Lectures in Wroclaw

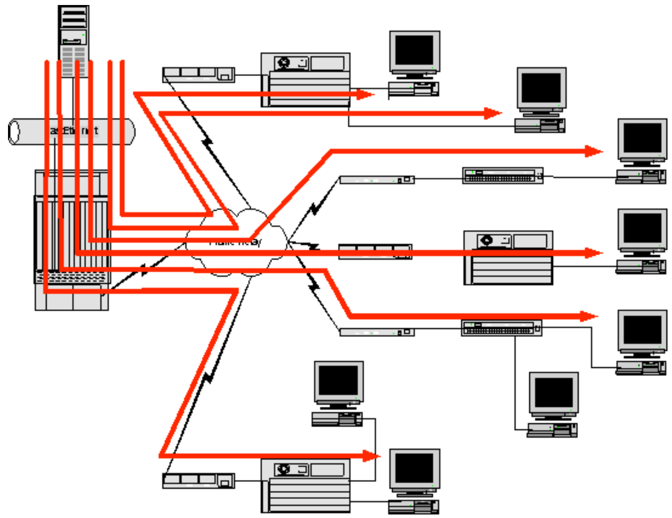
- **Epidemic Algorithms**
  - Monday, April 6th, 2009, 3pm
- **Random Networks**
  - Monday, April 6th, 2009, 6pm
- **Distributed Heterogeneous Hash Tables**
  - Tuesday, April 7th, 2009, 3pm
- **Network Coding**
  - Wednesday, April 8th, 2009, 11am
- **Locality in Peer-to-Peer Networks**
  - Wednesday, April 8th, 2009, 3pm

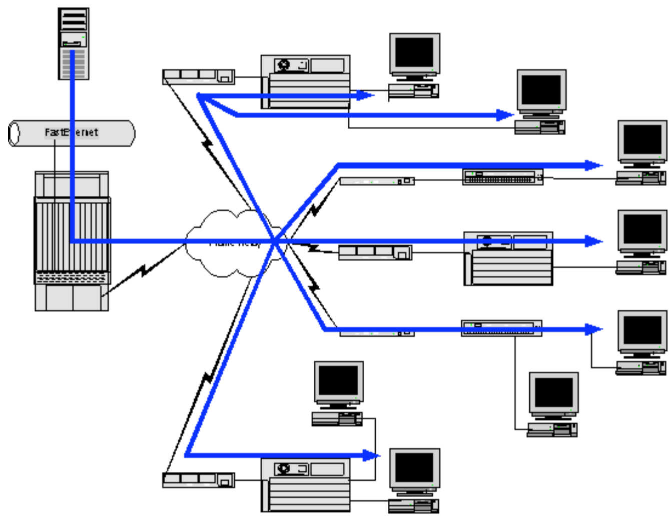
Peer to Peer Networks

Fast Download

IP Multicast

- **Motivation**
  - Transmission of a data stream to many receivers
- **Unicast**
  - For each stream message have to be sent separately
  - Bottleneck at sender
- **Multicast**
  - Stream multiplies messages
  - No bottleneck





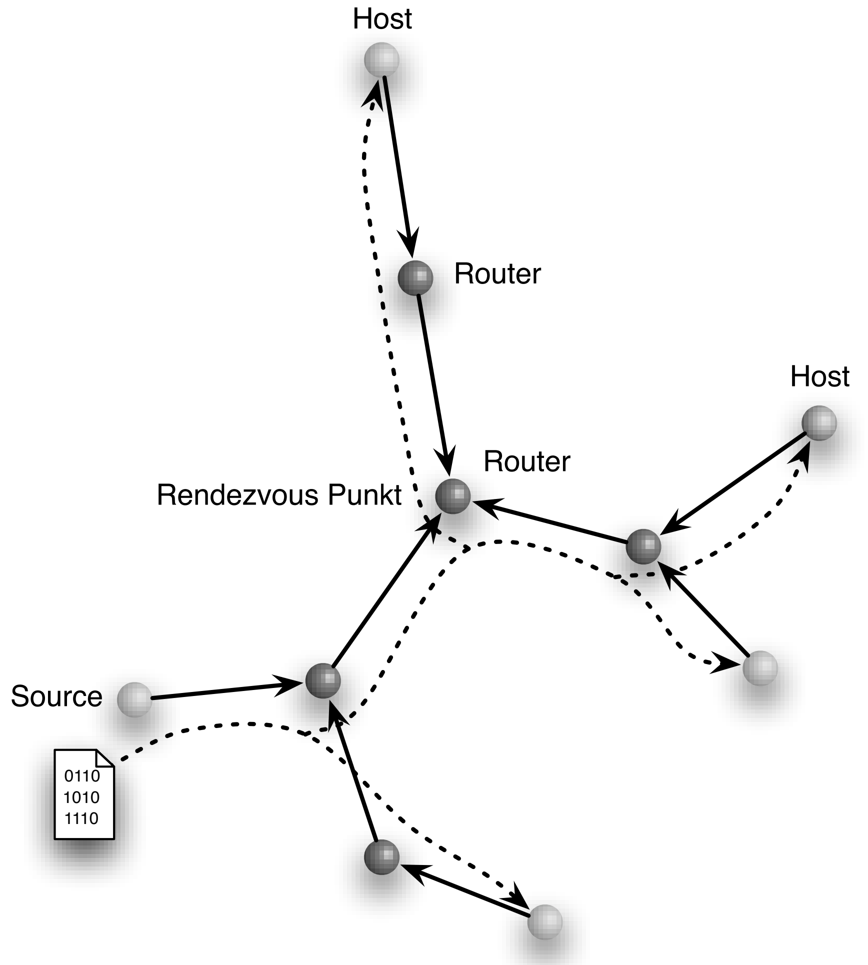
Peter J. Welcher
www.netcraftsmen.net/.../ papers/multicast01.html

Working Principle

- **IPv4 Multicast Addresses**
  - class D
    - outside of CIDR (Classless Interdomain Routing)
  - 224.0.0.0 - 239.255.255.255
- **Hosts register via IGMP at this address**
  - IGMP = Internet Group Management Protocol
  - After registration the multicast tree is updated
- **Source sends to multicast address**
  - Routers duplicate messages
  - and distribute them into sub-trees
- **All registered hosts receive these messages**
  - ends after Time-Out
  - or when they unsubscribe
- **Problems**
  - No TCP only UDP
  - Many routers do not deliver multicast messages
    - solution: tunnels

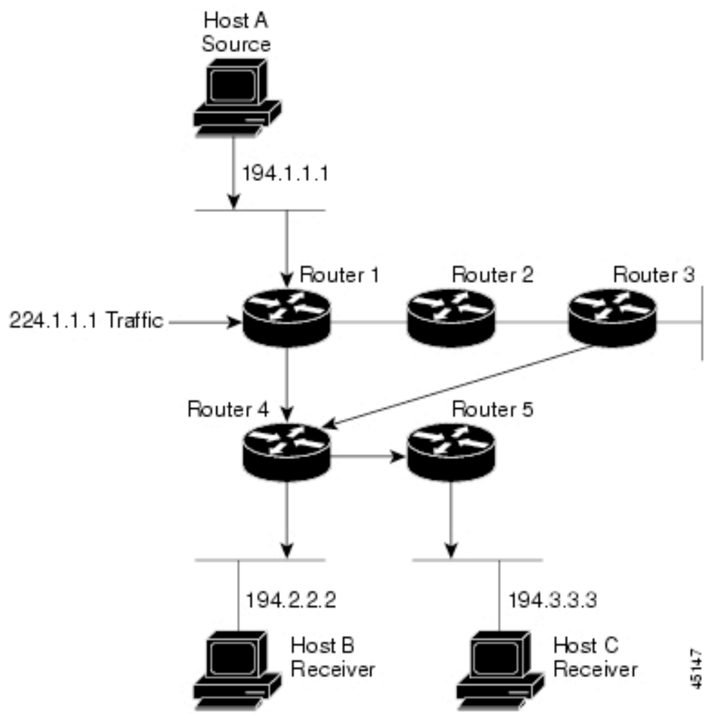
Routing Protocols

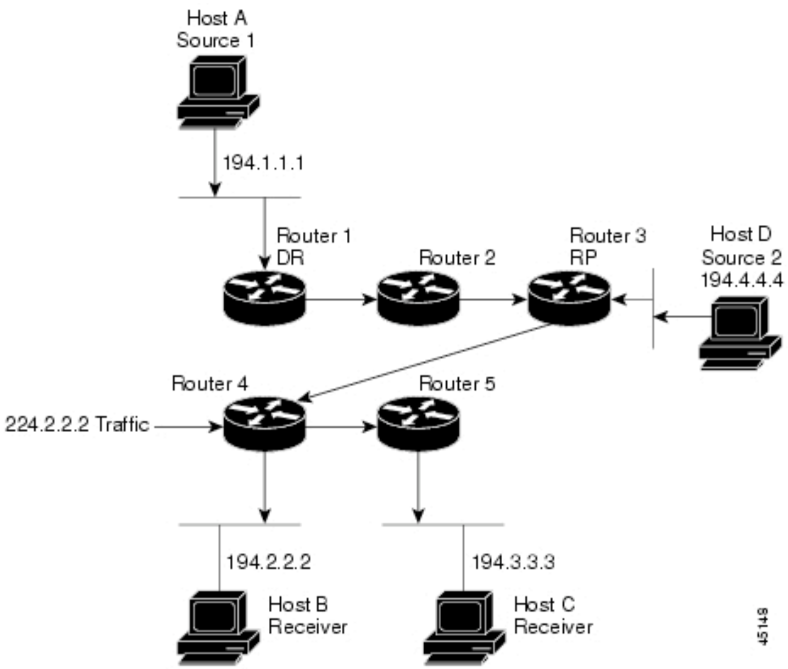
- **Distance Vector Multicast Routing Protocol (DVMRP)**
  - used for years in MBONE
    - particularly in Freiburg
  - own routing tables for multicast
- **Protocol Independent Multicast (PIM)**
  - in Sparse Mode (PIM-SM)
  - current (de facto) standard
  - prunes multicast tree
  - uses Unicast routing tables
  - is more independent from the routers
- **Prerequisites of PIM-SM:**
  - needs Rendezvous-Point (RP) in one hop distance
  - RP must provide PIM-SM
  - or tunneling to a proxy in the vicinity of the RP



PIM-SM
Tree Construction

- **Host A Shortest-Path-Tree**
- **Shared Distribution Tree**

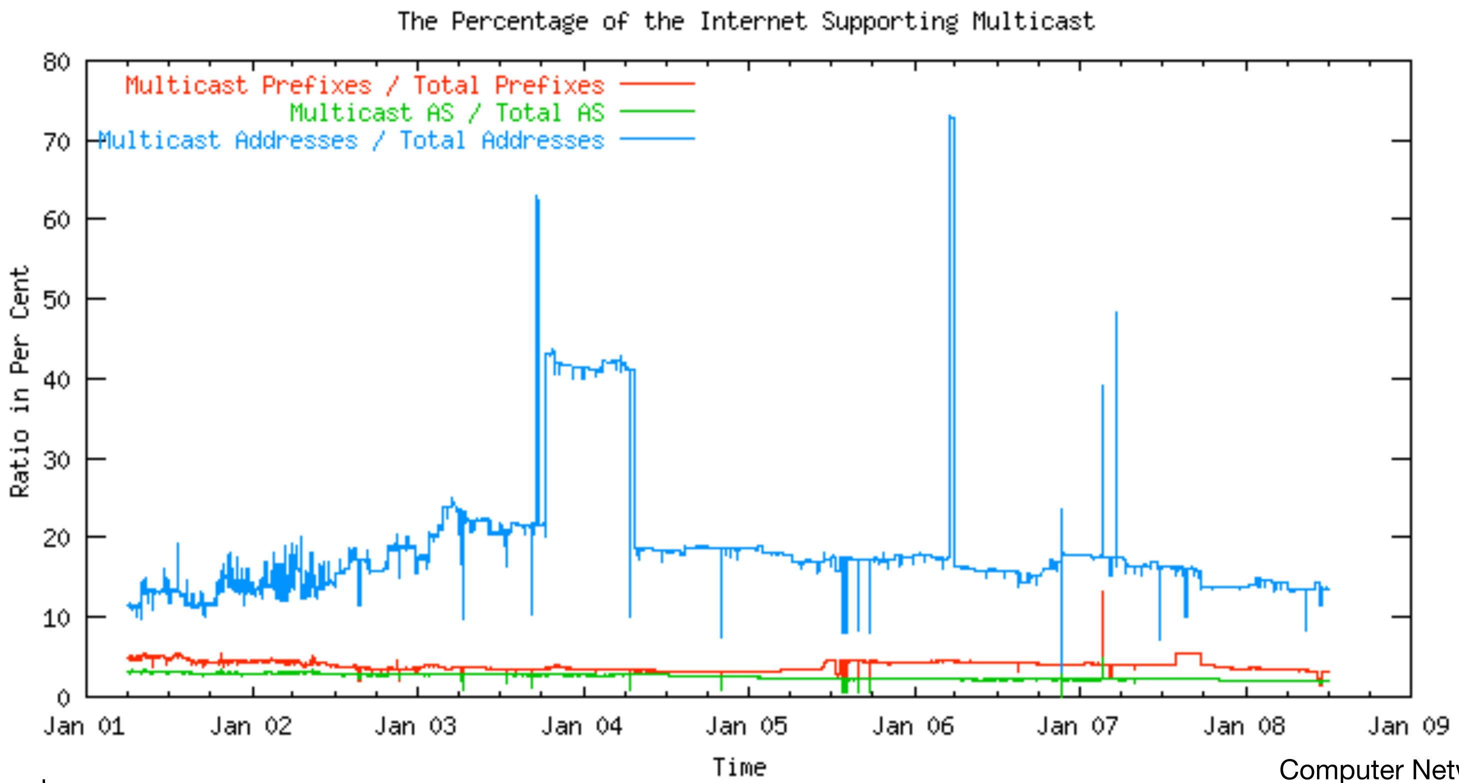




From Cisco: http://www.cisco.com/en/US/products/hw/switches/ps646/products_configuration_guide_chapter09186a008014f350.html

IP Multicast Seldomly Available

- **IP Multicast is the fastest download method**
- **Yet, not many routers support IP multicast**
  - –http://www.multicasttech.com/status/

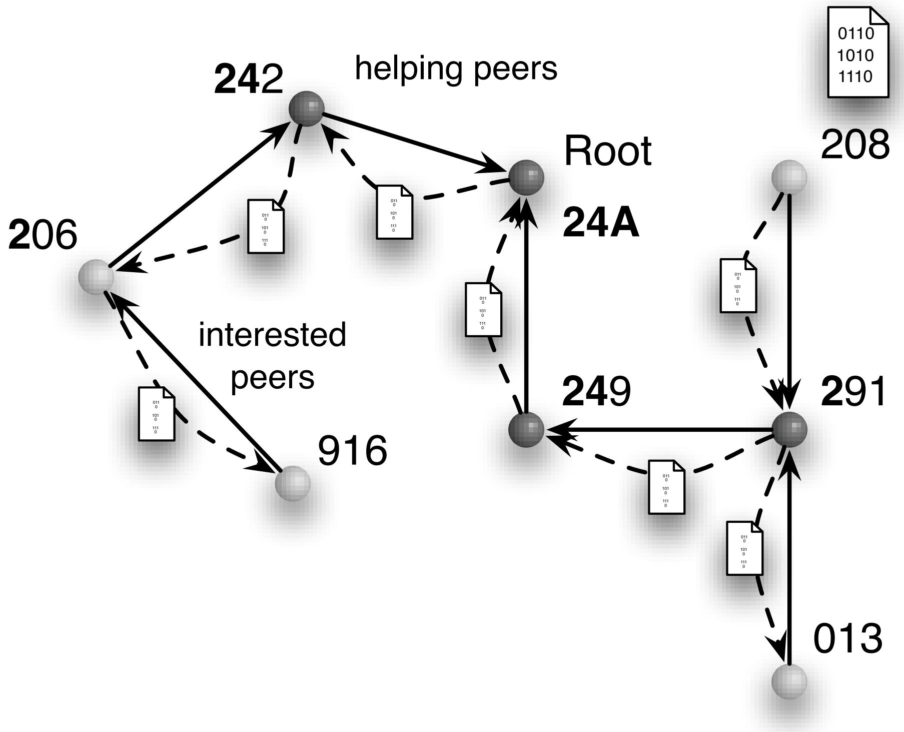


Why so few Multicast Routers?

- **Despite successful use**
  - in video transmission of IETF-meetings
  - MBONE (Multicast Backbone)
- **Only few ISPs provide IP Multicast**
- **Additional maintenance**
  - difficult to configure
  - competing protocols
- **Enabling of Denial-of-Service-Attacks**
  - Implications larger than for Unicast
- **Transport protocol**
  - only UDP
    - Unreliable
  - Forward error correction necessary
    - or proprietary protocols at the routers (z.B. CISCO)
- **Market situation**
  - consumers seldomly ask for multicast
    - prefer P2P networks
  - because of a few number of files and small number of interested parties the multicast is not desirable (for the ISP)
    - small number of addresses

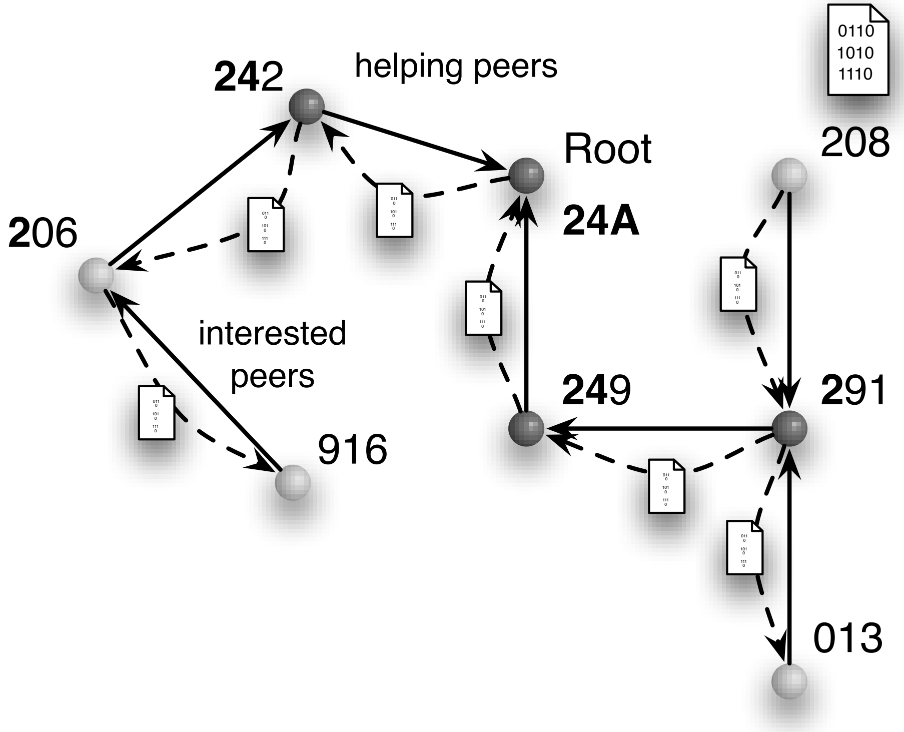
Scribe & Friends

- **Multicast-Tree in the Overlay Network**
- **Scribe [2001] is based on Pastry**
  - Castro, Druschel, Kermarrec, Rowstron
- **Similar approaches**
  - CAN Multicast [2001] based on CAN
  - Bayeux [2001] based on Tapestry
- **Andere Ansätze**
  - Overcast [´00] and Narada [´00]
  - construct multi-cast trees using unicast connections
  - do not scale



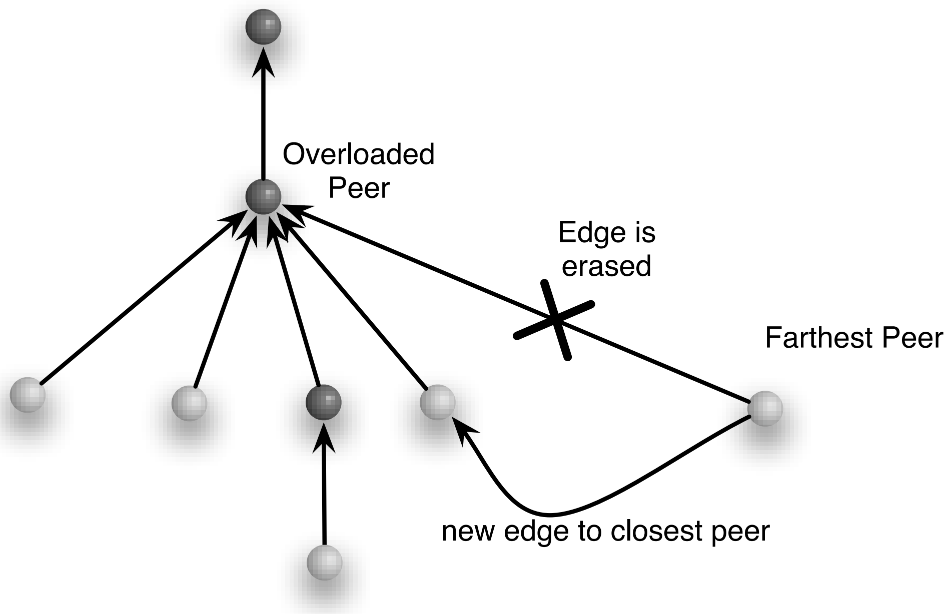
How Scribe Works

- **Create**
  - GroupID is assigned to a peer according to Pastry index
- **Join**
  - Interested peer performs lookup to group ID
  - When a peer is found in the Multicast tree then a new sub-path is inserted
- **Download**
  - Messages are distributed using the multicast tree
  - Nodes duplicate parts of the file



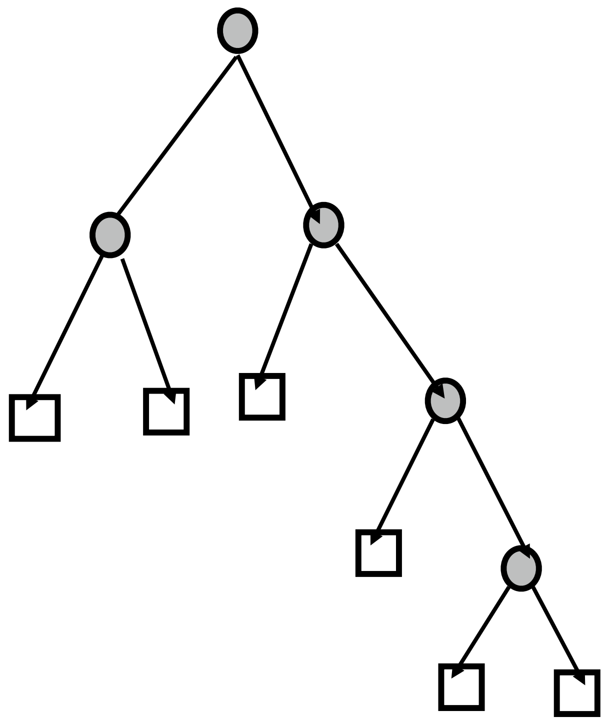
Scribe Optimization

- **Bottleneck-Remover**
  - If a node is overloaded then from the group of peers he sends messages
  - Select the farthest peer
  - This node measures the delay between it and the other nodes
  - and rebalances itself under the next (then former) brother



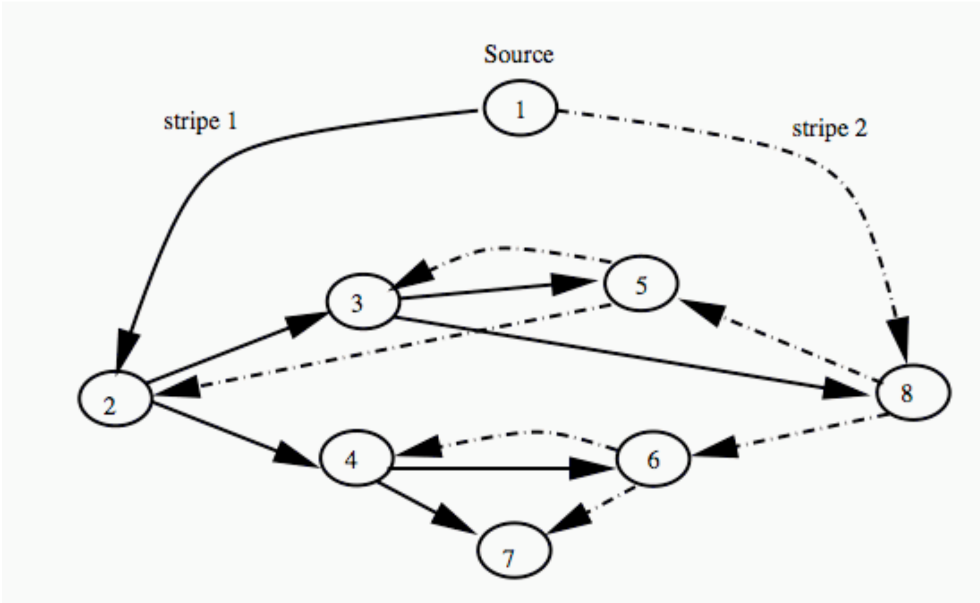
Split-Stream Motivation

- **Multicast trees discriminate certain nodes**
- **Lemma**
  - In every binary tree the number of leaves = number of internal nodes +1
- **Conclusion**
  - Nearly half of the nodes distribute data
  - While the other half does not distribute any data
  - An internal node has twice the upload as the average peer
- **Solution: Larger degree?**
- **Lemma**
  - In every node with degree d the number of internal nodes k und leaves b we observe
    - (d-1) k = b -1
- **Implication**
  - Less peers have to suffer more upload



Split-Stream

- **Castro, Druschel, Kermarrec, Nandi, Rowstron, Singh 2001**
- **Idea**
  - Partition a file of size into k small parts
  - For each part use another multicast tree
  - Every peer works as leave and as distributing internal tree node
    - except the source
- **Ideally, the upload of each node is at most the download**



Bittorrent

- **Bram Cohen**
- **Bittorrent is a real (very successful) peer-to-peer network**
  - concentrates on download
  - uses (implicitly) multicast trees for the distribution of the parts of a file
- **Protocol is peer oriented and not data oriented**
- **Goals**
  - efficient download of a file using the uploads of all participating peers
  - efficient usage of upload
    - usually upload is the bottleneck
    - e.g. asymmetric protocols like ISDN or DSL
  - fairness among peers
    - seeders against leeches
  - usage of several sources

Bittorrent
Coordination and File

- **Central coordination**
  - by tracker host
  - for each file the tracker outputs a set of random peers from the set of participating peers
    - in addition hash-code of the file contents and other control information
  - tracker hosts to not store files
    - yet, providing a tracker file on a tracker host can have legal consequences
- **File**
  - is partitions in smaller pieces
    - as describec in tracker file
  - every participating peer can redistribute downloaded parts as soon as he received it
  - Bittorrent aims at the Split-Stream idea
- **Interaction between the peers**
  - two peers exchange their information about existing parts
  - according to the policy of Bittorrent outstanding parts are transmitted to the other peer

Bittorrent
Part Selection

- **Problem**
  - The Coupon-Collector-Problem is the reason for a uneven distribution of parts
    - if a completely random choice is used
- **Measures**
  - Rarest First
    - Every peer tries to download the parts which are rarest
      - density is deduced from the comunication with other peers (or tracker host)
    - in case the source is not available this increases the chances the peers can complete the download
  - Random First (exception for new peers)
    - When peer starts it asks for a random part
    - Then the demand for seldom peers is reduced
      - especially when peers only shortly join
  - Endgame Mode
    - if nearly all parts have been loaded the downloading peers asks more connected peers for the missing parts
    - then a slow peer can not stall the last download

Bittorrent
Policy

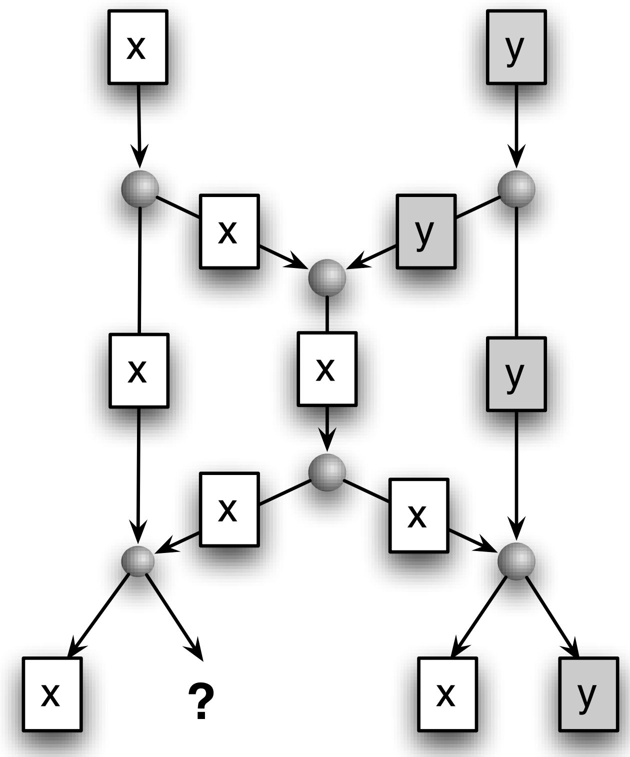
- **Goal**
  - self organizing system
  - good (uploading, seeding) peers are rewarded
  - bad (downloading, leeching) peers are penalized
- **Reward**
  - good download speed
  - un-choking
- **Penalty**
  - Choking of the bandwidth
- **Evaluation**
  - Every peers Peers evaluates his environment from his past experiences

Bittorrent
Choking

- **Every peer has a choke list**
  - requests of choked peers are not served for some time
  - peers can be unchoked after some time
- **Adding to the choke list**
  - Each peer has a fixed minimum amount of choked peers (e.g. 4)
  - Peers with the worst upload are added to the choke list
    - and replace better peers
- **Optimistic Unchoking**
  - Arbitrarily a candidate is removed from the list of choking candidates
    - the prevents maltreating a peer with a bad bandwidth

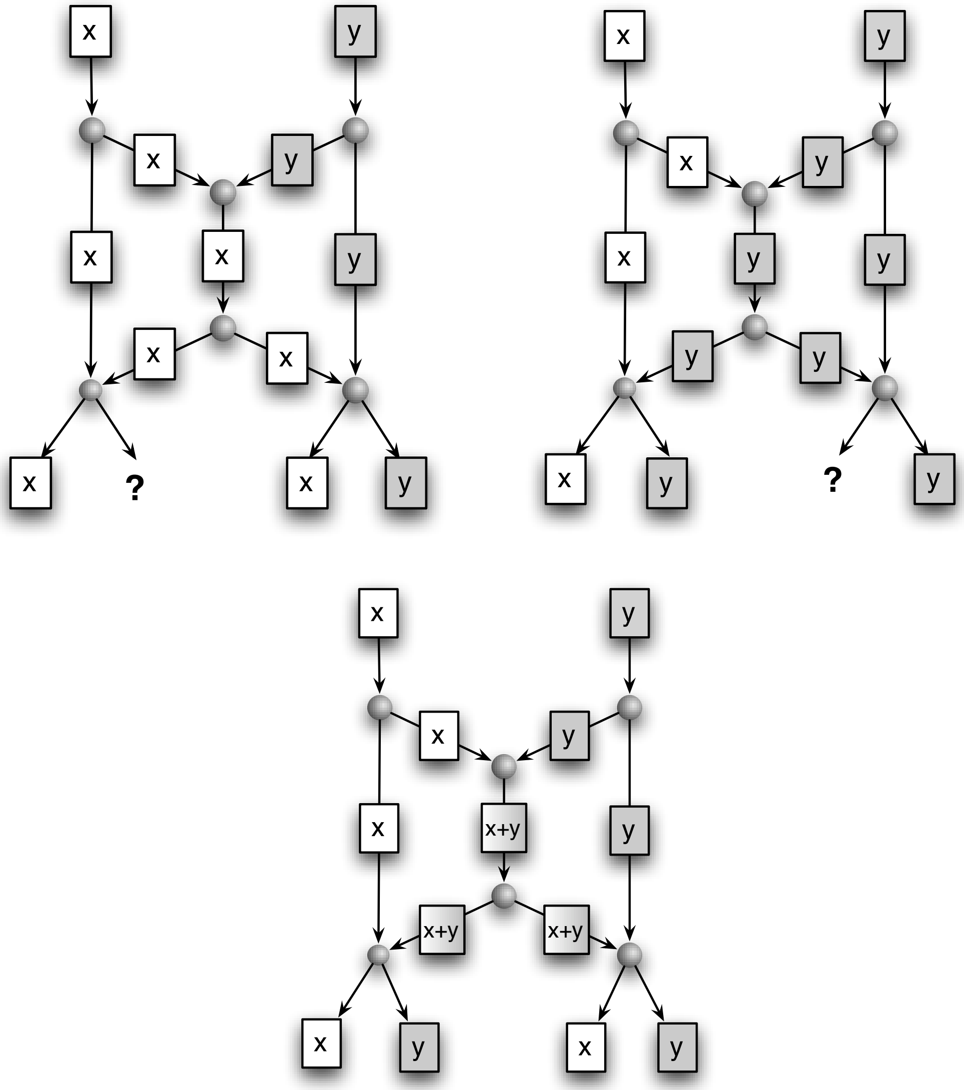
Network Coding

- **R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow", (IEEE Transactions on Information Theory, IT-46, pp. 1204-1216, 2000)**
- **Example**
  - Bits x and y need to be transmitted
  - Every line transmits one bit
  - If only bits are transmitted
    - then only x or y can be transmitted in the middle?
  - By using X we can have both results at the outputs



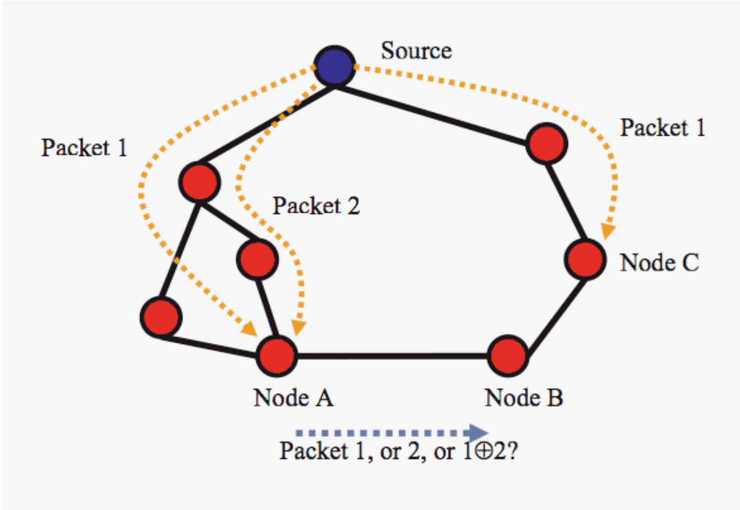
Network Coding

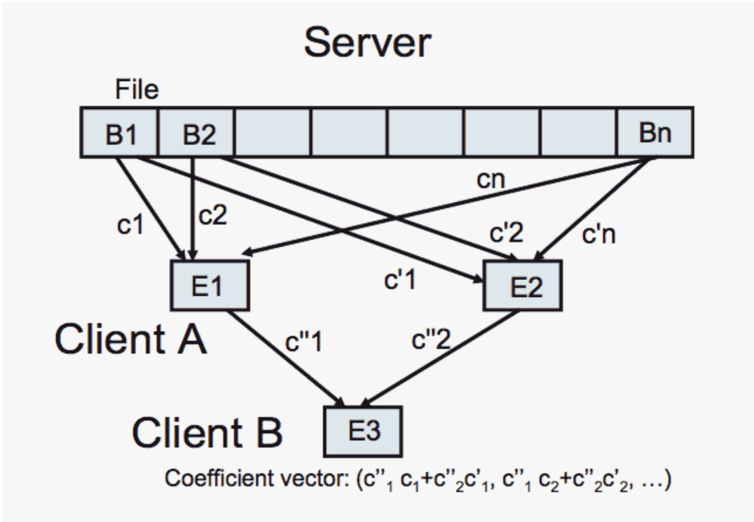
- **R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow", (IEEE Transactions on Information Theory, IT-46, pp. 1204-1216, 2000)**
- **Theorem [Ahlswede et al.]**
  - There is a network code for each graph such that each node receives as much information as the maximum flow of the corresponding flow problem



Practical Network Coding Avalanche

- **Christos Gkantsidis, Pablo Rodriguez Rodriguez, 2005**
- **Goal**
  - Overcoming the Coupon-Collector-Problem
    - a file of m parts can be always reconstructed if at least m network codes have been received
  - Optimal transmission of files within the available bandwidth
- **Method**
  - Use codes as linear combinations of a file
    - Produced code contains the vector and the variables
  - During the distribution the linear combination are re-combined to new parts
  - The receiver collects the linear combinations
  - and reconstructs the original file using matrix operations





Coding and Decoding

- File: x_1, x_2, ..., x_m
- Codes: y_1,y_2,...,y_m
- Random Variables r_{ij}

$$(r_{i1} r_{i2} \dots r_{im}) \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = y_i$$

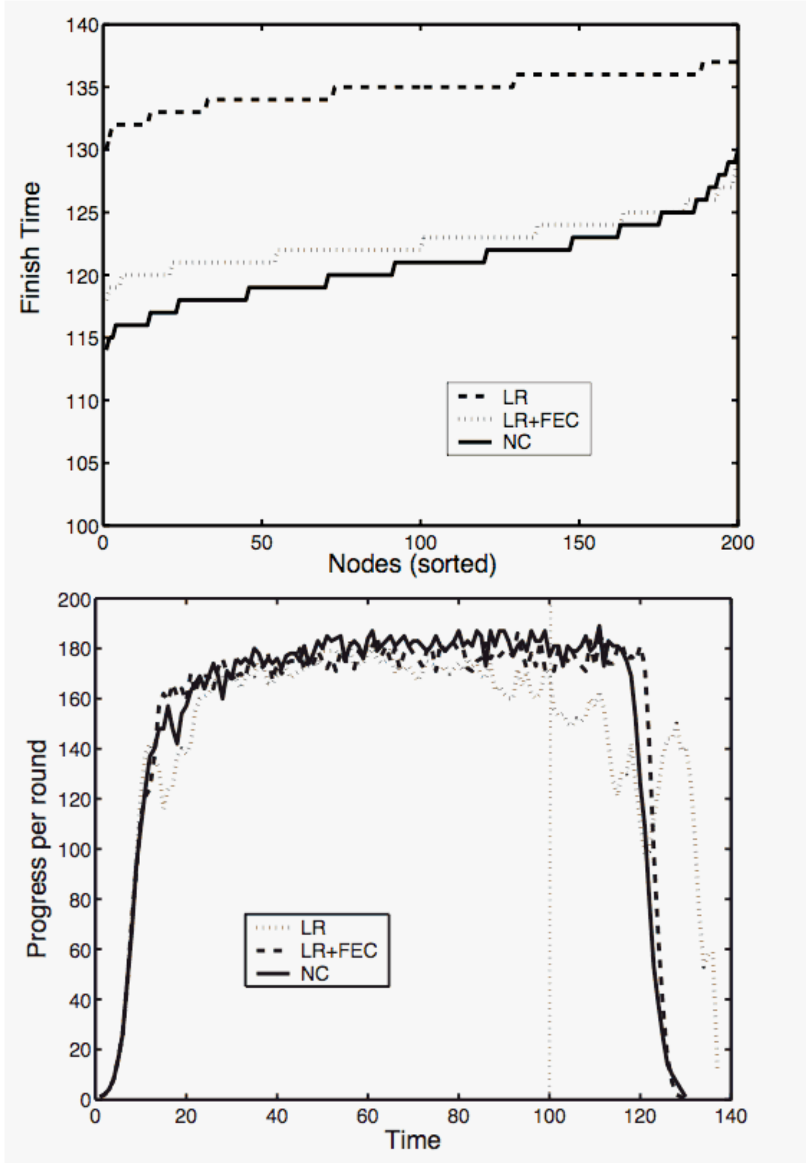
$$\begin{pmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mm} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$$

- If the matrix is invertable then

$$\begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mm} \end{pmatrix}^{-1} \cdot \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$$

Speed of Network-Coding

- **Comparison**
  - Network-Coding (NC) versus
  - Local-Rarest (LR) and
  - Local-Rarest+Forward-Error-Correction (LR+FEC)



Problems of Network-Coding

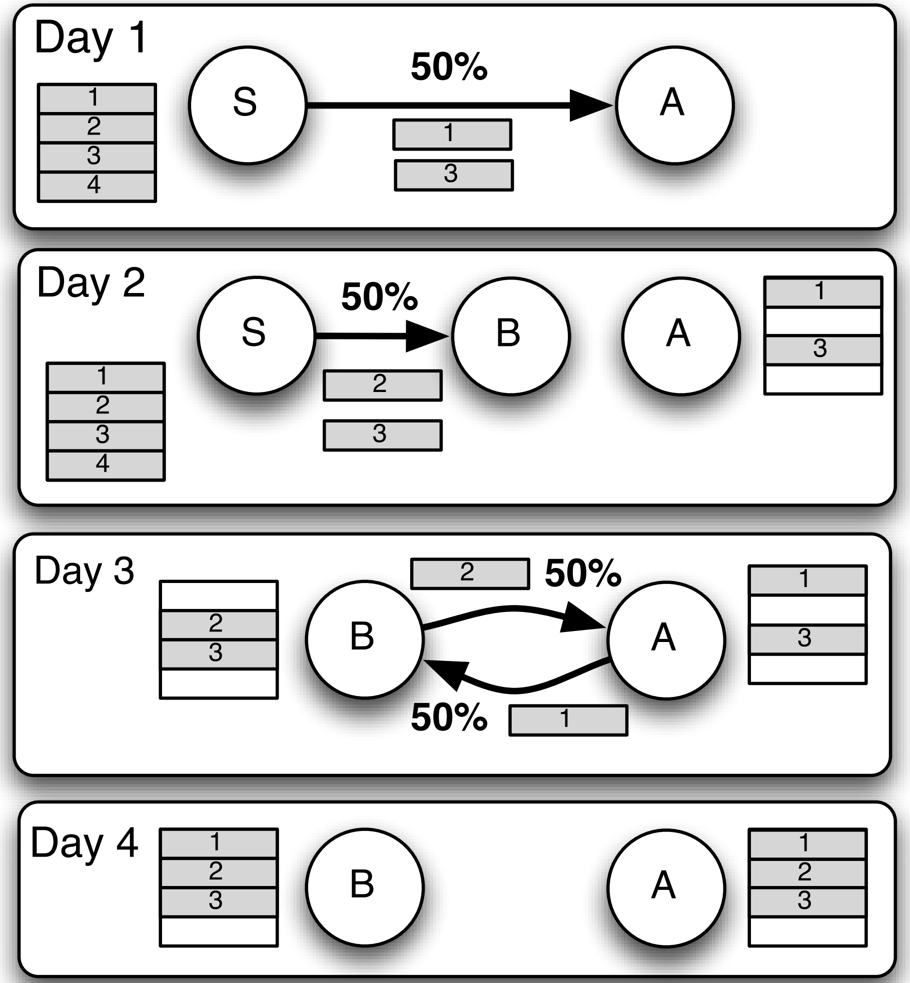
- **Overhead of storing of variables**
  - per block one variable vector
  - e.g. 4 GB file with 100 kB blocks
    - 4 GB/100 KB = 40 kB
    - Overhead of 40%
  - better: 4 GB und 1 MB-Block
    - 4kB Overhead = 0,4%
- **Overhead of Decoding**
  - Inversion of a m x m- Matrix needs time $O(m^3)$
- **Read/Write Accesses**
  - For writing m blocks each part must be read m times
  - Disk access is much slower than memory access

Pair-Coding

- **Paircoding: Improving File Sharing Using Sparse Network Codes Christian Ortolf Christian Schindelhauer Arne Vater**
- **Model Description**
  - Round model
    - complete information of the system can be described by file sharing state $\gamma(p,t)$ of each peer p after round t.
      - It is defined as the set of all code blocks that are available at peer p after round t.
  - Progress of a peer
    - number of indepdendent code blocks at a peer at round t
  - Availability at a set of peers
    - number of independent code blocks at the peers of the set divided by the number of code blocks

Scenario

- **Round model**
  - In each round each peer can upload and download a bounded number of blocks of the document
- **Peers do not know the future**
- **Progress**
  - number of (independent encoded) blocks that are available at the end of the rounds



Policy and Outperforming

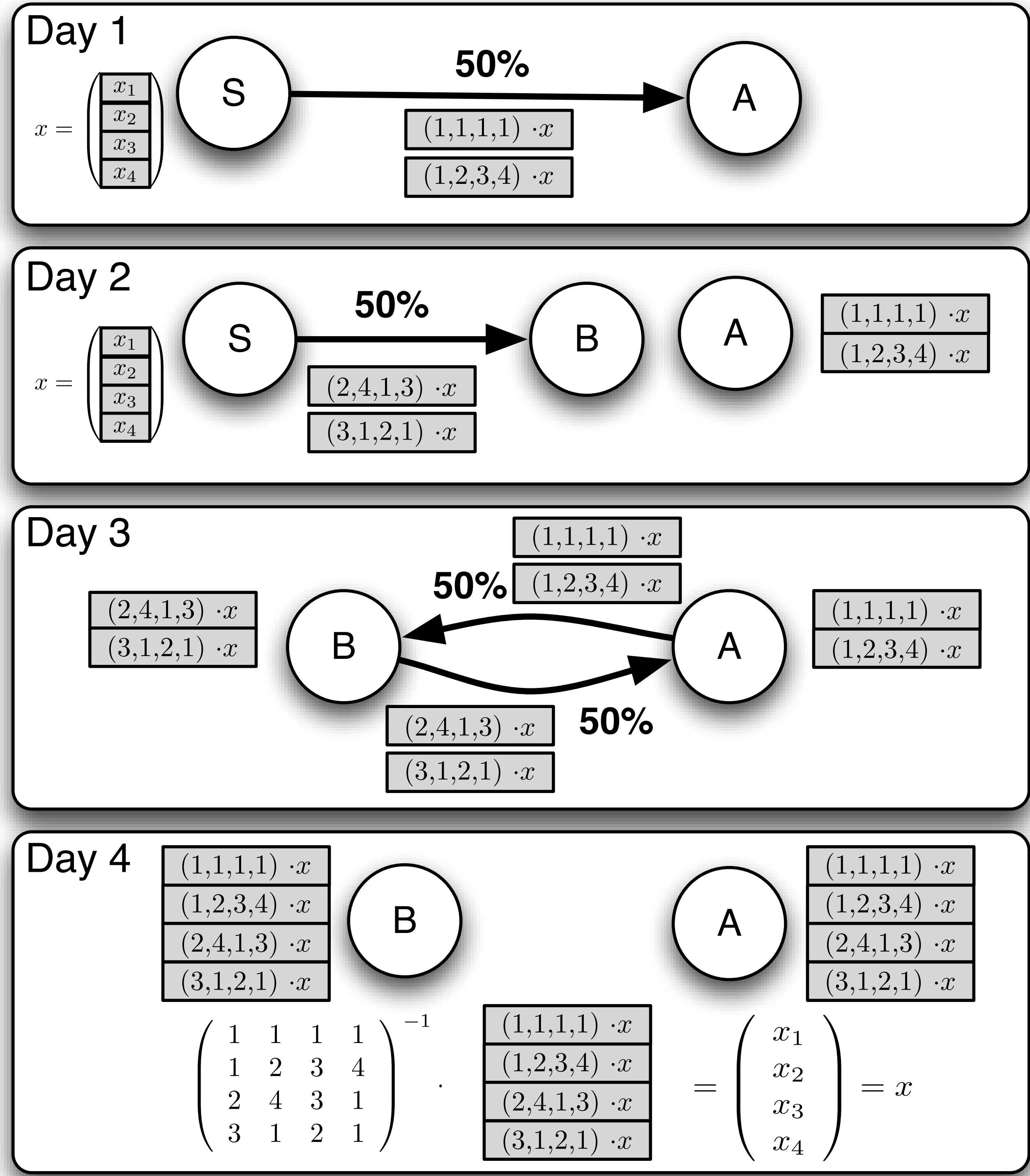
- **Policy of a scheme**
  - algorithmic choice of encoding of a block in a round
  - determine the efficiency of a scheme
- **Policies of Bittorrent**
  - chosen to optimize throughput and fairness
- **A scheme A is at least as good as B**

  $$\mathbf{A \geq B}$$

  - if for every scenario and every policy of B there is a policy in A such that A performs as well as B in all scenarios.

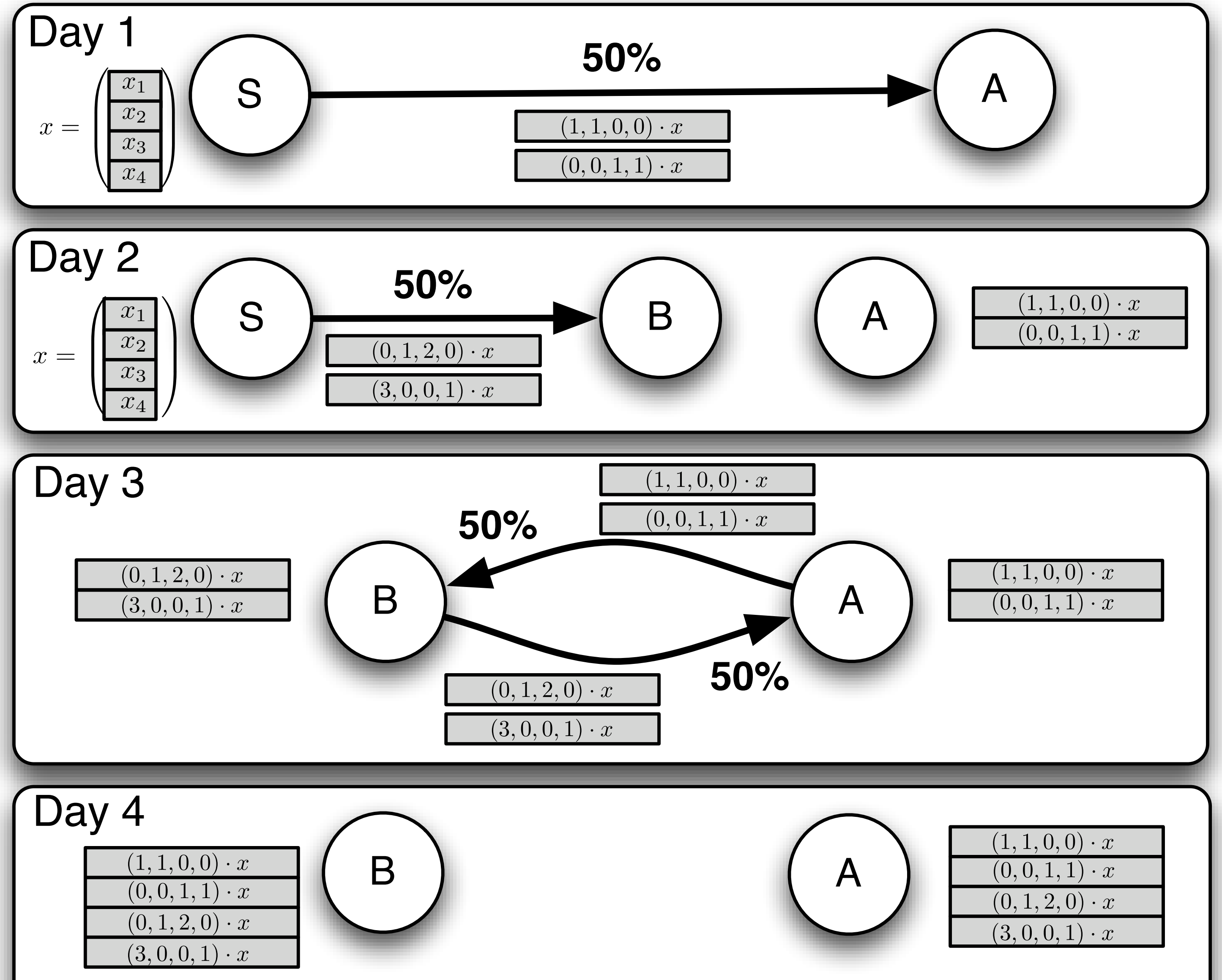
Network Coding

- **Practical Network Coding**
  - is the best possible method
  - as long as the underlying finite base is large enough
- **But:**
  - Decoding needs O(m) read/write operations



Pair Coding

- **Pair Coding**
  - is a reduced form of Network Coding
  - Only two components are combined
- **Theorem**
  - For all scenarios Pair-Coding is at least as efficient as Bittorrent
  - For some scenarios Pair-Coding is more efficient than Bittorrent
  - Encoding and Decoding can be performed with (almost) linear number of Read/Write-Operations



$$\begin{pmatrix}1&1&0&0\\0&0&1&1\\0&1&2&0\\3&0&0&1\end{pmatrix}^{-1}\cdot\begin{pmatrix}(1,1,0,0)\cdot x\\(0,0,1,1)\cdot x\\(0,1,2,0)\cdot x\\(3,0,0,1)\cdot x\end{pmatrix}=\begin{pmatrix}x_1\\x_2\\x_3\\x_4\end{pmatrix}=x$$

The Random Policy

- **Scenario**
  - one seeder
  - one downloading peer
- **Seeder sends a random block in each round**

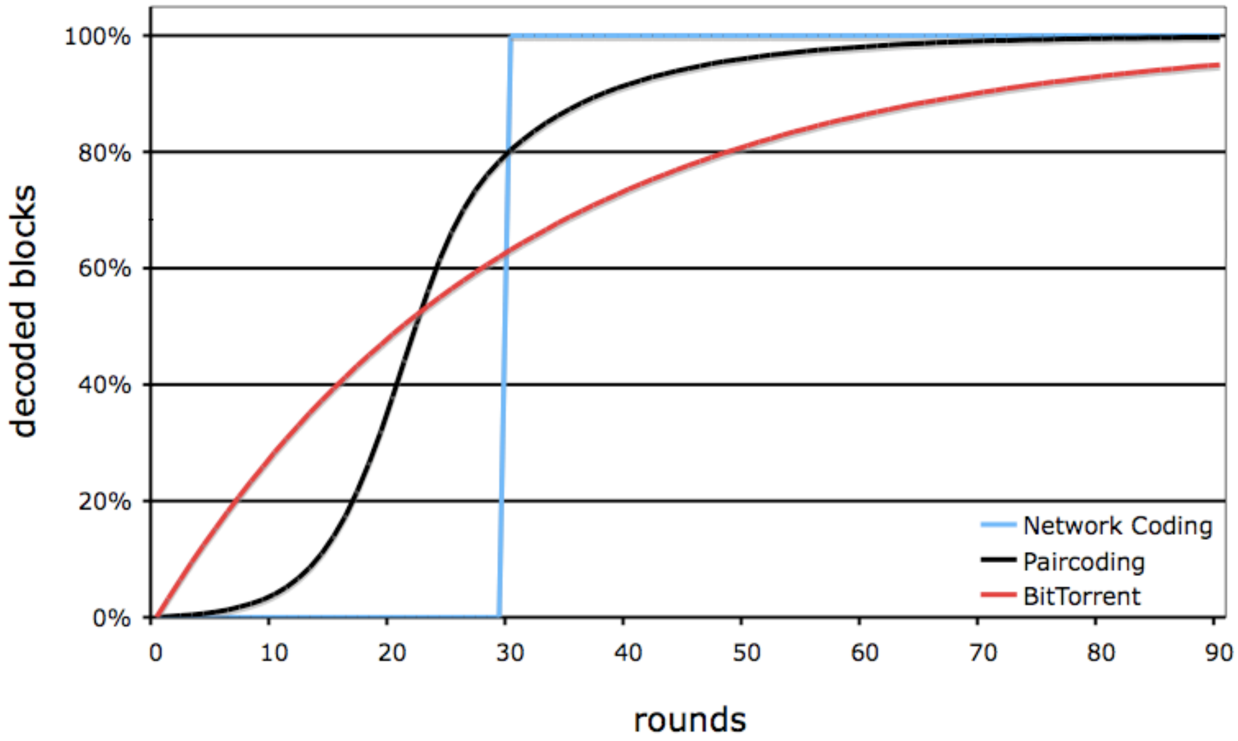


Figure 8. Simulation of decodability for one peer

Availability

- **Scenario:**
  - p peers
  - one seeder
  - every peer receives n/p+1 blocks from the seed
  - then the seed disappears

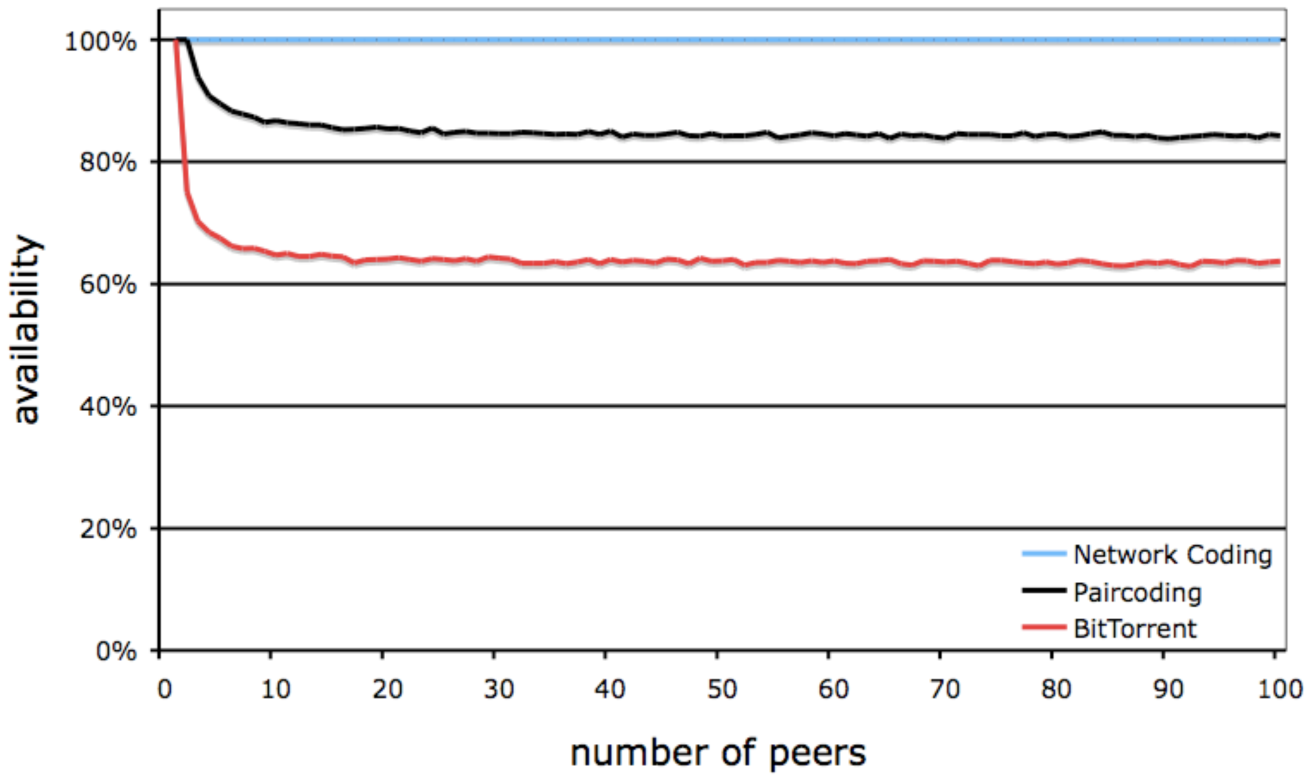
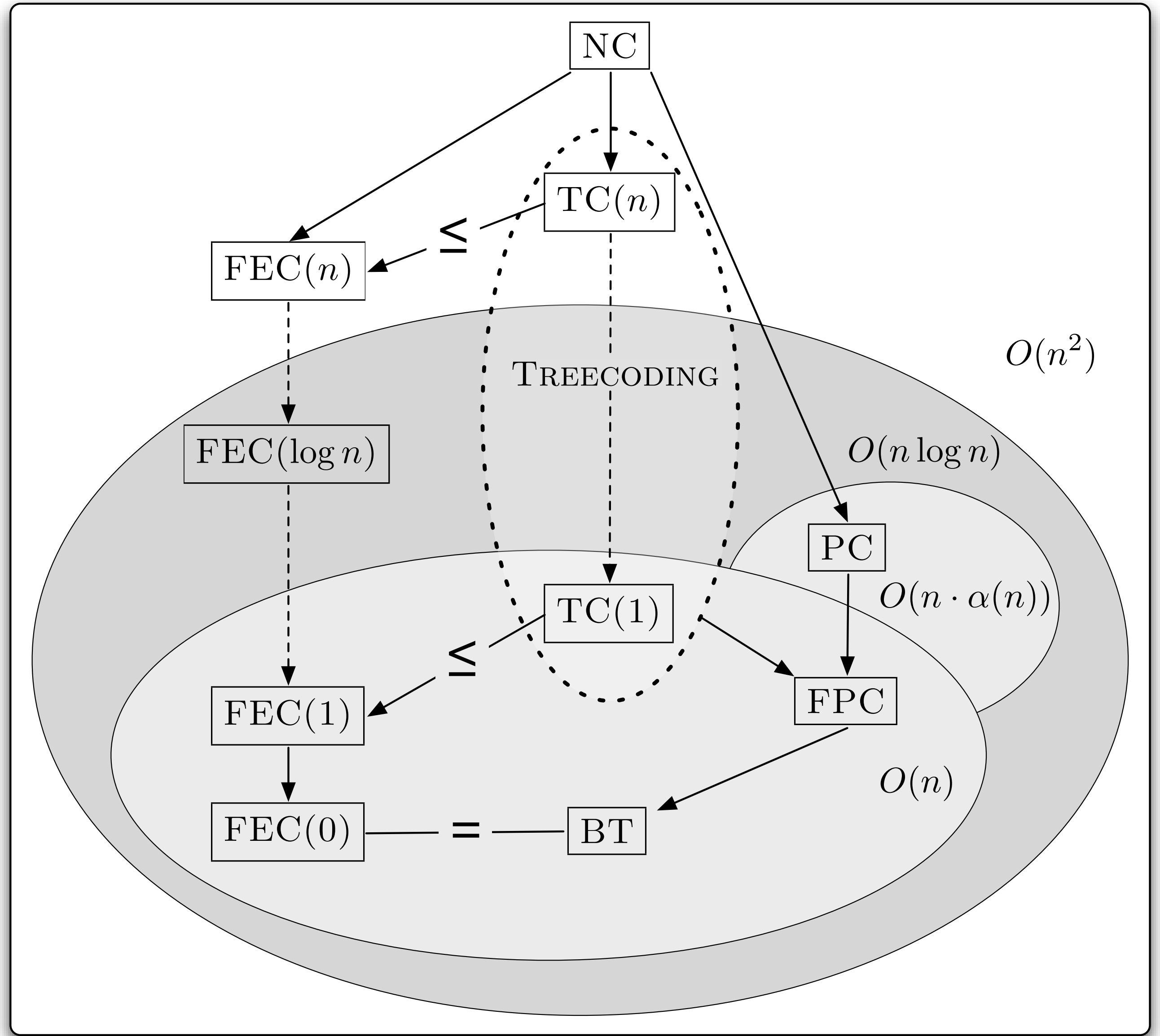


Figure 9. Simulation of availabilty for increasing number of peers

Results So Far



under submission