Peer-to-Peer Networks

12 Fast Download

Arne Vater

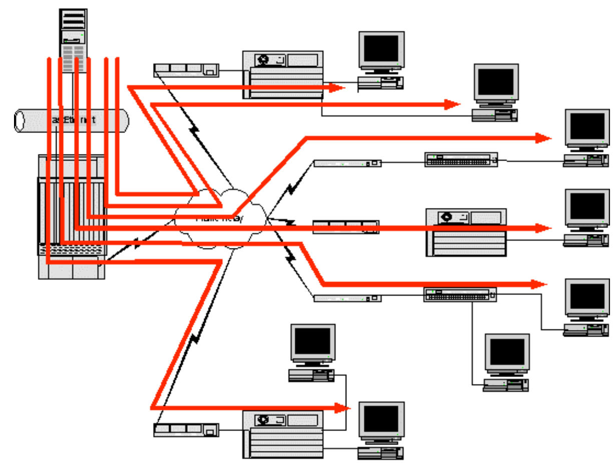
Technical Faculty

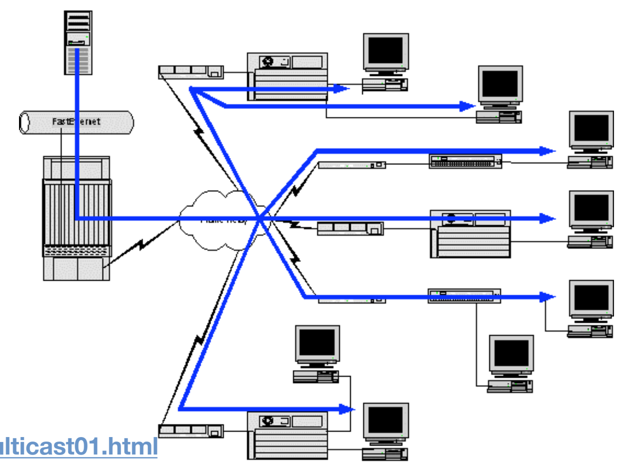
Computer Networks and Telematics

University of Freiburg

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





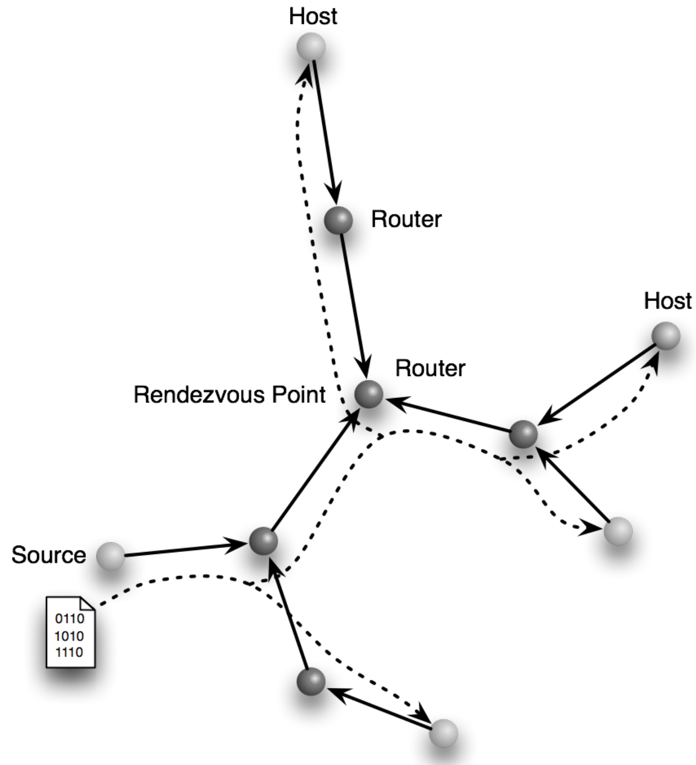
Peter J. Welcher
www.netcraftsmen.net/.../ papers/multicast01.html

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

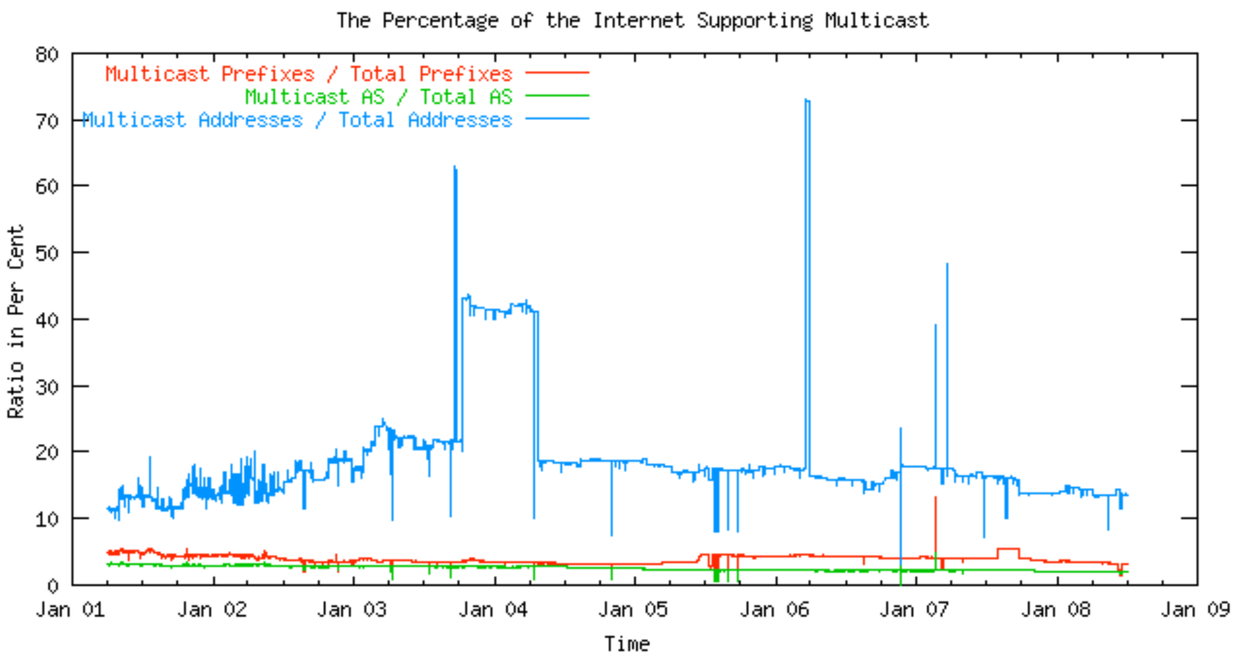
Routing Protocols

- 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



IP Multicast Seldomly Available

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

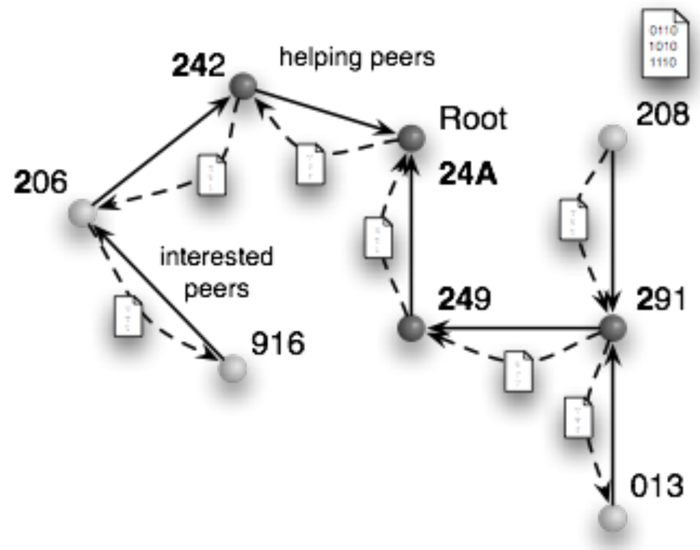


Why so few Multicast Routers?

- 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

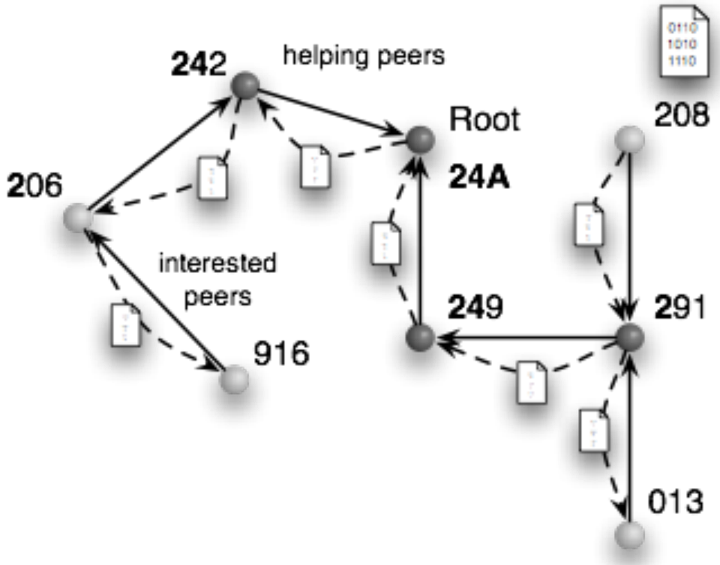
Scribe & Friends

- 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
- Other
  - Overcast [´00] and Narada [´00]
  - construct multi-cast trees using unicast connections
  - do not scale



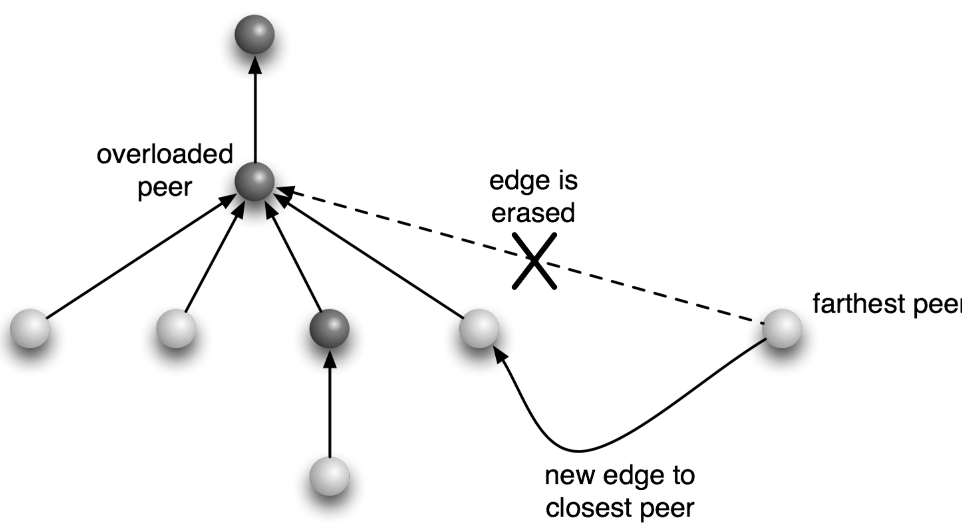
How Scribe Works

- 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



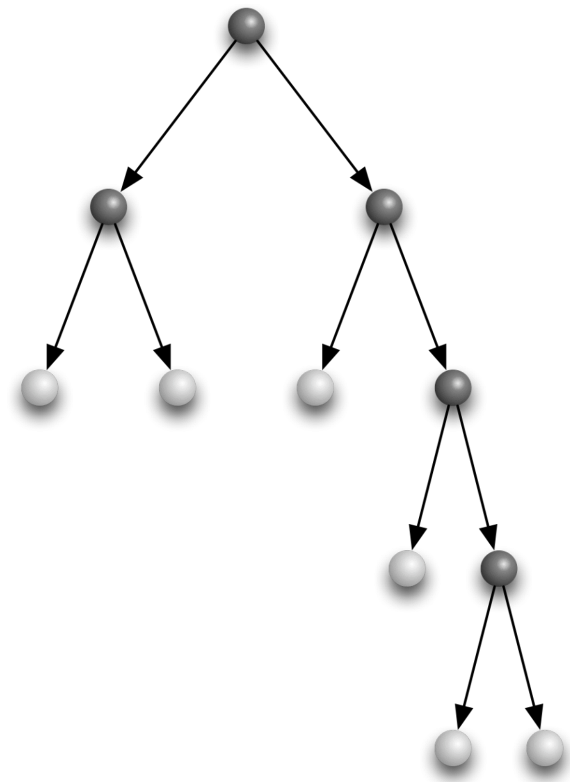
Scribe Optimization

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



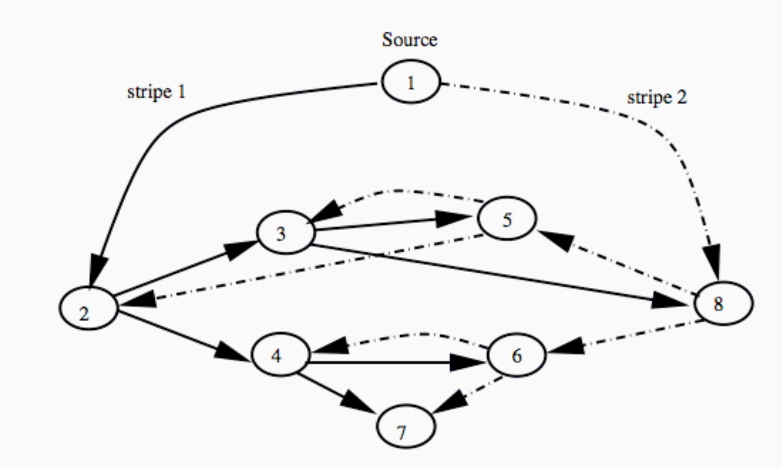
Split-Stream: Motivation

- 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
    - $(d-1) \cdot k = b - 1$
- 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 leaf 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 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
    - additional hash-code of the file contents and other control information
  - tracker hosts information about peers
    - does not store files
    - yet, providing a tracker file on a tracker host can have legal consequences
- File
  - is partitioned into smaller pieces
    - as described in tracker file
  - every participating peer can redistribute downloaded parts as soon as received
  - 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 an 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 join shortly only
  - 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

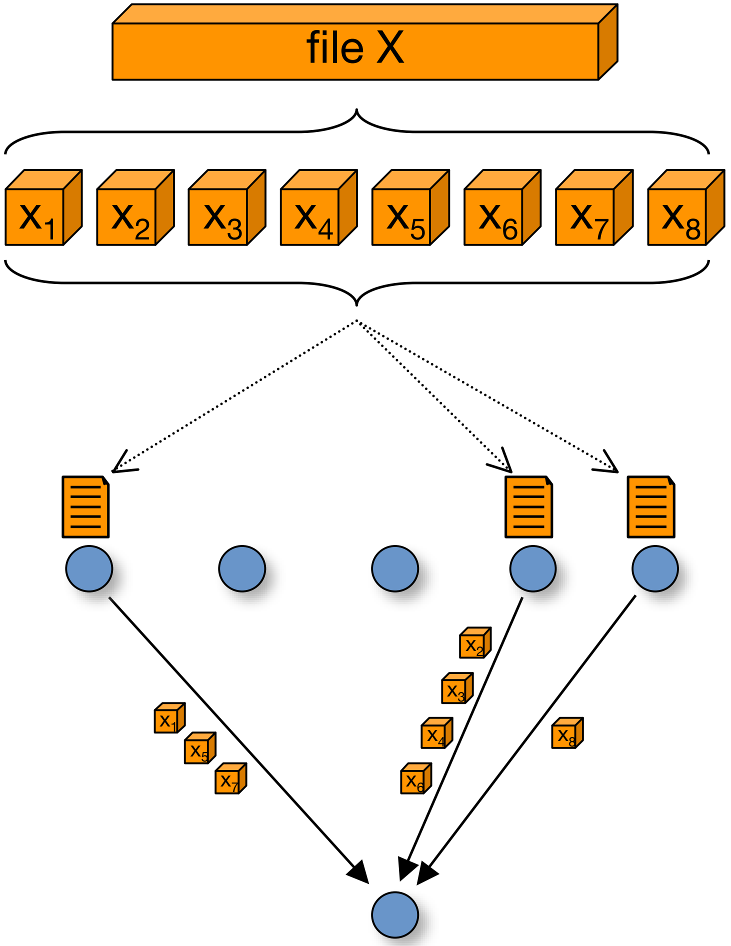
- Goal
  - self organizing system
  - good (uploading, seeding) peers are rewarded
  - bad (downloading, leeching) peers are penalized
- Reward
  - good download speed
  - unchoking
- Penalty
  - Choking of the bandwidth
- Evaluation
  - Every peer evaluates its environment by its past experiences

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
    - prevents maltreating a peer with a bad bandwidth

Alleviating the Coupon Collector

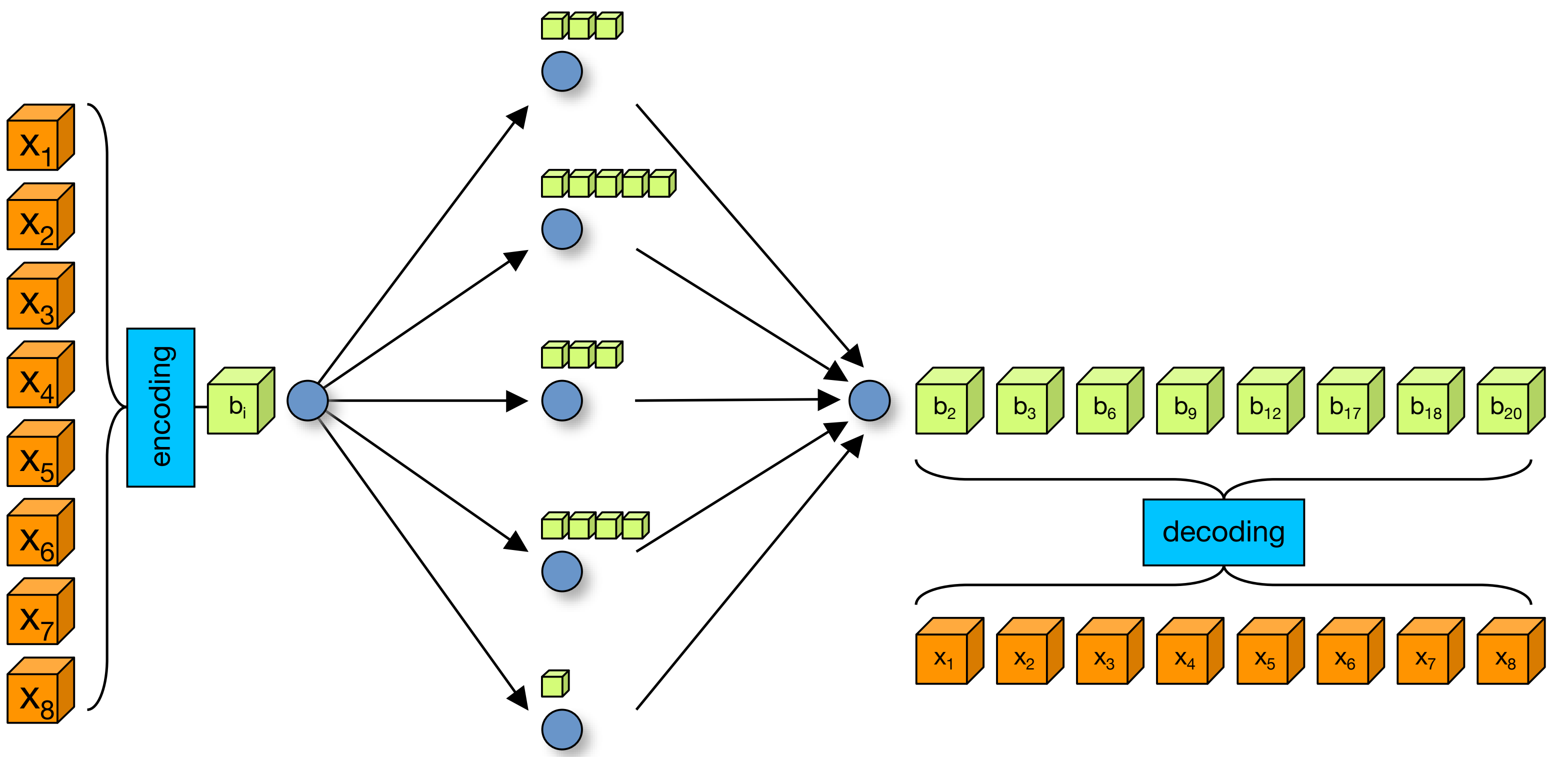
- Each peer needs one copy of all *n* blocks
  - regardless from whom
- Single blocks can get lost from the network
  - e.g. when the seed leaves
  - no download can succeed
- Network Coding can solve this problem



Practical Network Coding

- Gkantsidis, and Rodriguez
  - *"Network coding for large scale content distribution“* [2005]
- Method
  - sender transmits code blocks as linear combinations of the file‘s blocks
  - receiver collects code blocks and reconstructs the original file

Example



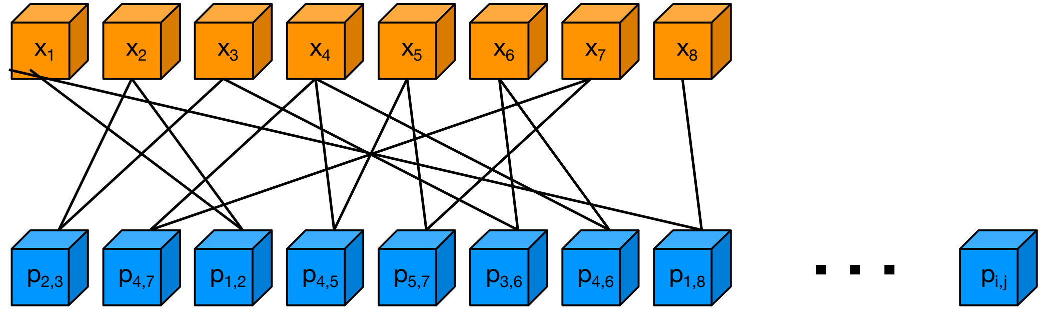
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
encoding
b_i
b_2
b_3
b_6
b_9
b_{12}
b_{17}
b_{18}
b_{20}
decoding
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8

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 $(n \times n)$-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

Paircoding

- Paircoding: Improving File Sharing Using Sparse Network Codes [ICIW 2009]
  - is a reduced form of Network Coding
  - combines only two original blocks into one code block
    - $p_{i,j} = c_i x_i + c_j y_j$



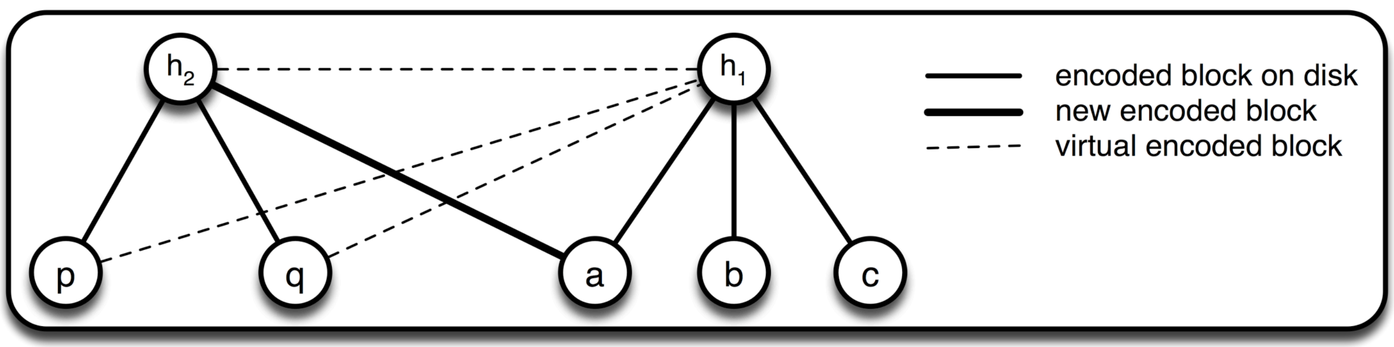
Decoding

- Connected block component
  - code blocks $p_{i,j}$ and $p_{m,n}$ are connected, if
    - $i \in \{m,n\}$ or
    - $j \in \{m,n\}$
  - all connected code blocks are recoded to
    - $p_{h,j}$ or
    - $p_{i,h}$
  - h head block
  - can be merged if i and j are in two different connected block components

Example

Decoding

- Recoding is delayed until block is read
  - “lazy“
- Decoding a component is fast by decoding head first



R/W Complexity

- Read/write cost
  - number of blocks to read from or write to disk
    - for coding
    - and decoding

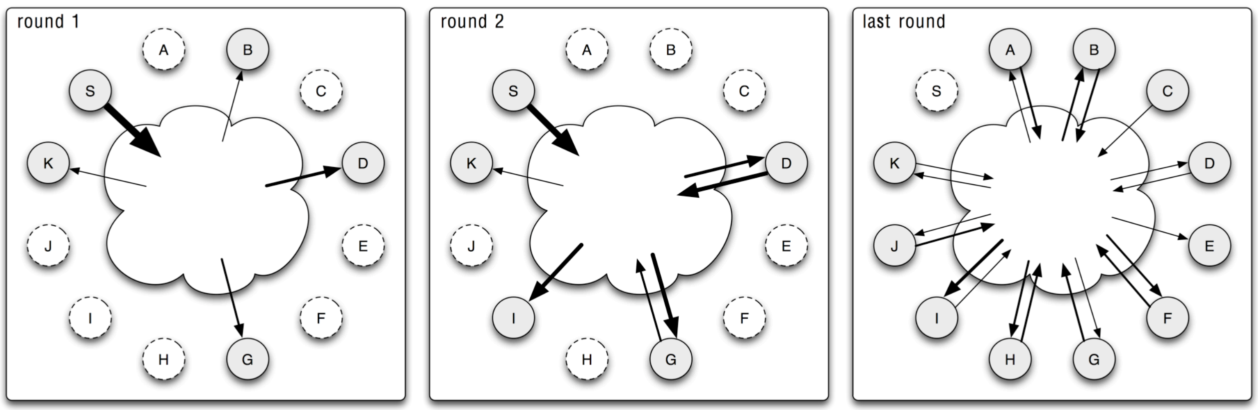
BitTorrent	Paircoding	Network Coding
O(n)	$O(n \cdot \alpha(n))$	$O(n^2)$

$\alpha(n)$ is the inverse Ackerman function

CoNe Freiburg

Round Model

- Network configuration
  - download & upload limits of each peer



Model

- progress of a peer
  - number of linearly independent code blocks divided by *n*
- availability at a set of peers
  - number of linearly independent code blocks at all peers of the set divided by *n*
- peers do not know the future

Outperforming

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

A ≥ B

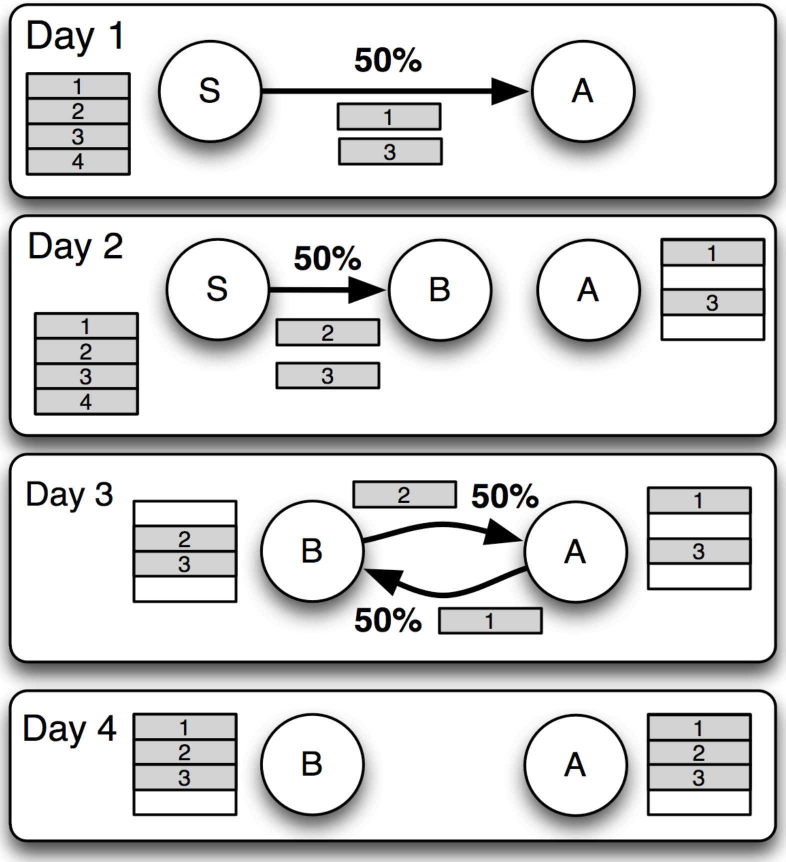
if for every scenario and every policy of B there is a policy in A such that A performs at least as well as B.

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

A > B

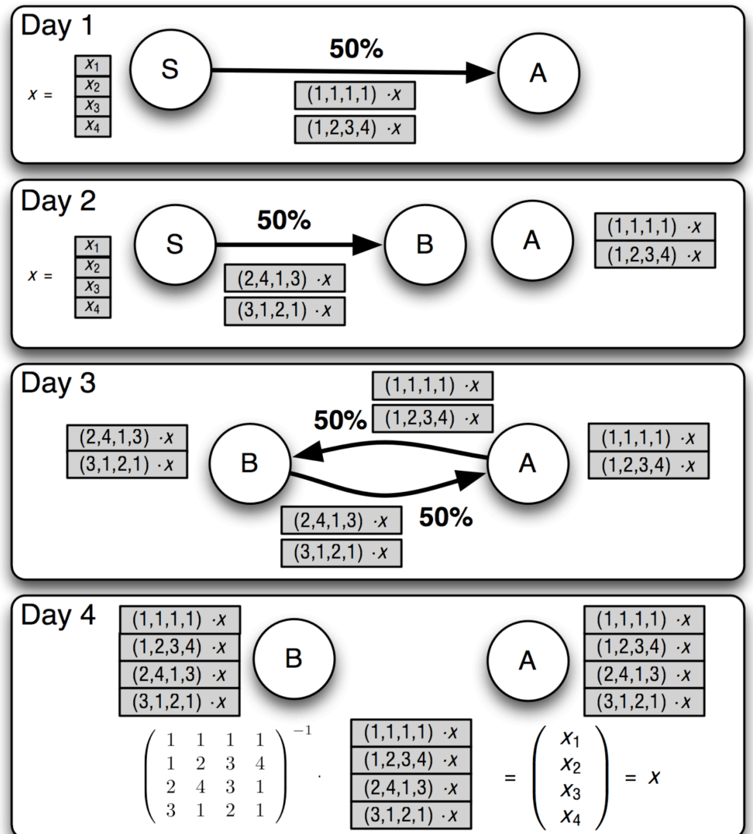
Analysis BitTorrent

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



Analysis Network Coding

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

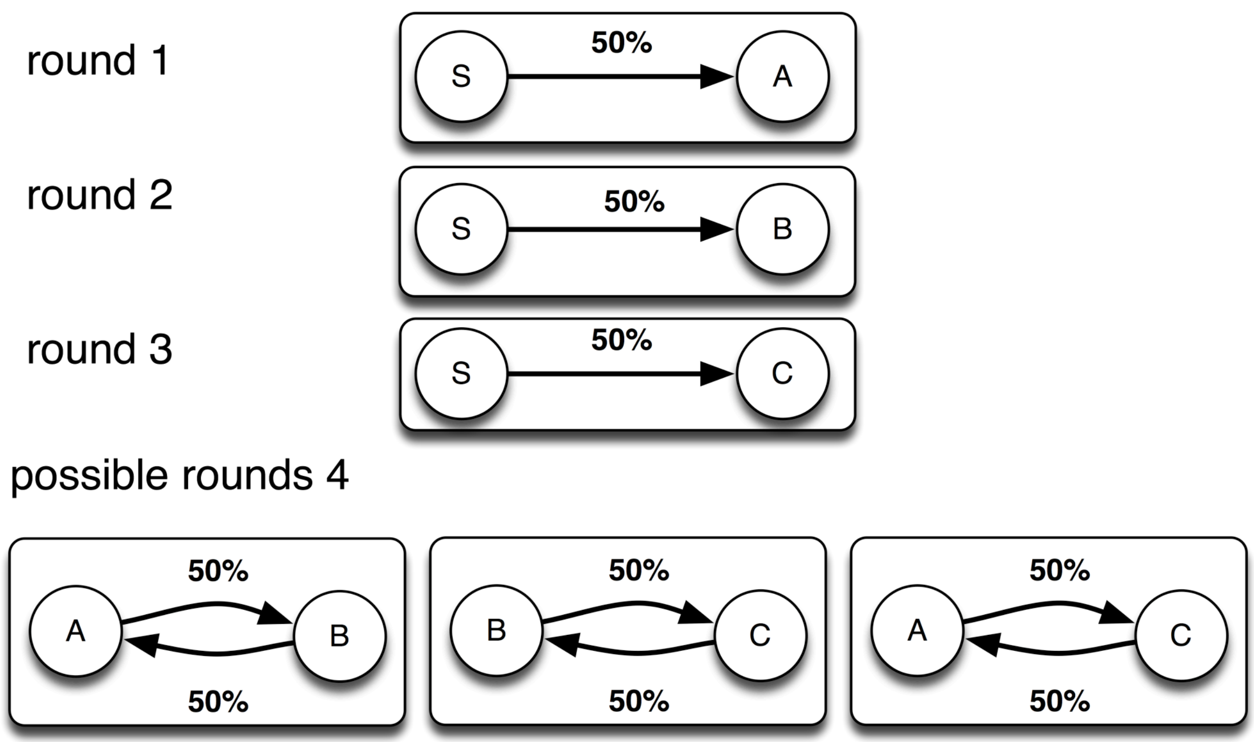


Analysis Paircoding

- Paircoding performs at least as good as BitTorrent
  - when BitTorrent sends block x_i
  - Paircoding sends code block $p(x_i, x_{n-i})$

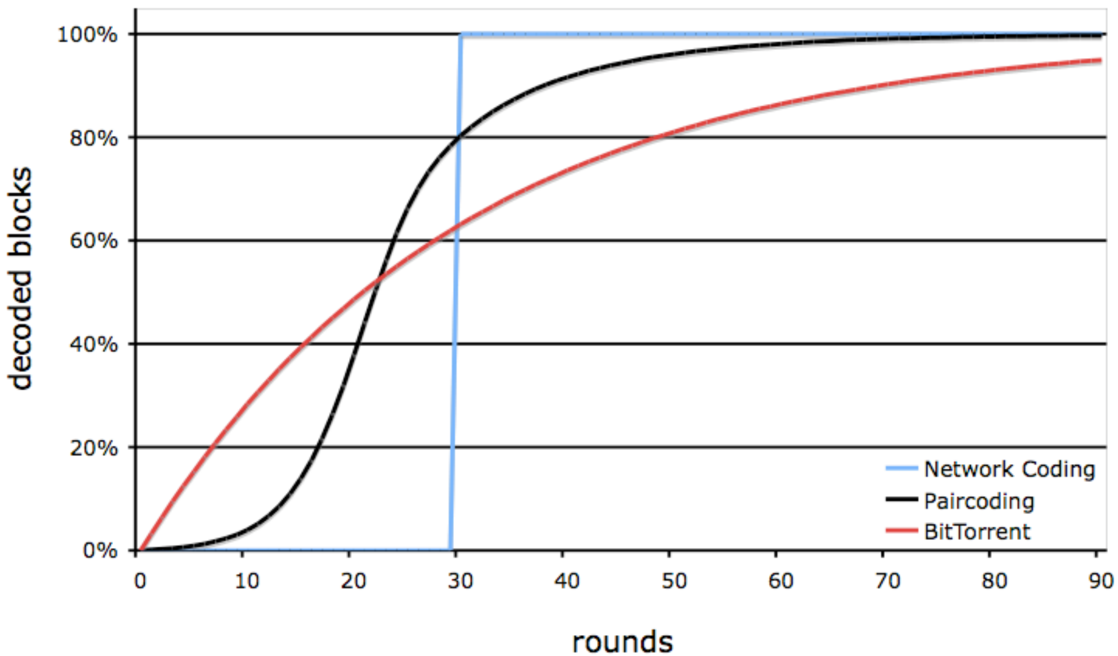
Analysis Paircoding

- Paircoding outperforms BitTorrent



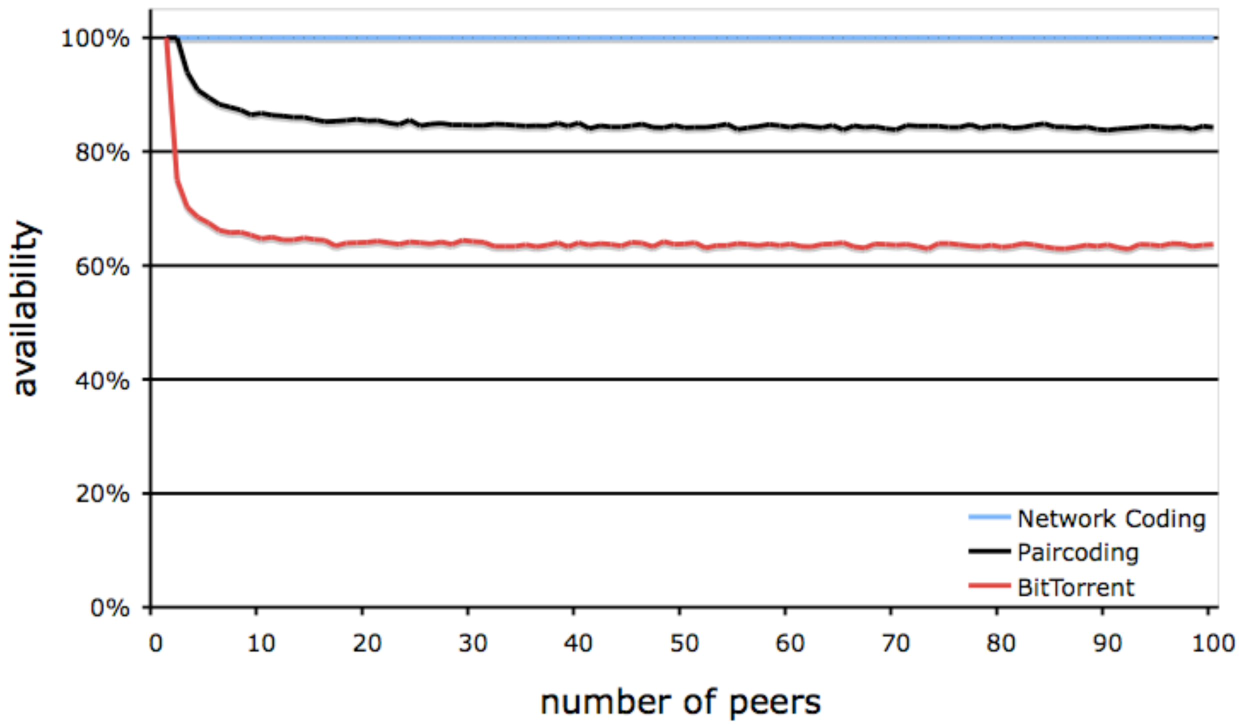
Simulation

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



Simulation

- 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



Peer-to-Peer Networks

12 Fast Download

Arne Vater
Technical Faculty
Computer Networks and Telematics
University of Freiburg