Problems of Network Coding in P2P
- and how to overcome it

Christian Schindelhauer

joint work with
Christian Ortolf &
Arne Vater

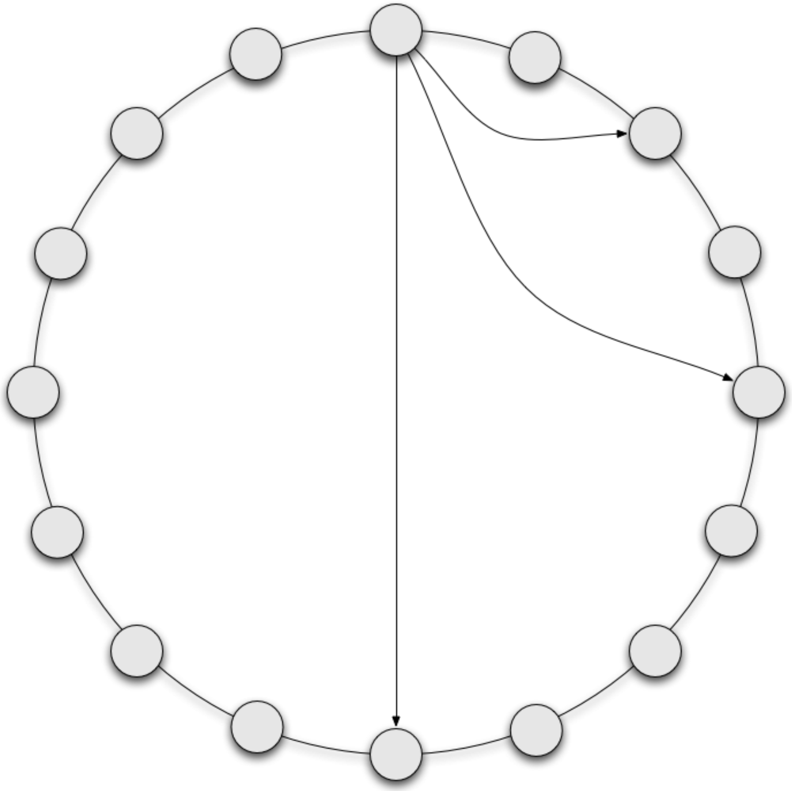
presented in SPAA 09 & 10

CoNe
Freiburg

Albert-Ludwig University Freiburg
Department of Computer Science
Computer Networks and Telematics

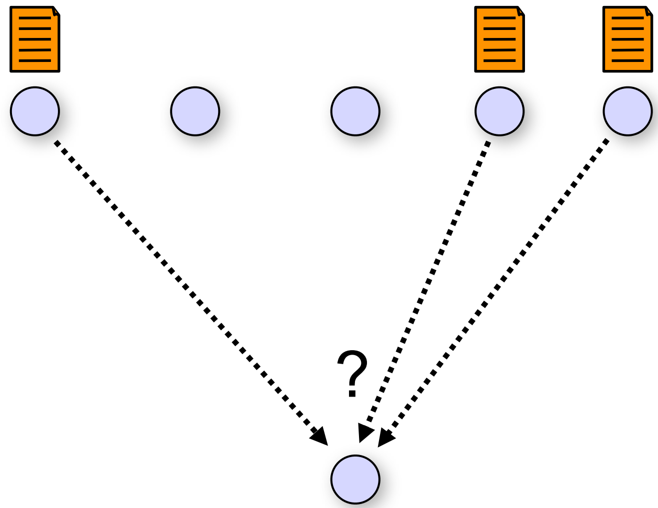
Motivation

- Peer-to-peer networks
  - distributed system
  - equal participants (peers)
  - no client/server structure
  - used for
    - communication (i.e. Skype)
    - data storage (i.e. OceanStore)
    - file sharing



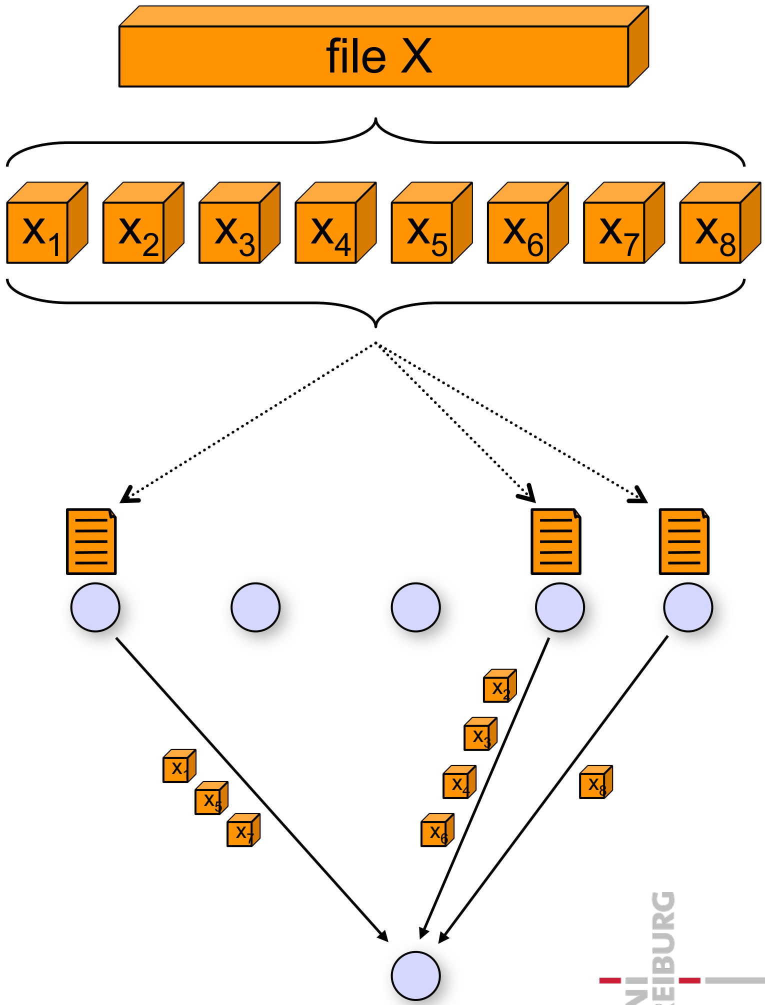
File Sharing

- A file is stored at one or more peers
- One or more different peers want to download this file
  - but from which peer?
- Problems
  - bandwidth
  - fairness
  - concurrency



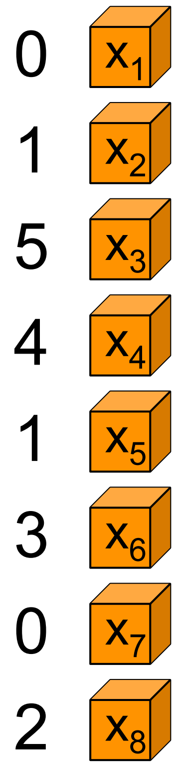
Block-Based

- Bram Cohen: BitTorrent
- Block-based file sharing system
  - divide file into blocks
  - efficient download
  - efficient usage of upload
  - usage of several sources
  - fairness amongst peers
  - uses implicit multicast trees for the distribution of blocks



BitTorrent Block Selection

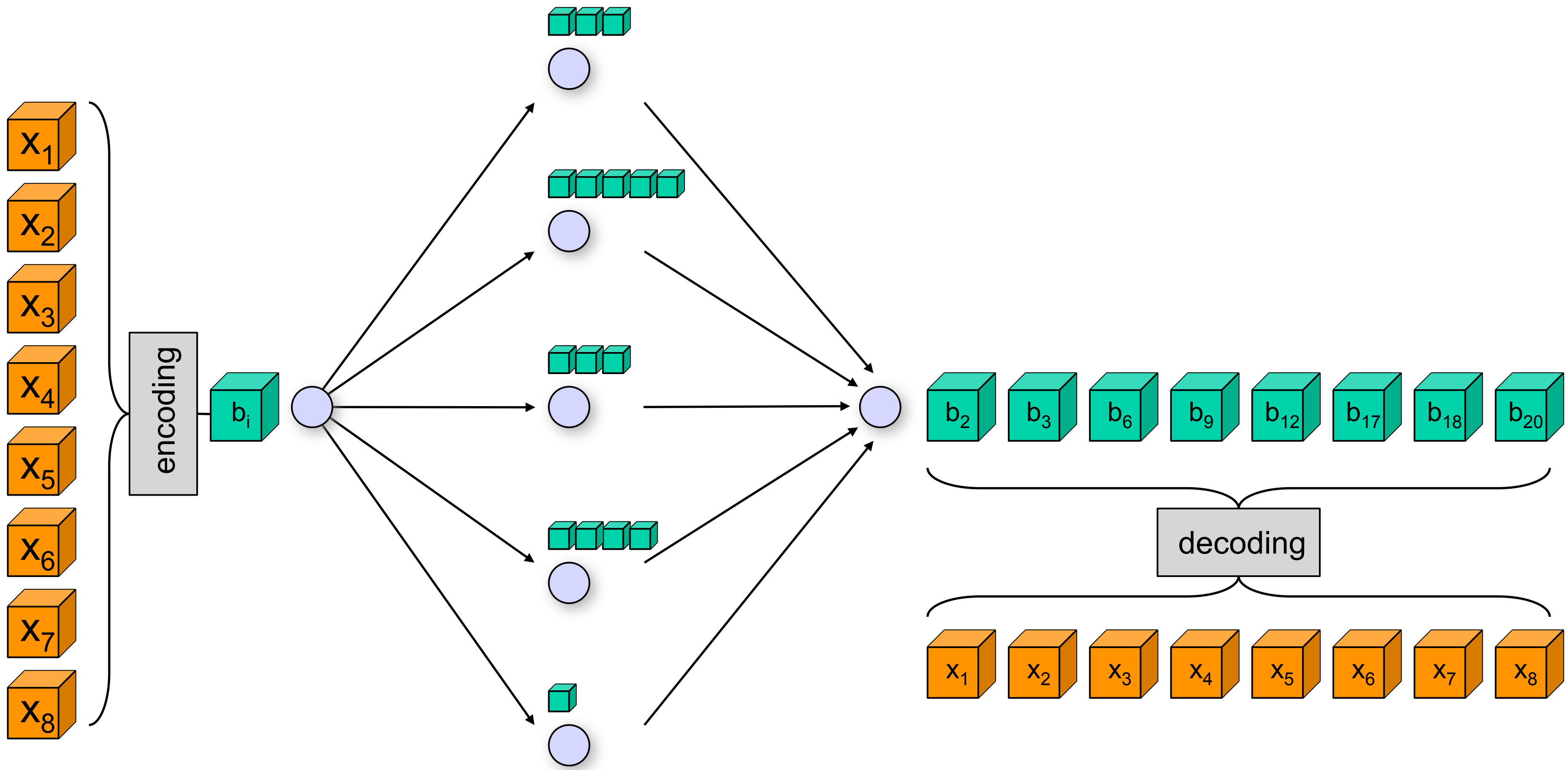
- Coupon-Collector-Problem
  - reason for unevenly distributed blocks
  - if blocks chosen randomly
- Measures: Policies
  - heuristics to select blocks for distribution
  - different policies depending on progress
    - random first
    - rarest first
    - endgame
  - very successful for popular files



Solution: Network Coding

- Optimal solution for Coupon-Collector-Problem / Policy
  - optimal network flow
    - Ahlswede, Cai, Li, and Yeung, "Network Information Flow"
  - practical network coding
    - Gkantsidis, and Rodriguez, "Network coding for large scale content distribution“
- Method
  - sender transmits code blocks as linear combinations of the file‘s blocks
  - receiver collects code blocks and reconstructs the original file

Coding and Decoding (I)



Coding and Decoding (II)

- file X = (x_1, x_2, ..., x_n)
- linear coefficients c_{ij}
- code blocks b_1, b_2, ..., b_n

$$(c_{i1}, \ldots, c_{in}) \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = b_i$$

$$\begin{pmatrix} c_{11} & \ldots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \ldots & c_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} c_{11} & \ldots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \ldots & c_{nn} \end{pmatrix}^{-1} \cdot \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

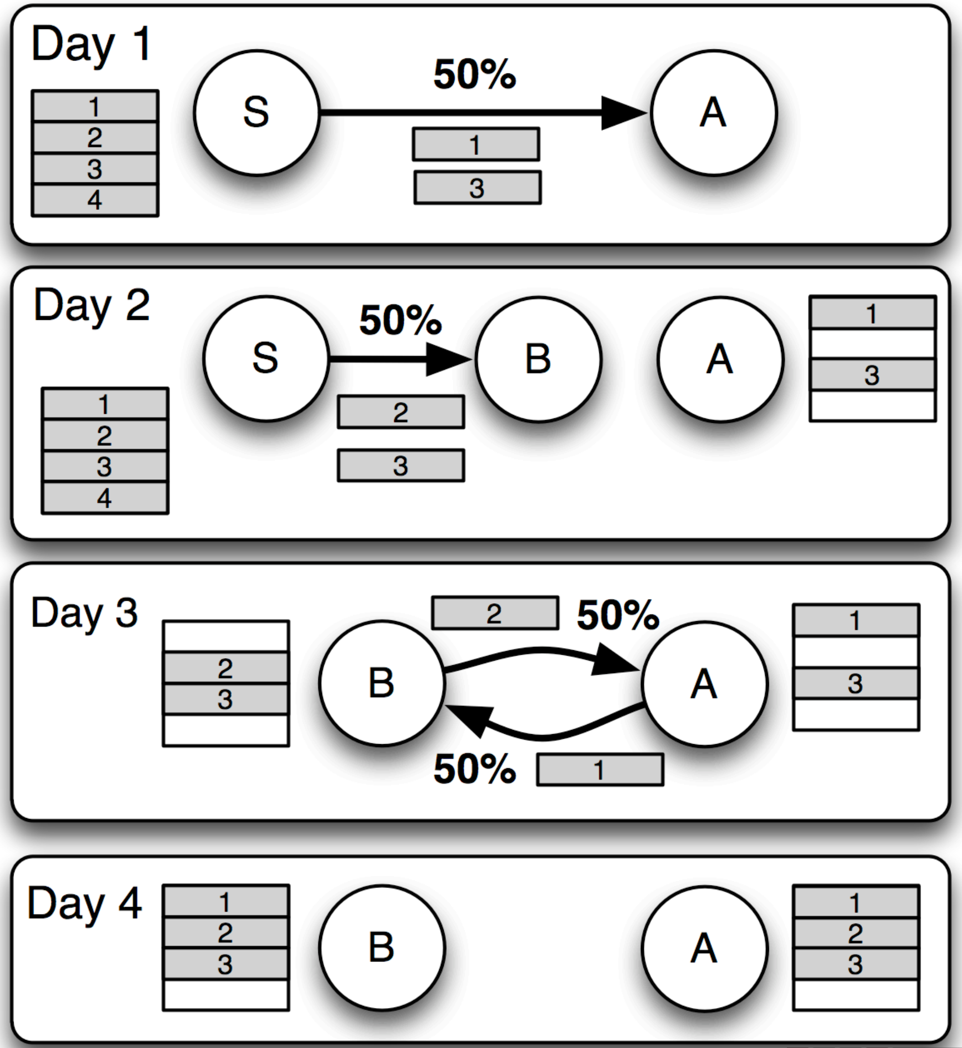
- if the matrix is invertable then

Problems of Network Coding

- Overhead of storing linear coefficients
  - one per block
  - e.g. 4 GB file with 100 KB blocks
    - 4 GB / 100 KB = 40 KB per block
    - overhead 40%
  - better: 4 GB file and 1 MB blocks
    - 4 KB overhead = 0.4%
- Overhead of decoding
  - Inversion of an (n×n)-matrix needs time $O(n^3)$
- Read/write accesses
  - writing n blocks requires reading each part n times: $O(n^2)$
  - disk access is much slower than memory access

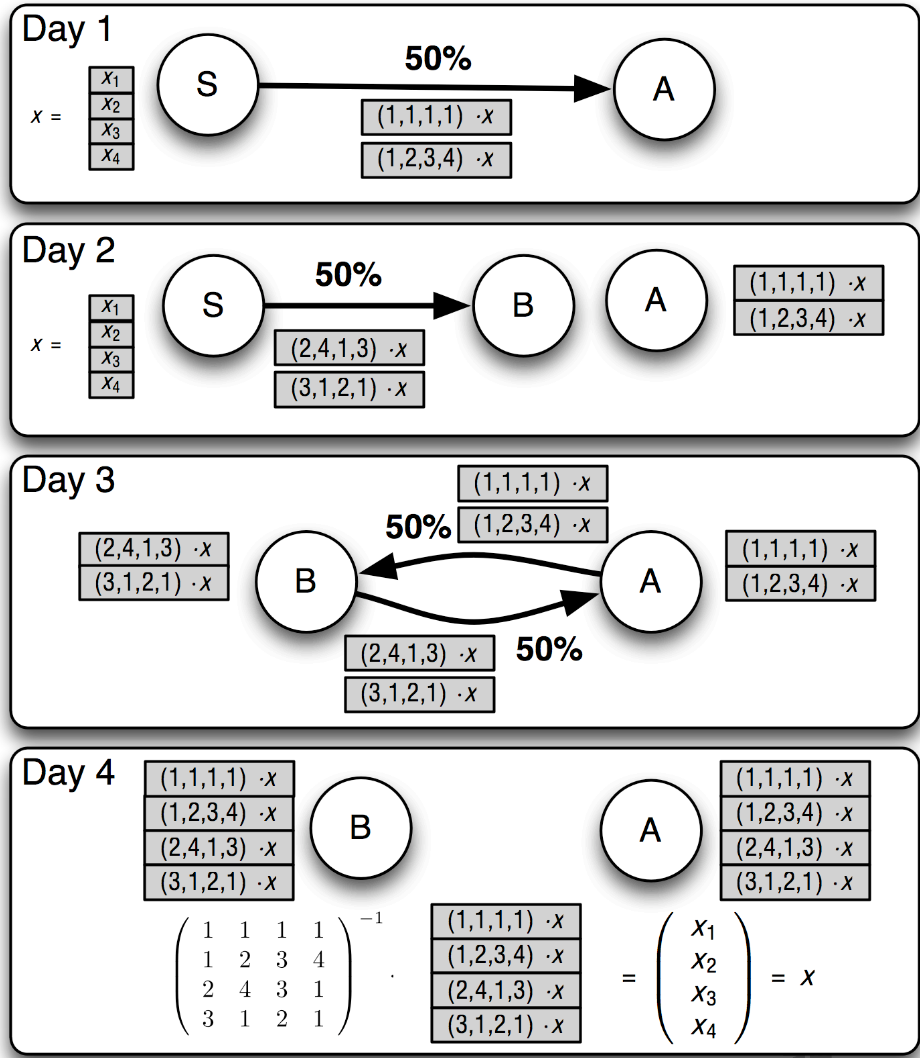
Scenario

- BitTorrent is optimal regarding disk access and computation overhead,
  - but it suffers from the coupon collector problem (availability).



Scenario

- Network Coding is optimal regarding availability
  - but it has a high computational overhead as well as high disk access overhead

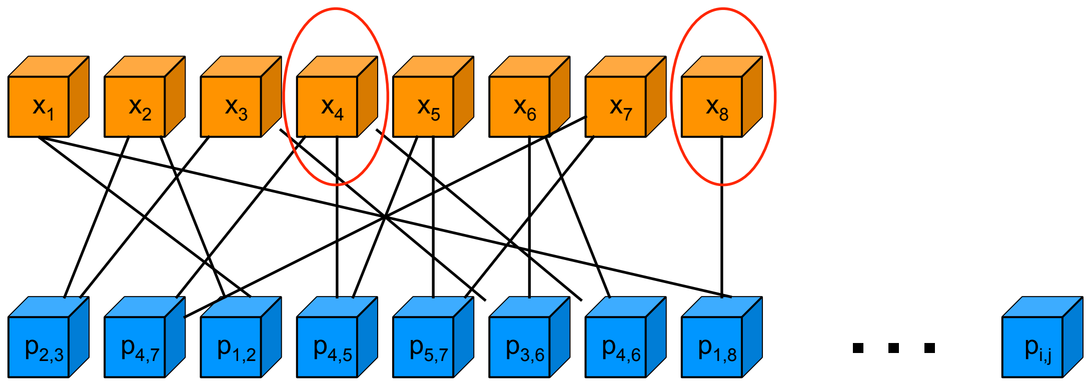


Our Goal

- We want to find a coding scheme that
  - performs better than BitTorrent regarding availability, and
  - requires less read/write accesses than Network Coding.

Paircoding

- Paircoding
  - is a reduced form of Network Coding
  - combines only two original blocks into one code block
    - $p_{i,j} = c_i\ x_i + c_j\ y_j$

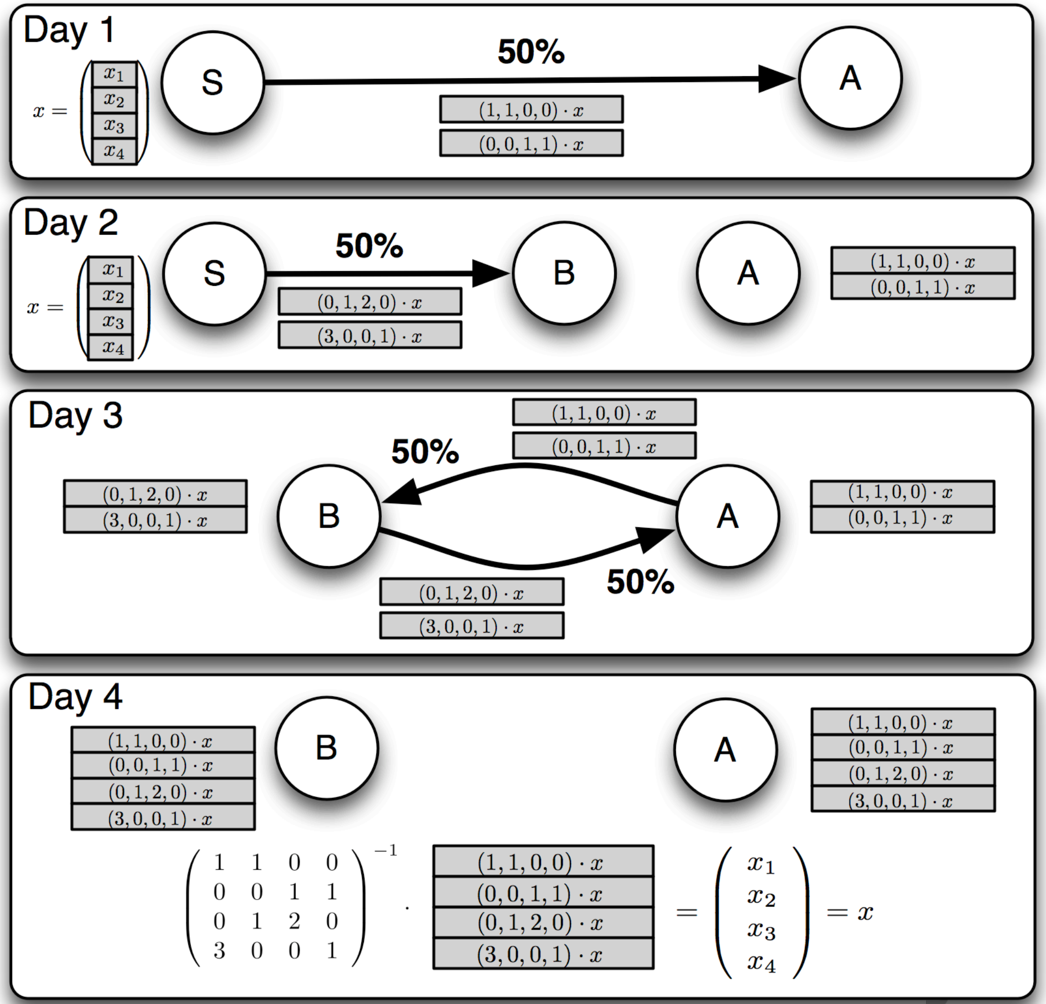


Advantages

- Easy code block creation
  - only two original blocks must be read
- Recoding
  - new code blocks can be created from different code blocks
  - no prior decoding necessary
- Decoding
  - requires little computation (sparse matrix inversion)
  - can be done lazy
- Coding alleviates the coupon collector problem

Scenario

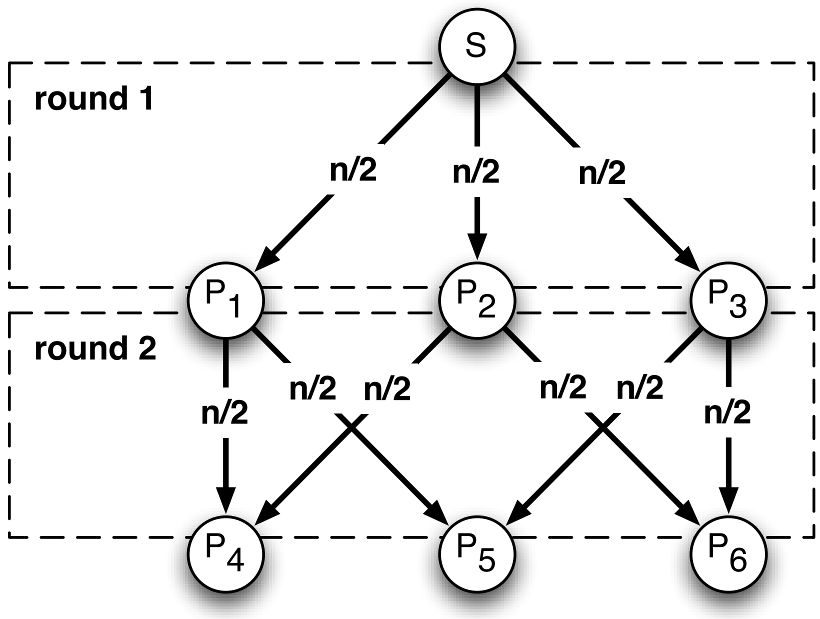
- Paircoding outperforms BitTorrent
  - with less overhead than Network Coding



BitTorrent: Bad Performance

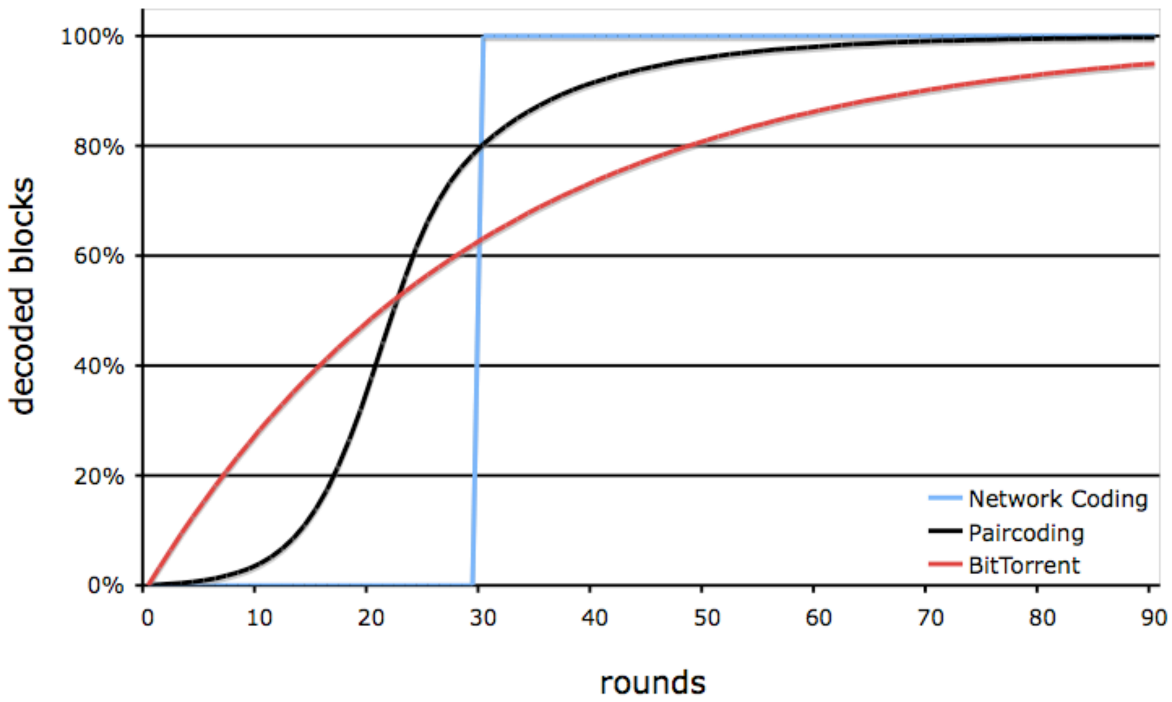
Theorem

There is a communication graph of depth 2 where BitTorrent fails in the offline model.



Simulation (I)

- one seed
- one downloading peer
- seeder sends one random block in each round

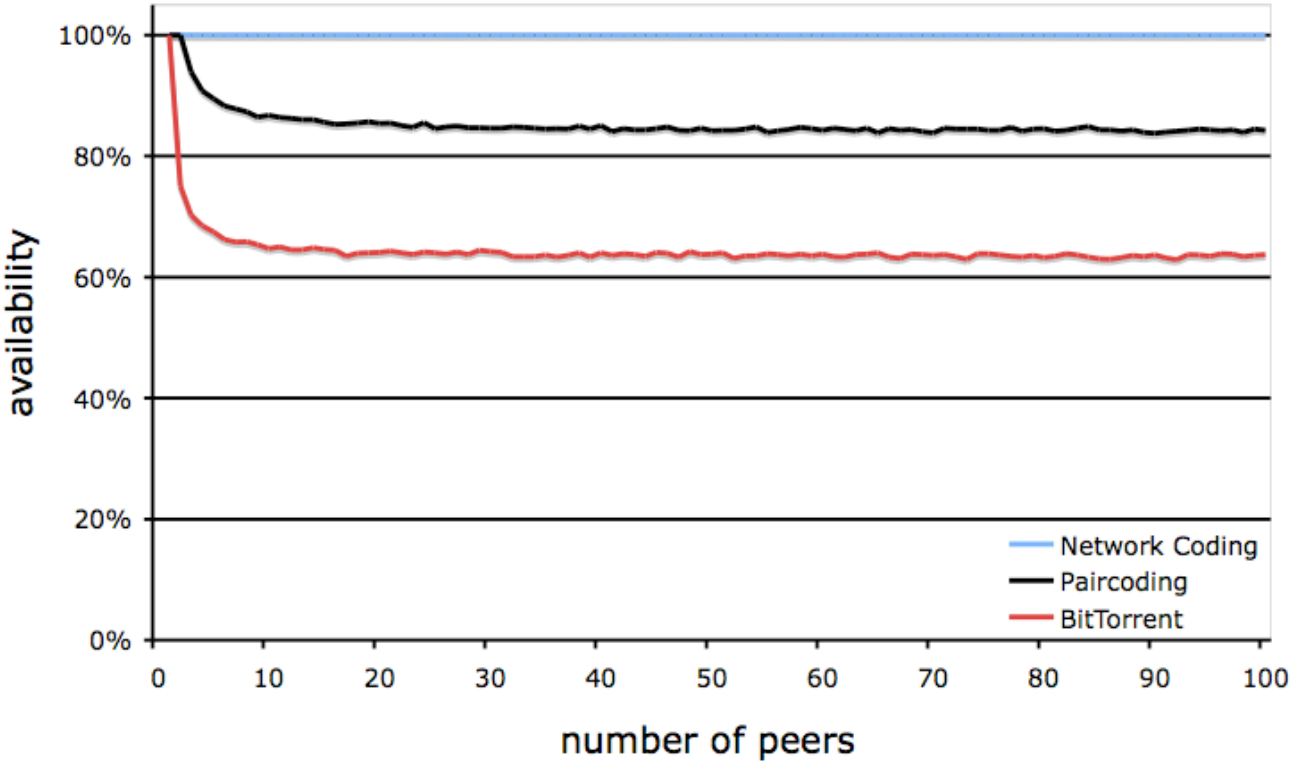


Measurements

- progress of a peer
  - number of linearly independent code blocks divided by n
  - (1 - progress) • n
    is the number of missing blocks
- availability at a set of peers
  - number of linearly independent code blocks at all peers of the set divided by n
  - (1 - availability) • n
    is the number of missing blocks at all peers after exchanging all available blocks
- read/write cost
  - total number of blocks to read from (or write to) disk to decode the original file

Simulation (II)

- each peer receives n/p blocks from a seed
  - rounded, such that the total amount of blocks equals n
  - coordination within peer allowed, otherwise random selection



CoNe Freiburg

Policy

- Policy
  - algorithmic choice of creating and uploading a code block depending on
    - receiving peer
    - current network configuration
    - availability
    - progress
- Policy of BitTorrent
  - optimize throughput and fairness
- Policy of Practical Network Coding
  - always send linearly independent code blocks
  - optimal

UNI FREIBURG

Outperforming

A file sharing system A is at least as good as B,

$A \geq B$

if for every scenario and every policy of B there is a policy in A such that A performs at least as well as B.

If $A \geq B$ and there exists a scenario in which A has larger progress than B, A outperforms B.

$A > B$

Scenario

- Network Coding is optimal regarding availability
  - but it has a high computational overhead as well as high disk access overhead.
- BitTorrent is optimal regarding disk access and computation overhead,
  - but it suffers from the coupon collector problem (availability).
- Forward error correction is in-between, depending on κ

Our Goal

- We want to find a coding scheme that
  - performs better than BitTorrent regarding availability, and
  - requires less read/write accesses than Network Coding.

Advantages

- Easy code block creation
  - only two original blocks must be read
- Recoding
  - new code blocks can be created from different code blocks
  - no prior decoding necessary
- Decoding
  - requires little computation (sparse matrix inversion) and disk access
  - can be done lazily
- Coding alleviates the coupon collector problem

CoNe Freiburg

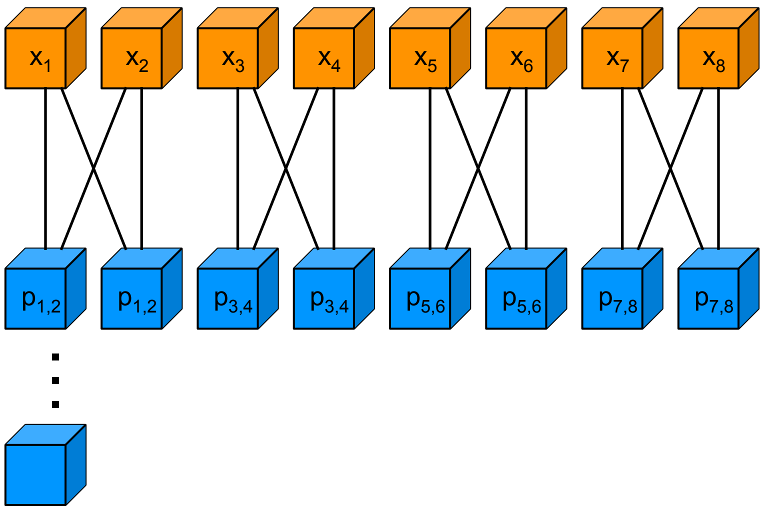
Paircoding - Results

- Theorems
  - Paircoding outperforms BitTorrent
    - For all scenarios and any BitTorrent policy, Paircoing is at least as efficient as BitTorrent.
    - For some scenarios Paircoding is more efficient than Bittorrent, i.e. Paircoding outperforms BitTorrent.
  - Encoding and decoding can be performed with an almost linear number of read/write operations: $O(n \bullet \alpha(n))$.
    - $\alpha(n)$ is the inverse Ackerman function

UNI FREIBURG

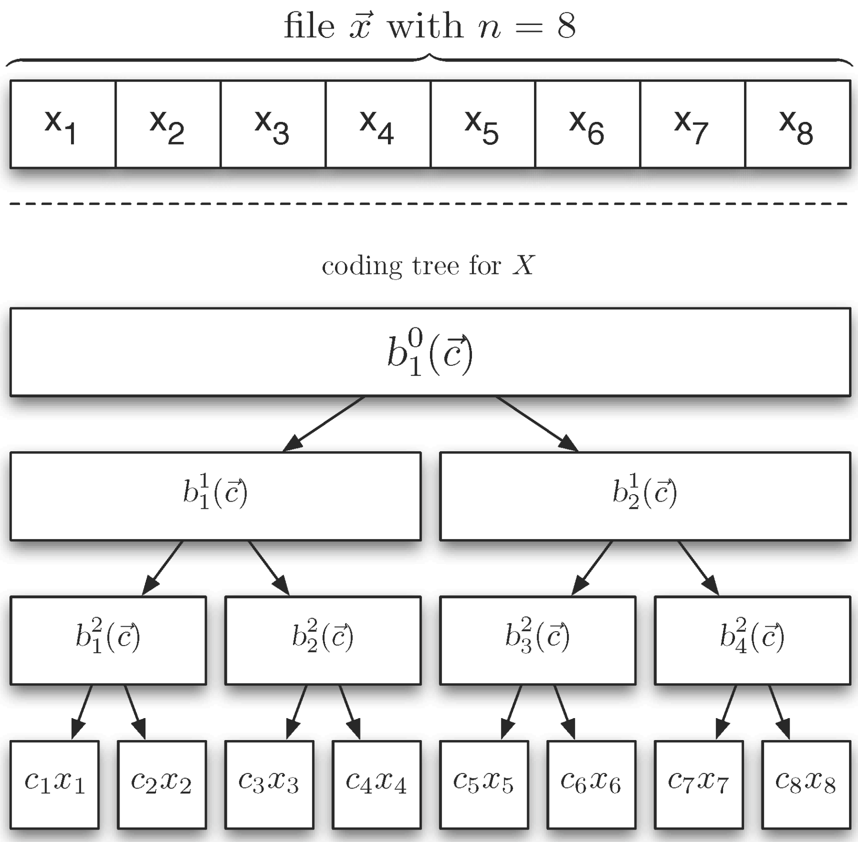
Fixed Paircoding

- restricted version of Paircoding
  - $p_{2i-1,2i} = c_{2i-1}\, x_{2i-1} + c_{2i}\, y_{2i}$
- also outperforms BitTorrent
- is outperformed by Paircoding
- has linear read/write cost



Treecoding(κ)

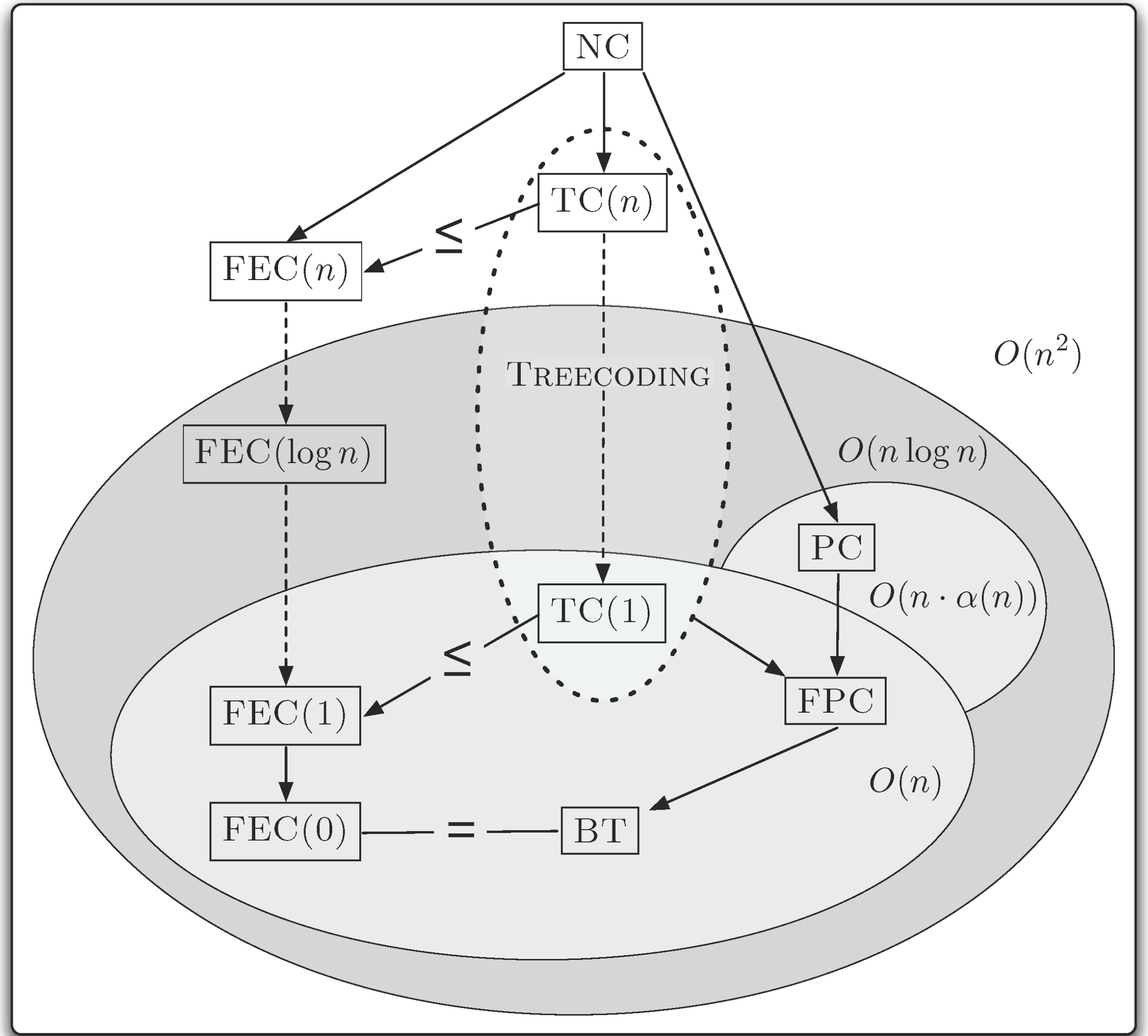
- tree structure
  - fixed linear coefficients for all blocks x_i
  - Xor of two nodes creates parent node
- κ different trees
  - with linearly independent linear coefficients
- root node is equivalent to a network coding block
- leaves are equivalent to uncoded blocks



Treecoding - Results

- Performance hierarchy
  - Treecoding(κ + 1) > Treecoding(κ)
- Treecoding performs as well as forward error correction
  - Treecoding(κ) ≥ FEC(κ)
- Treecoding outperforms Fixed Paircoding
  - $\bigcup_{\kappa}$ Treecoding(κ) > FixedPaircoding
  - if the number of trees is arbitrary
- Treecoding and Paircoding are incomparable
- Treecoding has read/write cost of
  - O(n), if $\kappa = 1$
  - $O(\kappa\, n \log^2 n)$, for any κ

Class Hierarchy



Problems of Network Coding in P2P
- and how to overcome it

Christian Schindelhauer

joint work with
Christian Ortolf &
Arne Vater

presented in SPAA 09 & 10

CoNe
Freiburg

Albert-Ludwig University Freiburg
Department of Computer Science
Computer Networks and Telematics