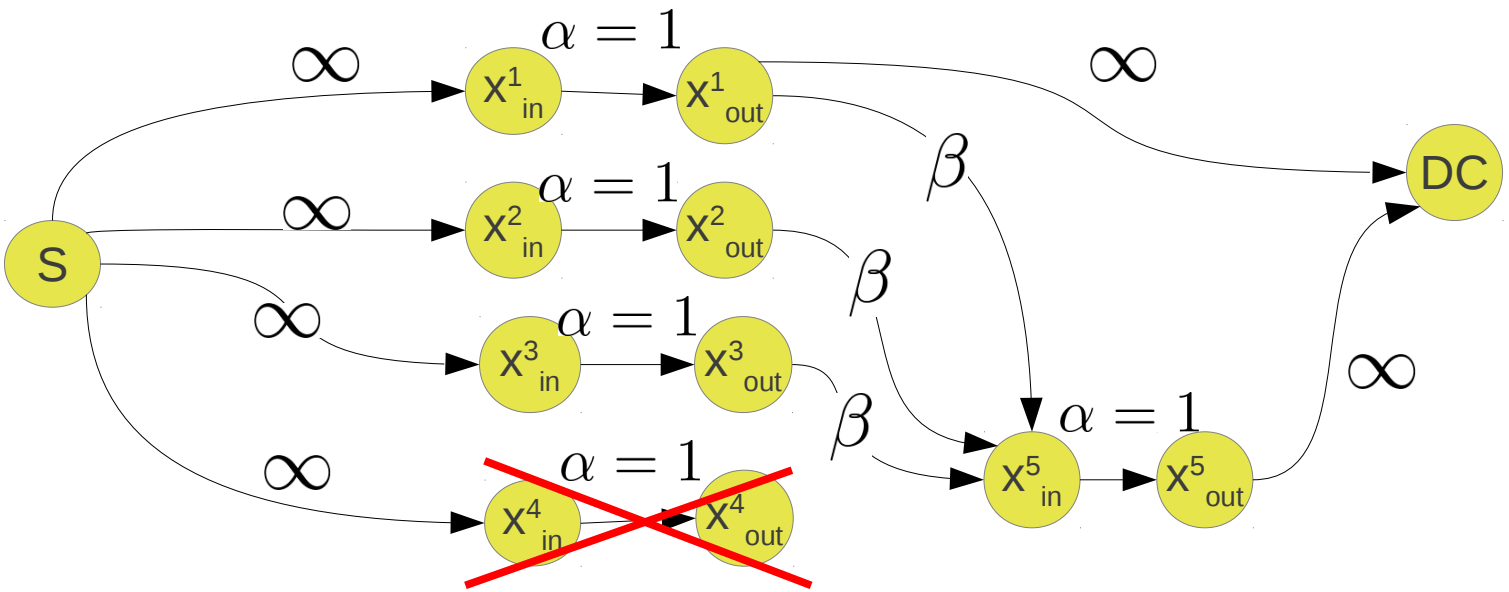
Part 2- Outline

- Introduction
- The repair problem
- Reduction of repair problem to single source multicast network
- Family of single source multi-cast networks arising from the reduction
- A lower bound on min-cuts(i.e. An upper bound on max-flow and hence coding capacity of network)
- Minimization of storage bandwidth subject to this lower bound

Reduction to information flow graph

- Replace each storage and repair node x_i by 2 nodes: x_{in}^i and x_{out}^i with a directed edge from x_{in}^i to x_{out}^i of capacity α
- Add a source node S with outgoing edges to each of the storage nodes storage node and having capacity ∞
- Assume all nodes transfer same amount of information to the repair node when connected to
- Add a data collector node DC and that connects to any k-subset of active nodes via directed edges of infinite capacity

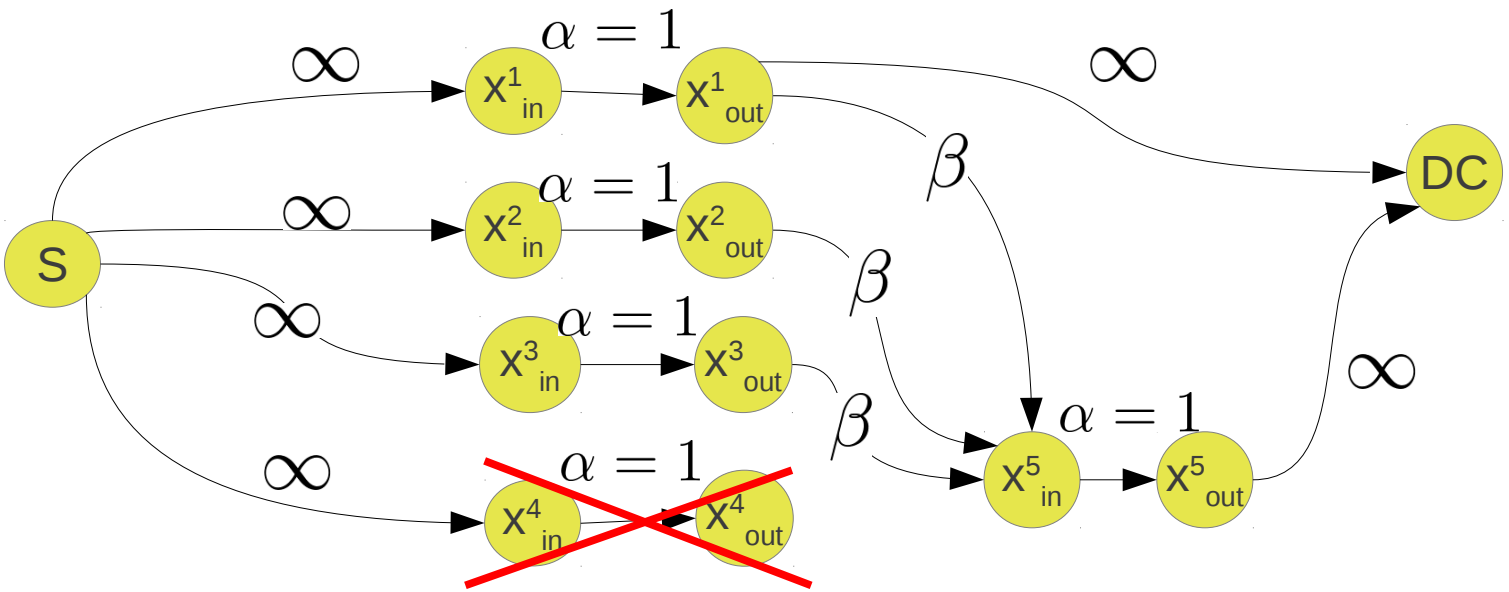
Example



Information flow graph corresponding to Example 1: A (4,2) MDS code

Node 4 has failed

Dynamic nature of information flow graph due to given failure pattern

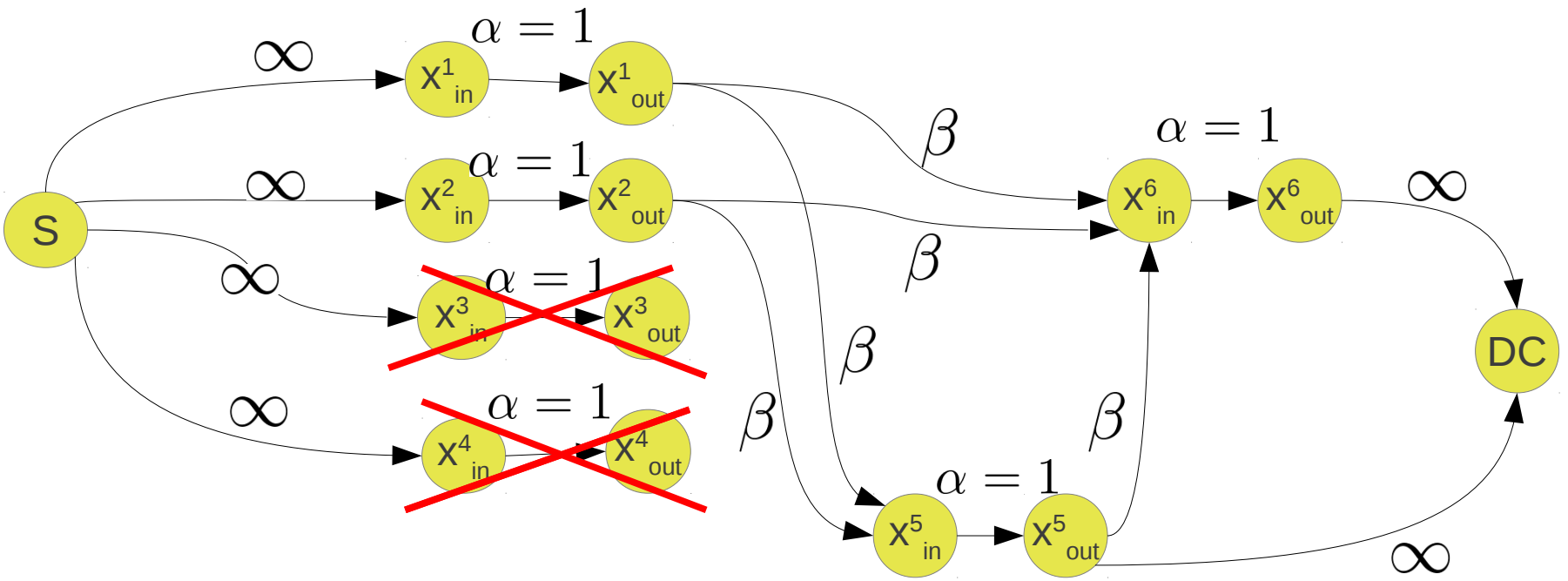


Information flow graph corresponding to Example 1: A (4,2) MDS code

Node 4 has failed

Family of information flow graphs

$$\mathcal{G}(n, k, d, \alpha, \beta)$$



Information flow graph corresponding
to Example 1: A (4,2) MDS code
Node 3 also failed say a few minutes later

Lemma 1

Lemma 1: No data collector can reconstruct the initial data object if the minimum cut in $\mathcal{G}$ between S and DC is smaller than the initial object size $\mathcal{M}$

Proposition 1: Given a finite information flow graph and a finite number of data collectors, if the minimum of the min-cuts seperating the source with each data collector is larger or equal to the data object size $\mathcal{M}$, then there exists a linear network code defined over a sufciently large finite field (whose size depends on the graph size) such that all data collectors can recover the data object. Further, randomized network coding guarantees that all collectors can recover the data object with probability that can be driven arbitrarily high by increasing the field size.

Outline

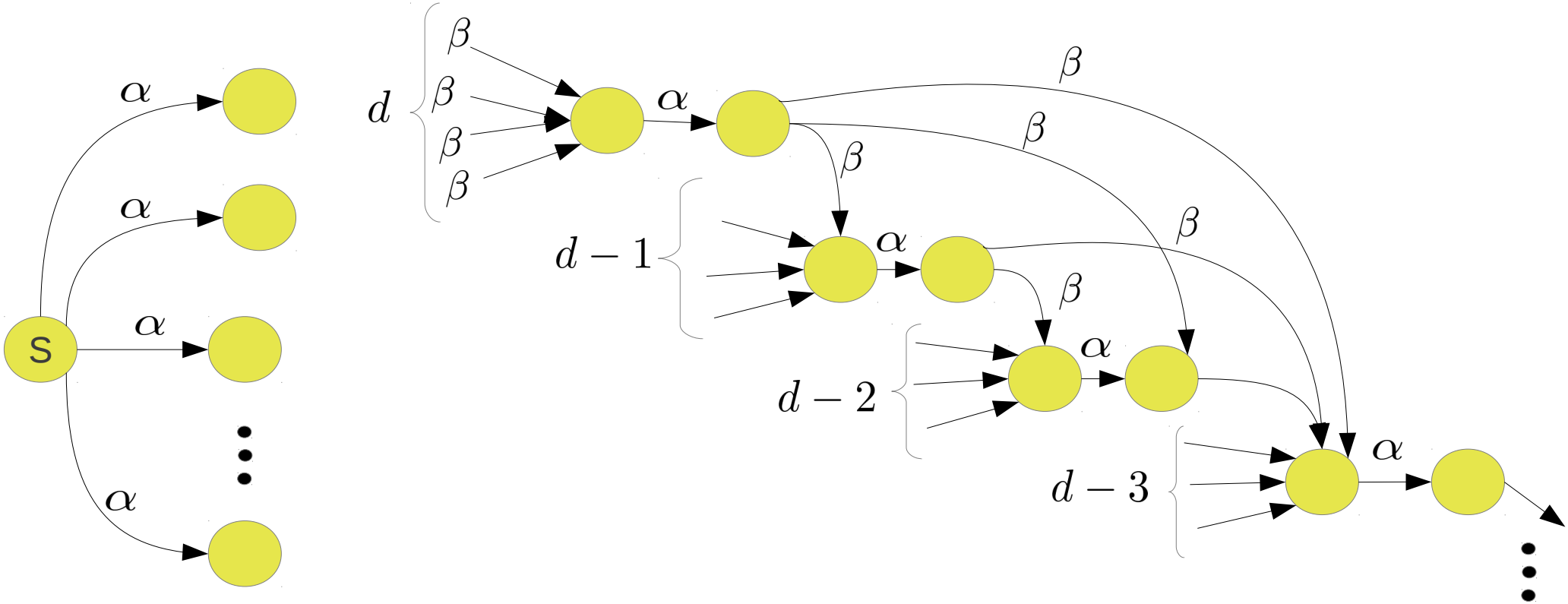
- The repair problem
- Reduction of repair problem to single source multicast network
- Family of single source multi-cast networks arising from the reduction
- A lower bound on min-cuts(i.e. An upper bound on max-flow and hence coding capacity of network)
- Minimization of storage bandwidth subject to this lower bound

Lemma 2: Consider any (potentially infinite) information flow graph G, formed by having n initial nodesthat connect directly to the source and obtain α bits, while additional nodes join the graph by connecting to d existing nodes and obtaining β bits from each. Any data collector t that connects to a k-subset of "out-nodes" of G must satisfy

$$mincut(s,t) \geq \sum_{i=0}^{min\{d,k\}-1} min\{min\{(d-i)\beta, \alpha\}\}$$

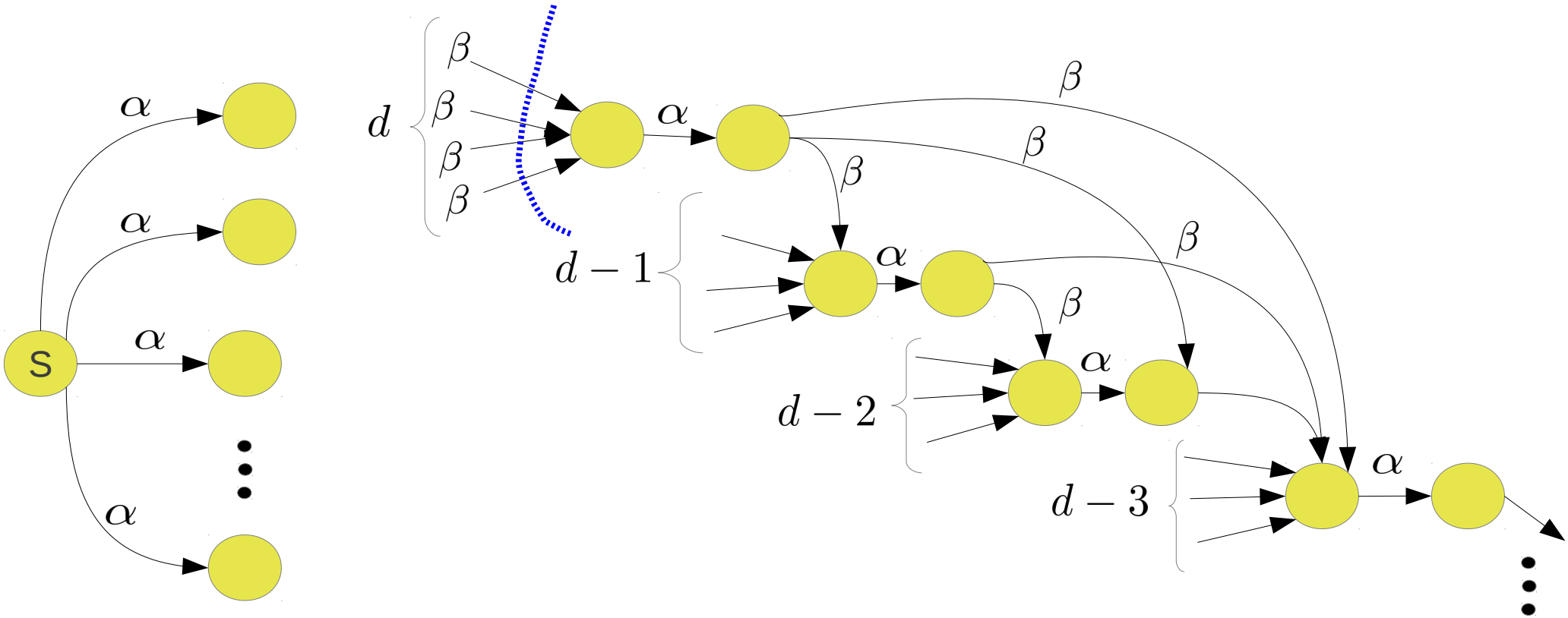
Furthermore, there exists an information flow graph $G^* \in \mathcal{G}(n,k,d,\alpha,\beta)$ where this bound is matched with equality

Information flow graph G^*



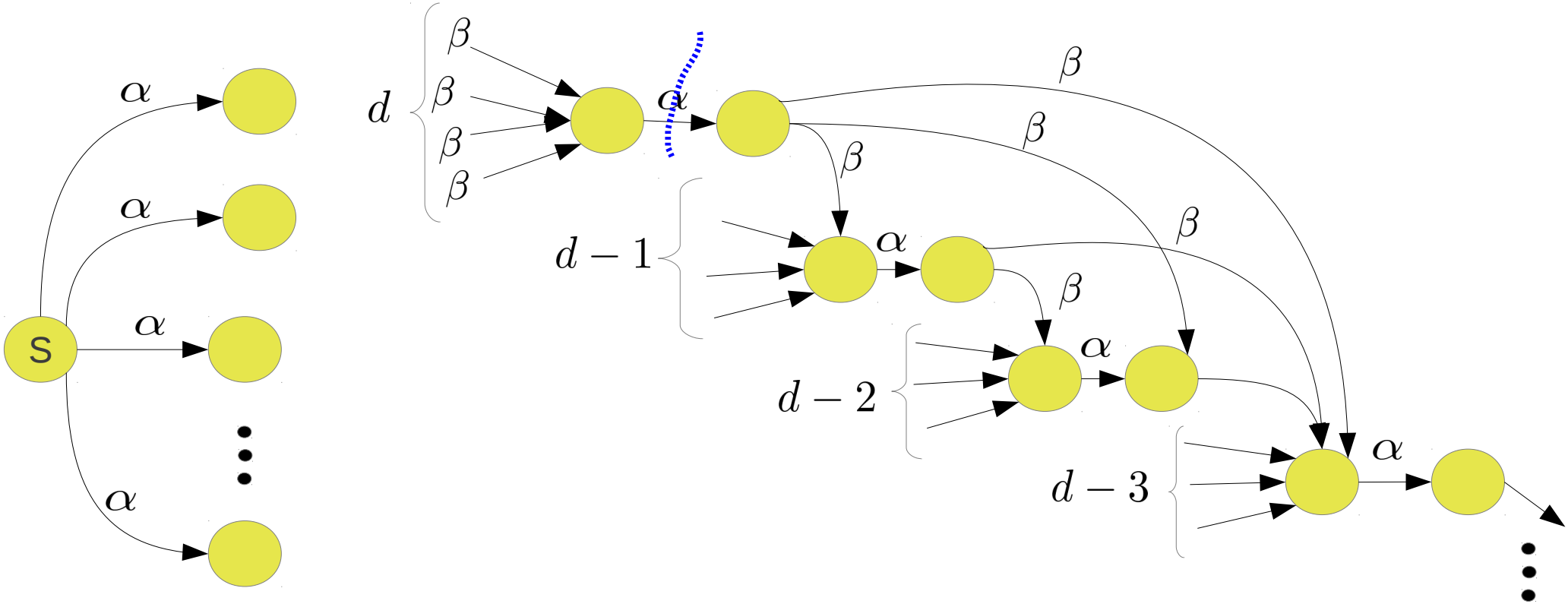
Claim: There exists a way of connecting DC to G^* s.t. min-cut achieves equality in the bound in Lemma 2

Information flow graph G^*



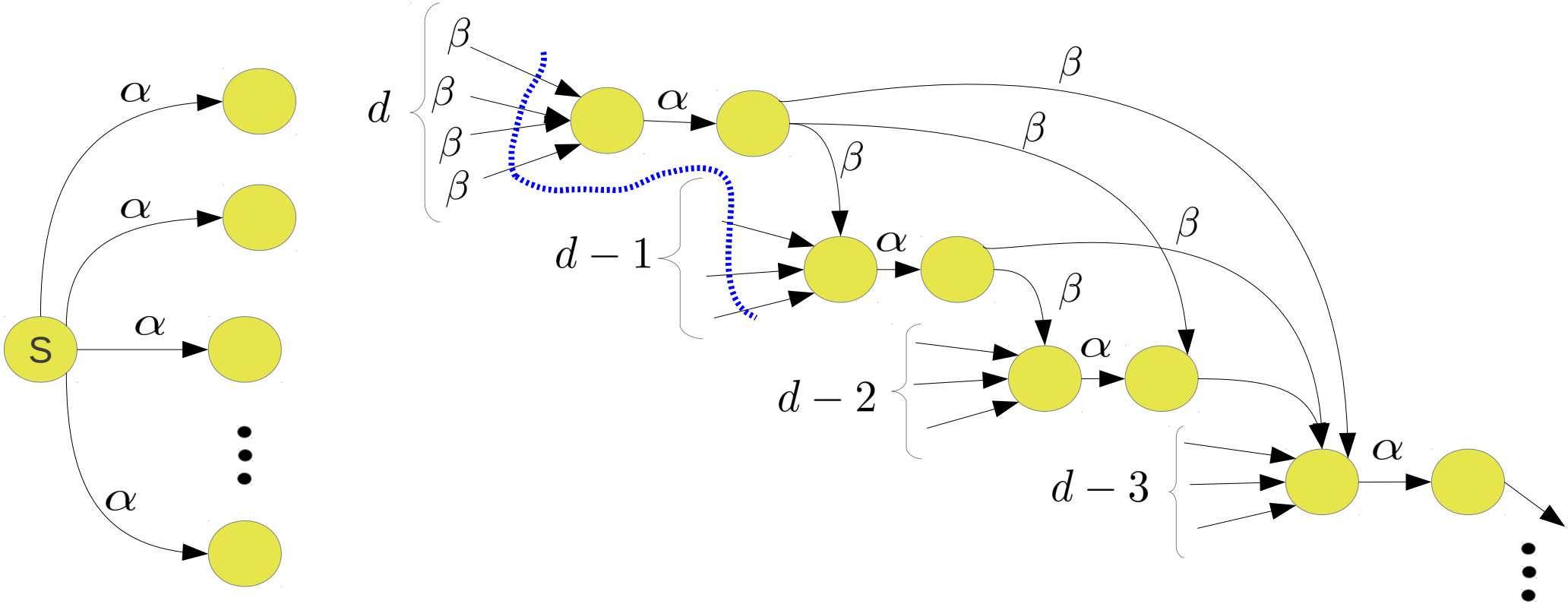
Claim: There exists a way of connecting DC to G^* s.t. min-cut achieves equality in the bound in Lemma 2

Information flow graph G^*



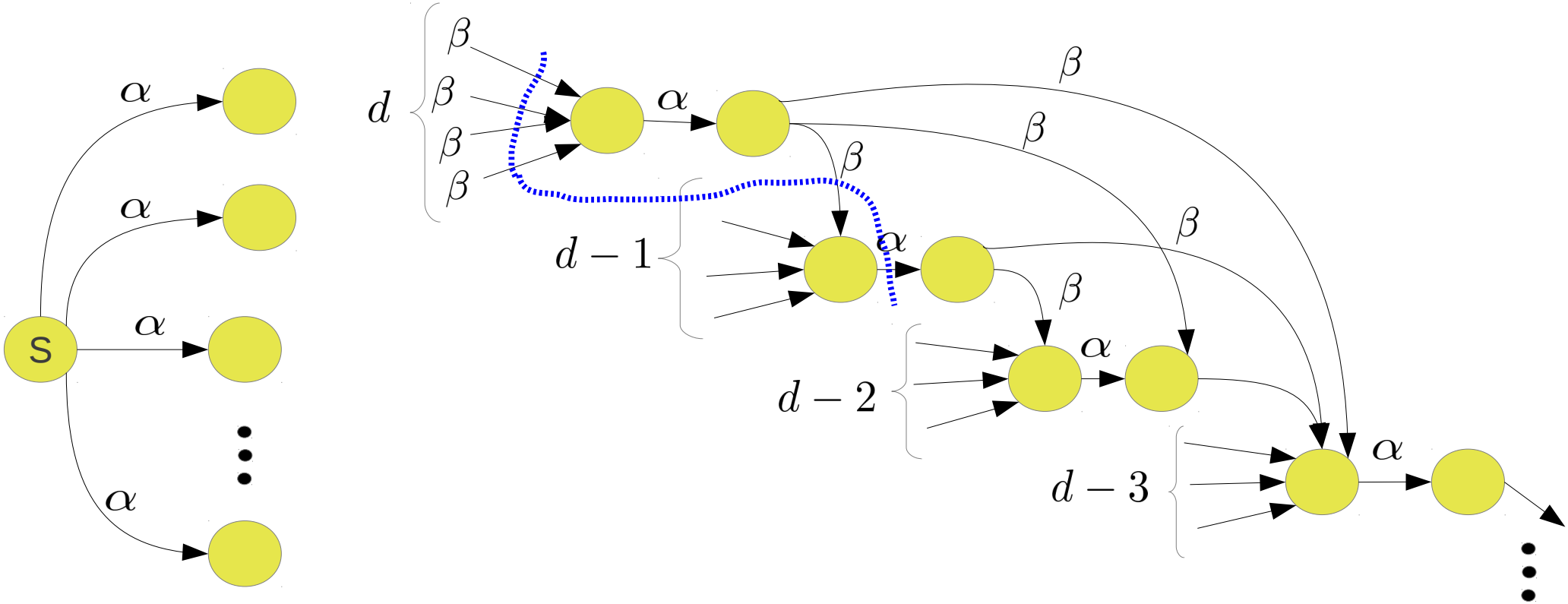
Claim: There exists a way of connecting DC to G^* s.t. min-cut achieves equality in the bound in Lemma 2

Information flow graph G^*



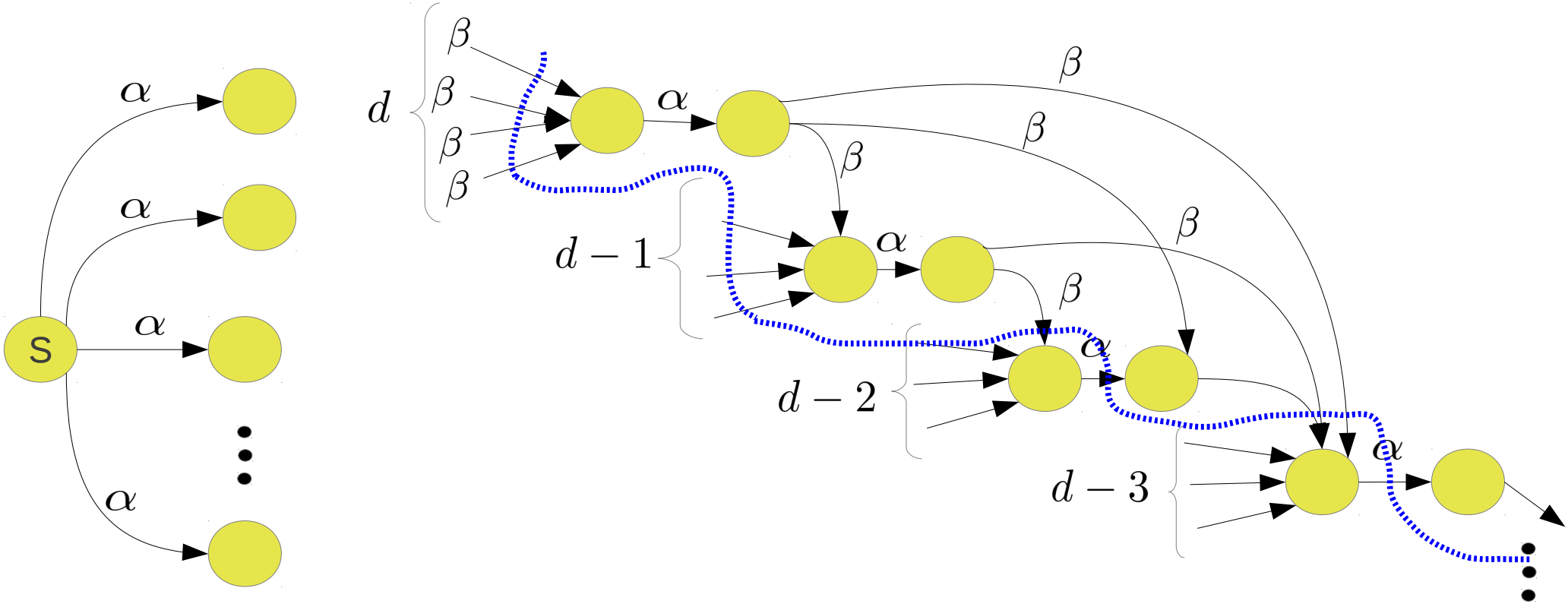
Claim: There exists a way of connecting DC to G^* s.t. min-cut achieves equality in the bound in Lemma 2

Information flow graph G^*



Claim: There exists a way of connecting DC to G^* s.t. min-cut achieves equality in the bound in Lemma 2

Information flow graph G^*



Claim: There exists a way of connecting DC to G^* s.t. min-cut achieves equality in the bound in Lemma 2

Proof

- The are initially n nodes ;labeled 1 to n
- k newcomers are labeled as $n+1, \ldots, n+k$
- Newcomer $n+i$ connects to nodes $n+1, \ldots, n+i-1$
- A DC t connects to $n+1, \ldots, n+k$
- Consider a cut $(U, \bar{U})$ formed as follows:
  - For each newcomer $n+i$ to $n+k$, if $\alpha \leq (d-i+1)\beta$, include only x_{out}^{n+i} in $\bar{U}$
  - otherwise include x_{out}^{n+i} and x_{in}^{n+i} both in $\bar{U}$

The capacity of resultant cut is $\sum_{i=0}^{min(d,k)-1} min((d-i)\beta, \alpha)$

Claim: The bound holds for any graph G in which newcomers connect to $k - subset$ of x_{out} nodes in arbitrary pattern.

Proof:

- It suffices to prove the result for cuts $(U, \bar{U})$ s.t. $s \in U$ and $x_{out} \in \bar{U}$
- We consider x_{out} nodes in $\bar{U}$ in topologically sorted order
- for $x^i_{out}, i = 0, \ldots, min\{d, k\} - 1$ we add either an edge of capacity α or $d - i$ edges of capacity β
- The bound in Lemma 2 sums up these contributions

Capacity of (n, k, d, α, β) regenerating codes

$$C \triangleq \sum_{i=0}^{min\{d,k\}-1} min\{(d-i)\beta, \alpha\}$$

$d \geq k$ WLOG

- If $d < k$, we set $k = d$
- i.e. we require d storage nodes to have a flow of $\mathcal{M}$
- This leads to same lower bound on mincuts

Outline

- The repair problem
- Reduction of repair problem to single source multicast network
- Family of single source multi-cast networks arising from the reduction
- A lower bound on min-cuts(i.e. An upper bound on max-flow and hence coding capacity of network)
- Minimization of storage bandwidth subject to this lower bound

Minimize α subject to the lower bound

- We want to characterize the achievable tradeoffs between storage α and the repair bandwidth $d\beta$
- We fix d, γ to some integer value and keep n, k as variables
- We can write the optimization as:

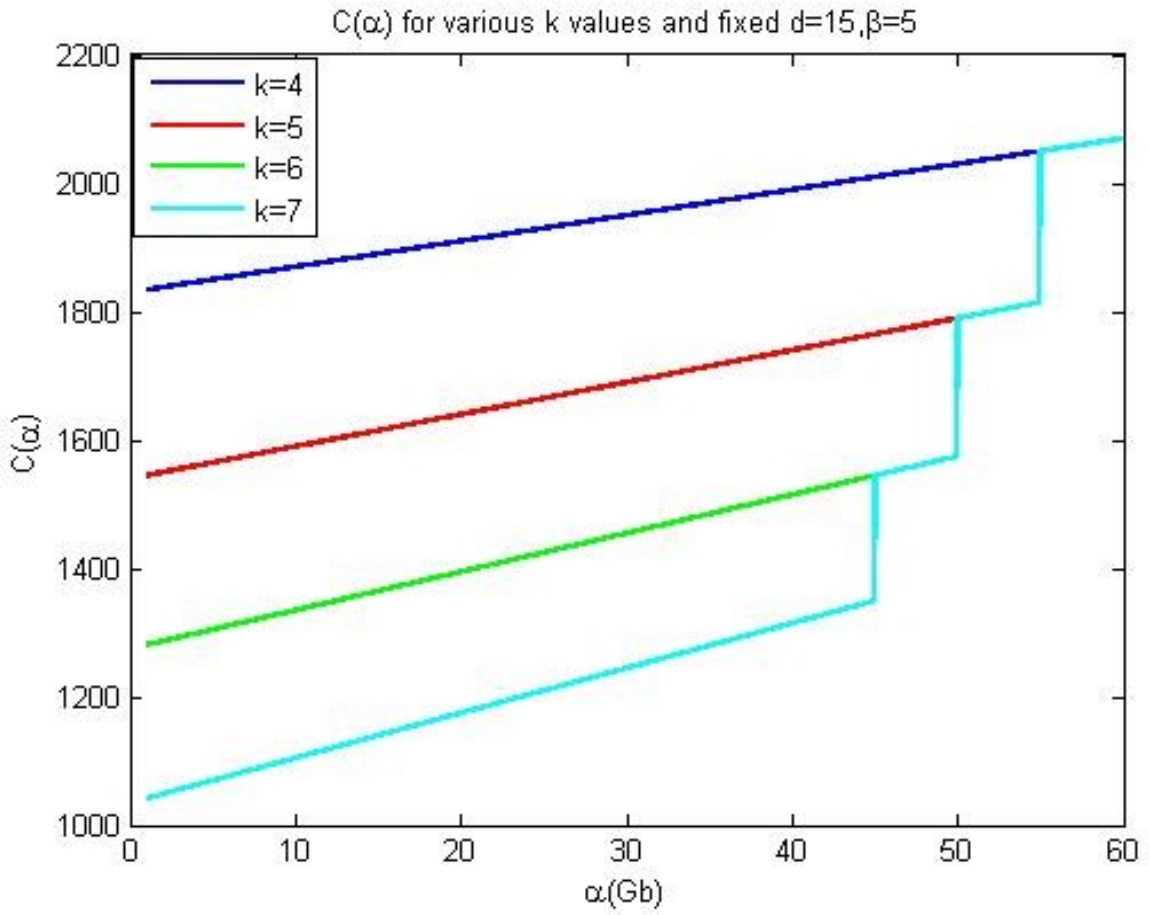
$$\alpha^*(d,\gamma) \triangleq \quad min \quad \alpha$$
$$s.t. \sum_{i=0}^{k-1} min\left\{\left(1-\tfrac{i}{d}\right)\gamma, \alpha\right\} \geq \mathcal{M}$$

Nature of constraint

- Let $b_i = \left(1 - \frac{k-1-i}{d}\right)\gamma, \quad \text{for } i = 0, \ldots k-1$
- Hence the constraints now becomes $\sum_{i=0}^{k-1} min\{b_i, \alpha\} \geq B$
- The LHS is a piecewise linear function of alpha:

$$C(\alpha) = \begin{cases} k\alpha, & \alpha \in [0, b_0] \\ b_0 + (k-1)\alpha, & \alpha \in (b_0, b_1] \\ \vdots & \\ b_0 + \ldots + b_{k-2} + \alpha, & \alpha \in (b_{k-2}, b_{k-1}] \\ b_0 + \ldots + b_{k-1}, & \alpha \in (b_{k-1}, \infty) \end{cases}$$

LHS of constraint as function of α



LHS of constraint as function of α

- $C(\alpha)$ is strictly increasing function of α
- It has an inverse which is also a strictly increasing function.
- Hence for a given lower bound on min-cut, say B, $\alpha^* = C^{-1}(B)$

$$\alpha^* = \begin{cases} \dfrac{B}{k} & B \in [0, kb_0) \\ \dfrac{B - b_0}{k-1} & B \in (kb_0, b_0 + (k-1)b_1] \\ \vdots & \vdots \\ B - \sum_{j=0}^{k-2} b_j & B \in \left(\sum_{j=0}^{k-2} b_j + b_{k-2}, \sum_{j=0}^{k-1} b_j \right] \end{cases}$$

Solution to the optimization

In general, for $i = 1, \ldots, k-1$ the ith condition is:

$$\alpha^* = \frac{B = \sum_{j=0}^{i-1} b_j}{k - i}$$

$$for \quad B \in \left(\sum_{j=0}^{i-1} b_j + (k-i) b_{i-1}, \sum_{j=0}^{i} b_j + (k-i-1) \right]$$

Simplification of solution

$$
\begin{aligned}
\sum_{j=0}^{i-1} &= \sum_{j=0}^{i-1}(1 - \frac{k-1-j}{d})\gamma \\
&= \gamma \left[i \left(1 - \frac{k-1}{d} \right) + \frac{i(i-1)}{2d} \right] \\
&= \gamma i \frac{2d - 2k + i + 1}{2d} \\
&= \gamma g(i)
\end{aligned}
$$

Simplification of solution

$$\sum_{j=0}^{i} b_j + (k-i-1)b_i$$

$$= \gamma(i+1)\frac{2d-2k+i+2}{2d}$$

$$+ (k-i-1)\gamma\left(1 - \frac{k-1-i}{d}\right)$$

$$= \gamma\frac{2ik - i^2 - i + 2k + 2kd - 2k^2}{2d}$$

$$= \gamma\frac{B}{f(i)}$$

Solution

$$\alpha^*(d,\gamma) = \begin{cases} \dfrac{\mathcal{M}}{k}, & \gamma \in [f(0), +\infty] \\ \dfrac{\mathcal{M} - g(i)\gamma}{k-i}, & \gamma \in [f(i), f(i-1)) \end{cases}$$

where

$$f(i) \triangleq \frac{2d}{2ik - i^2 - i + 2k + 2kd - 2k^2}$$

$$g(i) \triangleq i\frac{2d - 2k + i + 1}{2d}$$

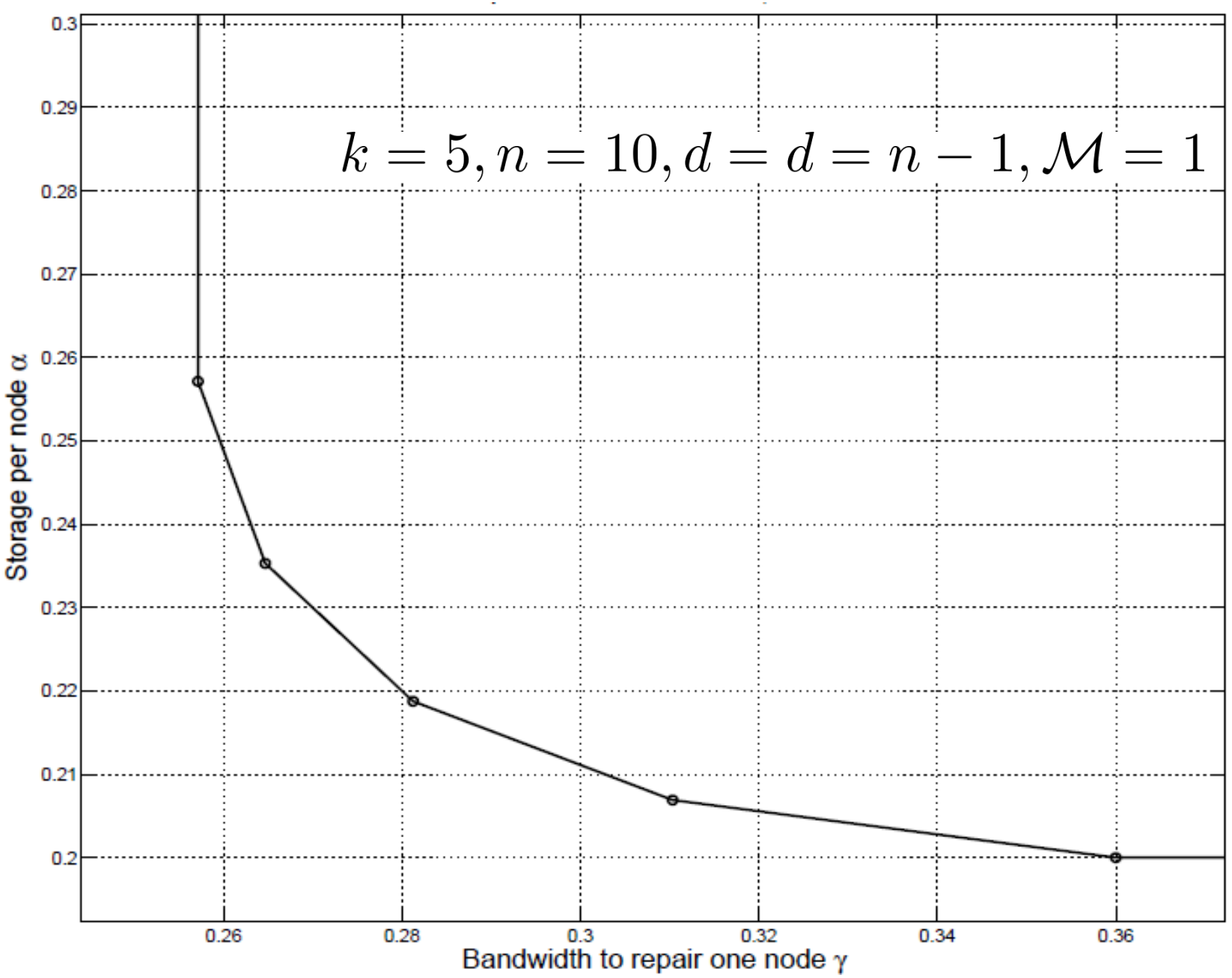
where $d \leq n - 1$

Minimum repair bandwidth

$$\gamma_{min} = f(k-1) = \frac{2\mathcal{M}d}{2kd - k^2 + k}$$

Storage-Bandwidth Tradeoff

Relationship between α and γ [1]



References

- [1]Alexandros G. Dimakis, P. Brighten Godfrey, Yunnan Wu, Martin J. Wainwright, and Kannan Ramchandran. 2010. Network coding for distributed storage systems. IEEE Trans. Inf. Theor. 56, 9 (September 2010), 4539-4551.
- [2]Koetter, R.; Medard, M., "An algebraic approach to network coding," Networking, IEEE/ACM Transactions on , vol.11, no.5, pp.782,795, Oct. 2003
- [3]Tracey Ho and Desmond Lun. 2008. Network Coding: An Introduction. Cambridge University Press, New York, NY, USA.
- [4]Dimakis, A.G.; Ramchandran, K.; Wu, Y.; Changho Suh, "A Survey on Network Codes for Distributed Storage," Proceedings of the IEEE , vol.99, no.3, pp.476,489, March 2011