CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

Peer-to-Peer Networks

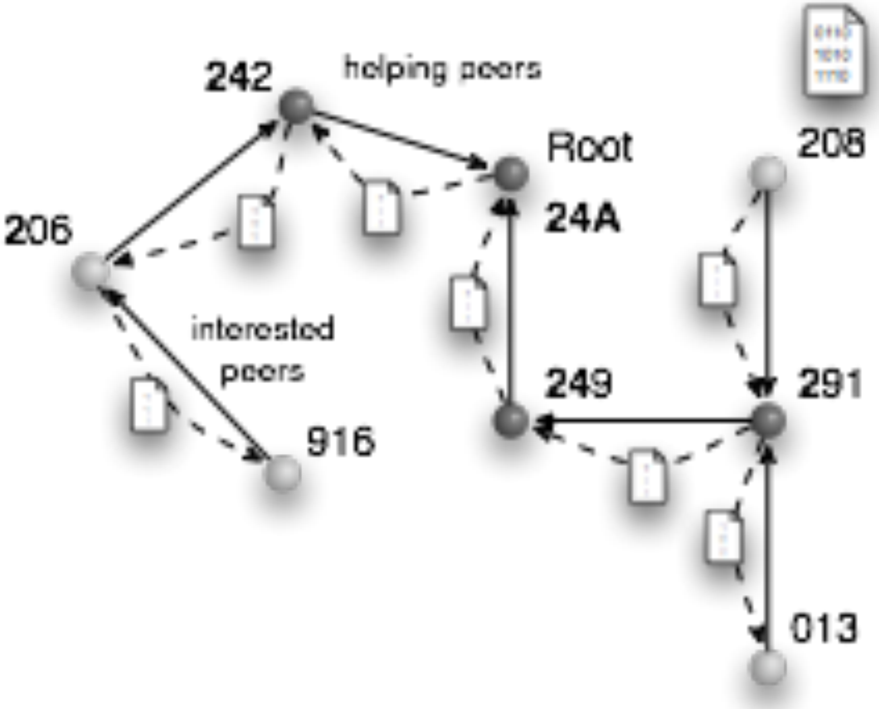
10 Fast Download

Christian Schindelhauer
Technical Faculty
Computer-Networks and Telematics
University of Freiburg

UNI FREIBURG

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



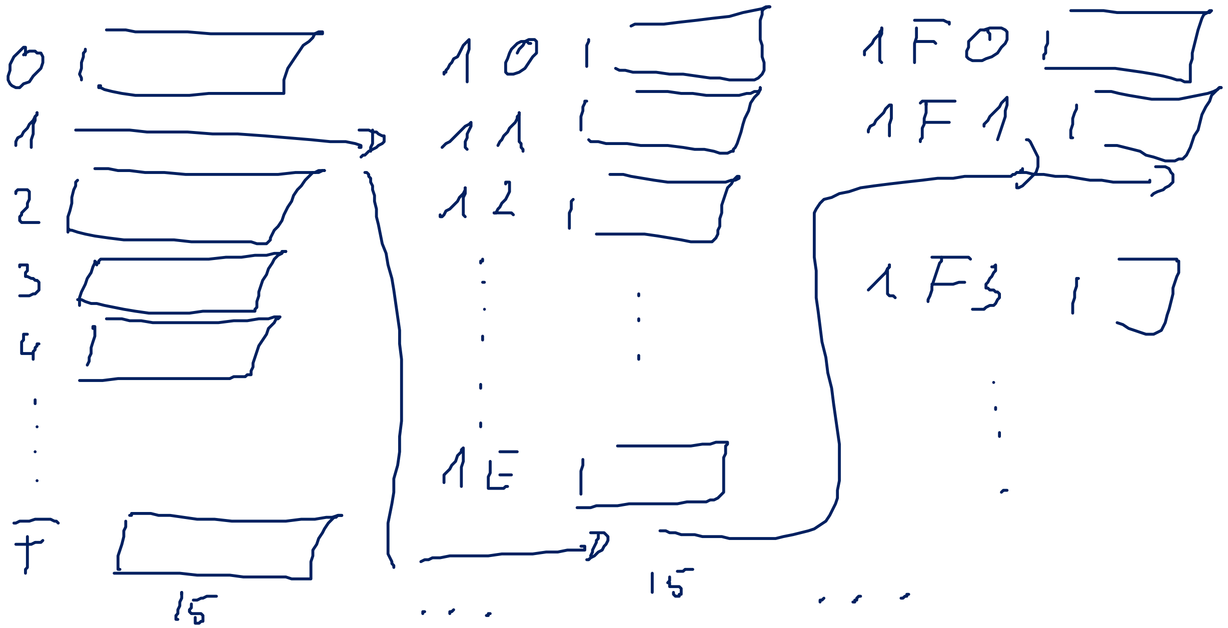
Pastry

-> Zip-code thing

Plaxton Routing

Address 1F2A..

1st Link



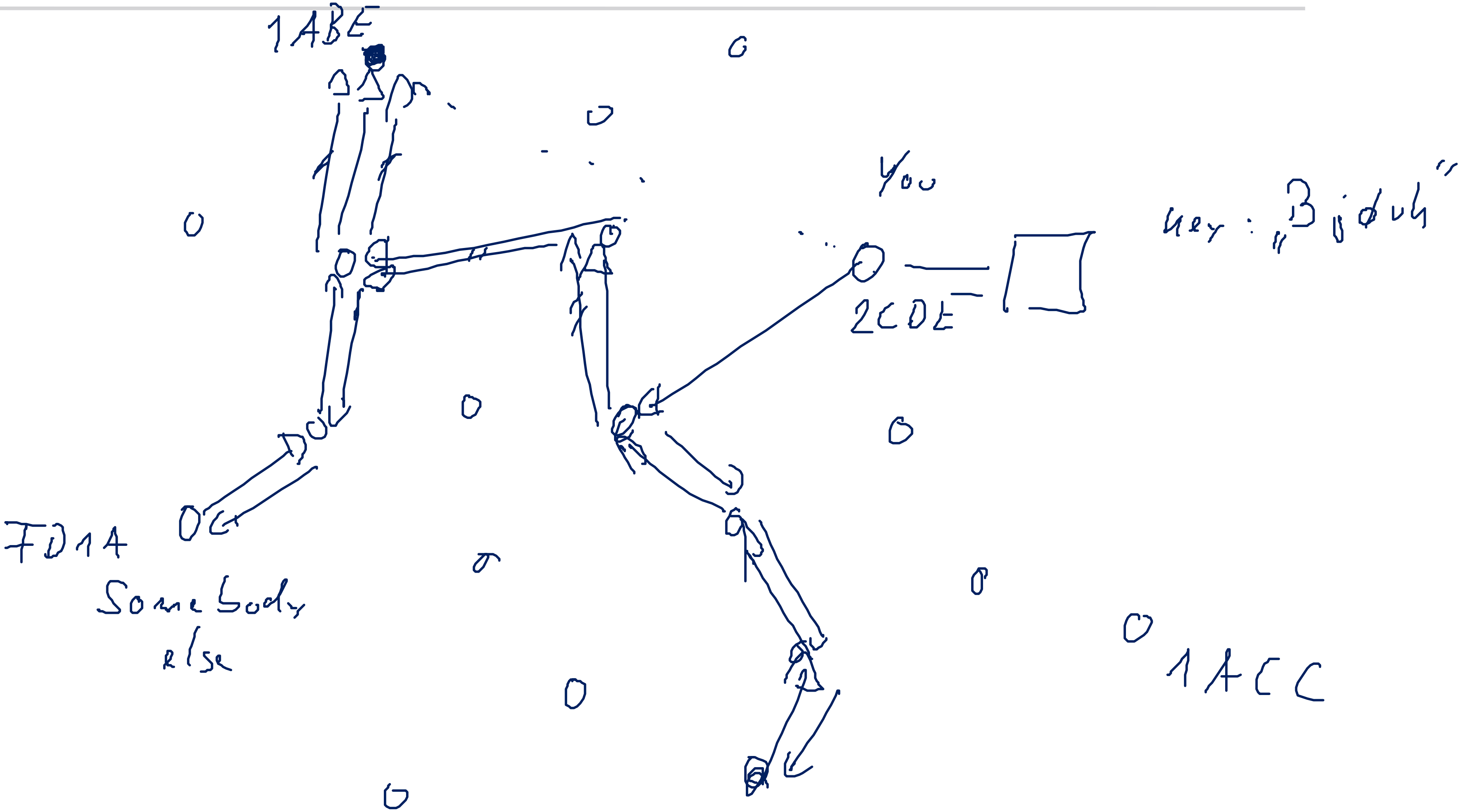
2nd Link

3rd Links

Magic

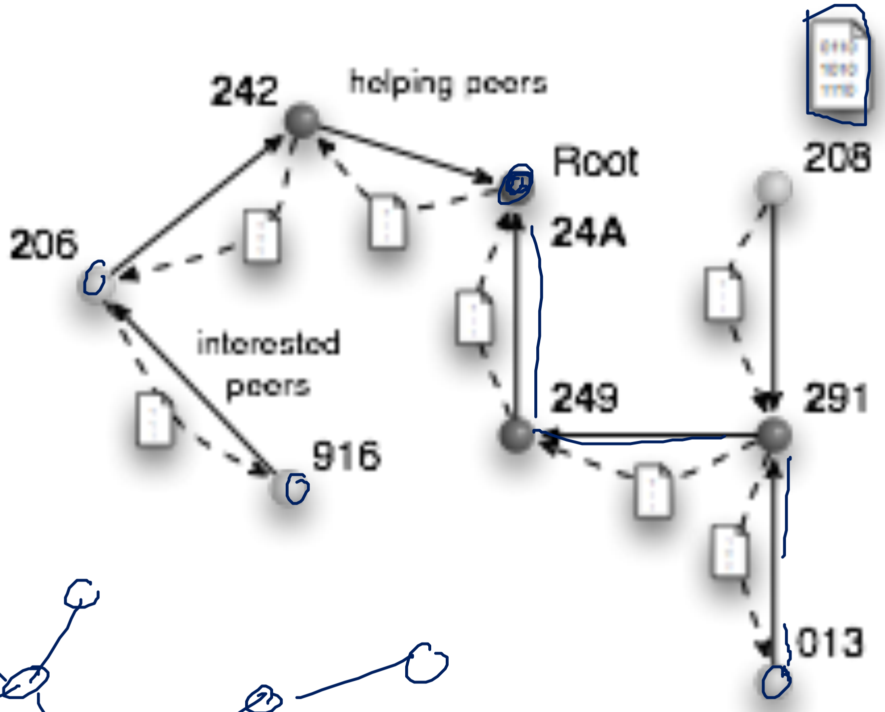
DHT

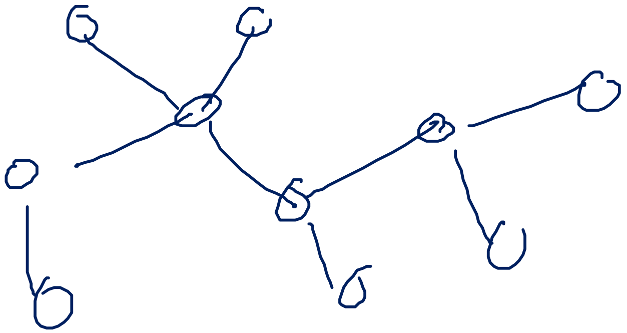
closest to h(„3jdvk") = 1ABC



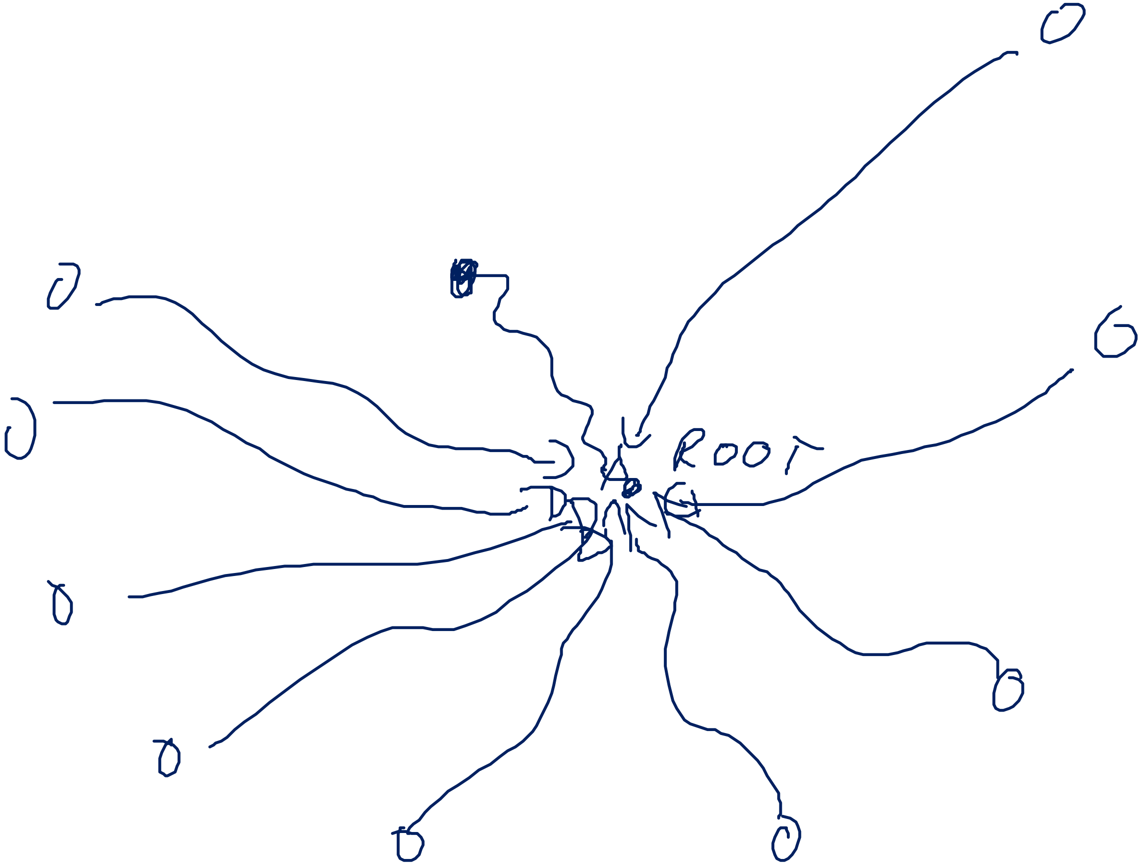
How Scribe Works

- **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



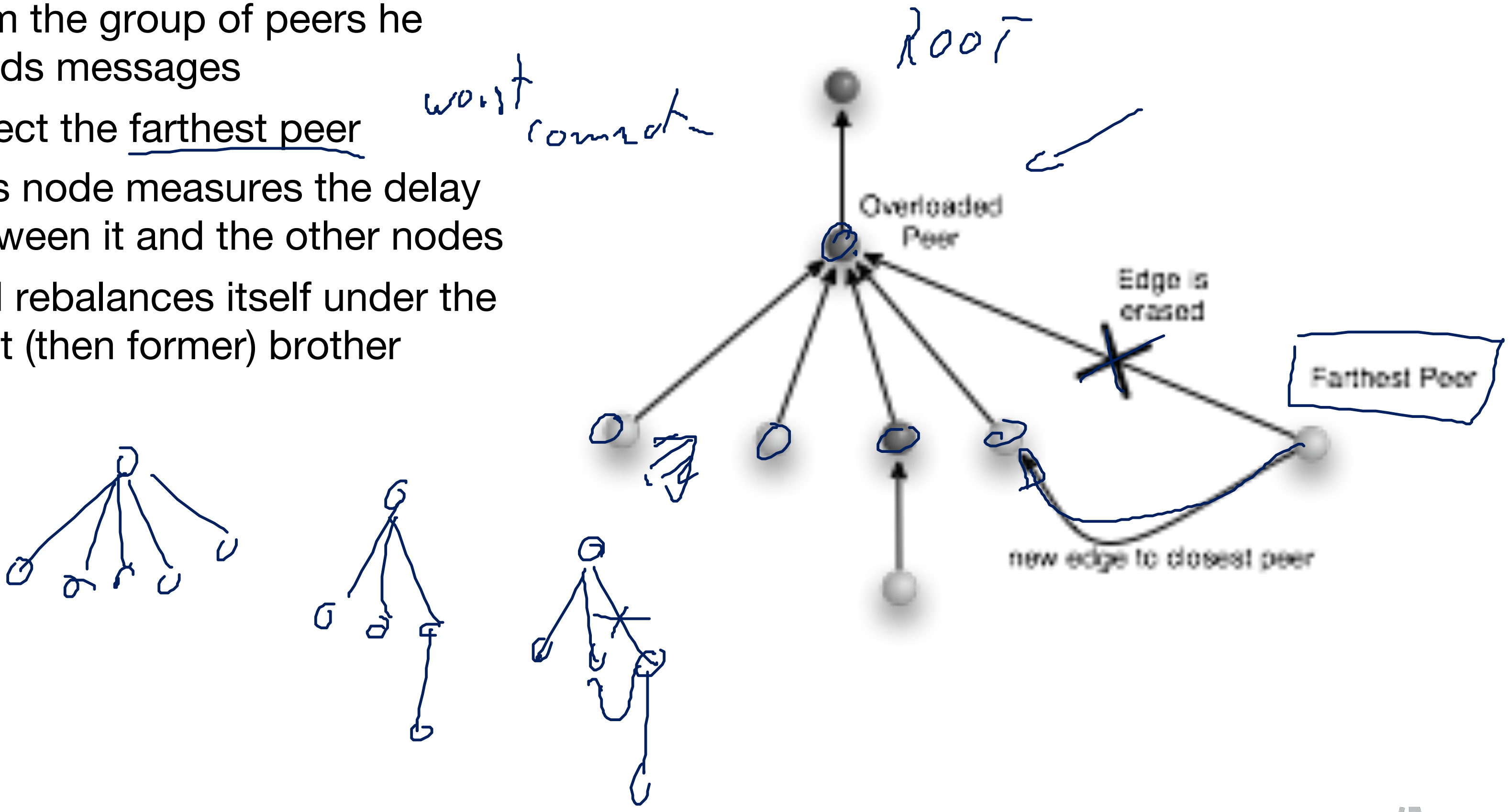


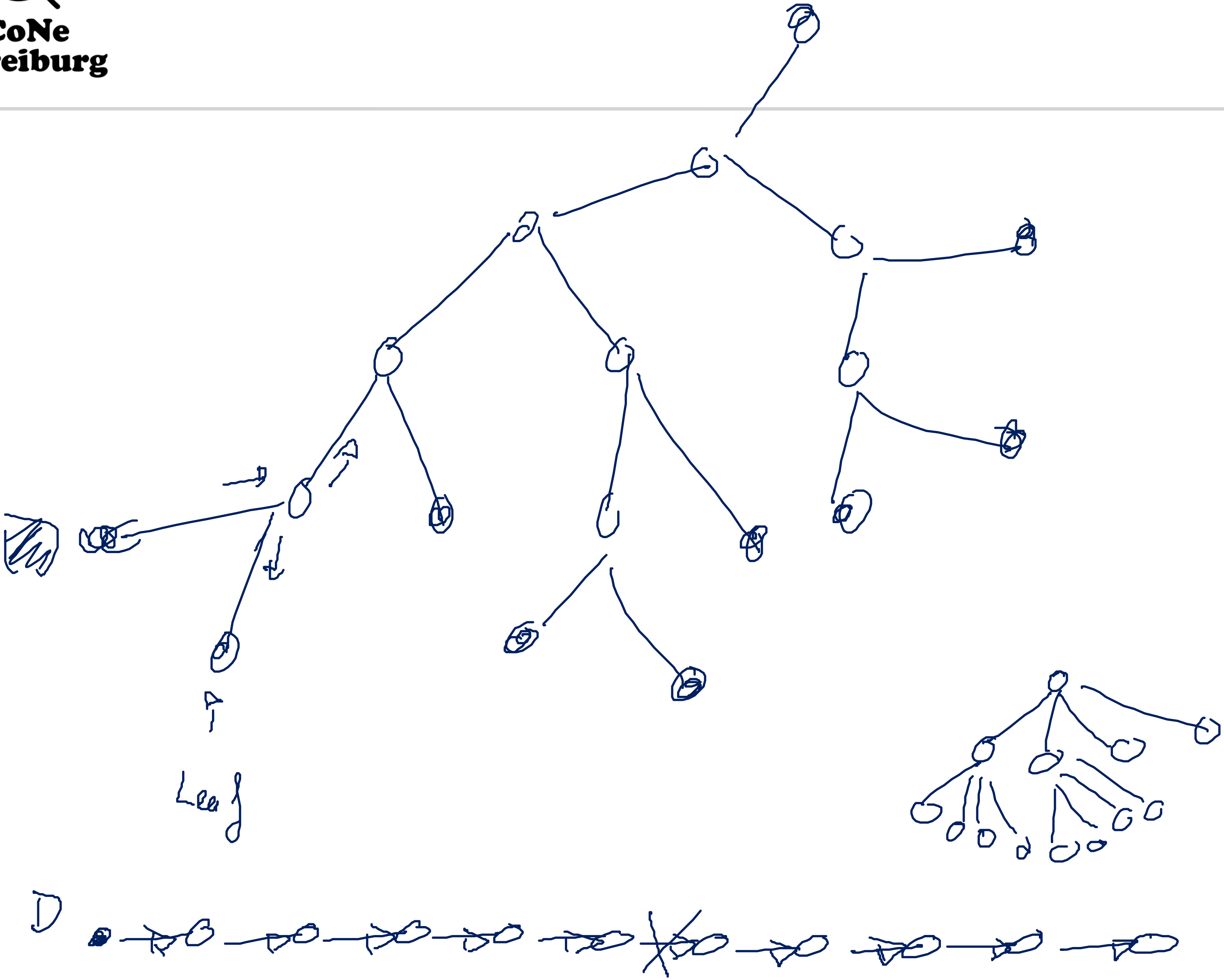
$$\underline{15} \cdot \underline{\frac{\log n}{\log 16}}$$



Scribe Optimization

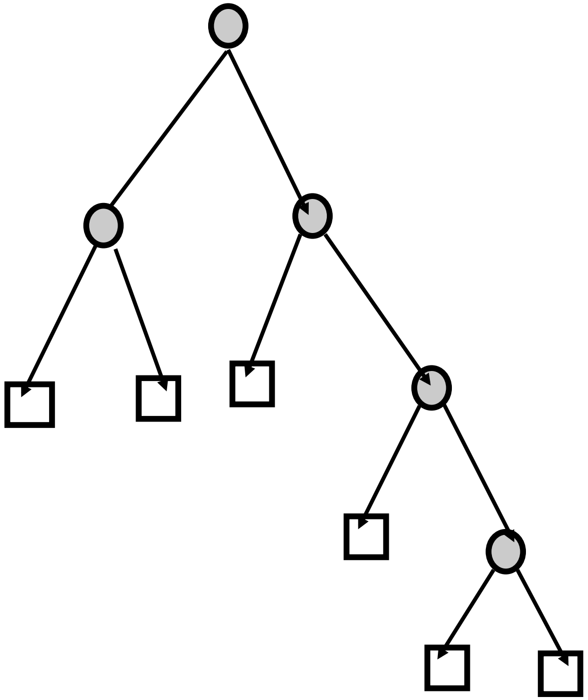
- **Bottleneck-Remover**
  - If a node is overloaded then from the group of peers he sends messages
  - Select the farthest peer
  - This node measures the delay between it and the other nodes
  - and rebalances itself under the next (then former) brother

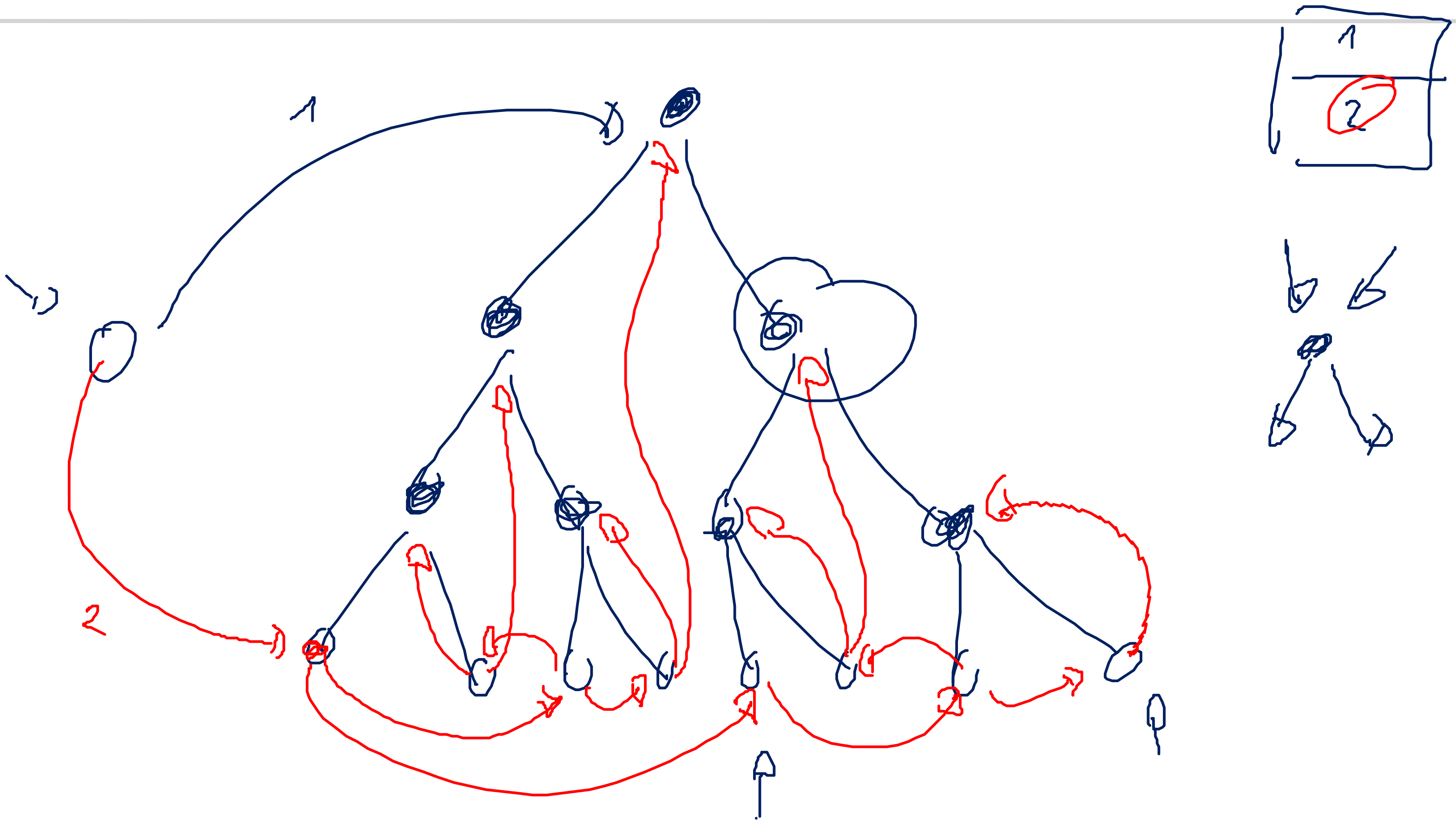



Leaf

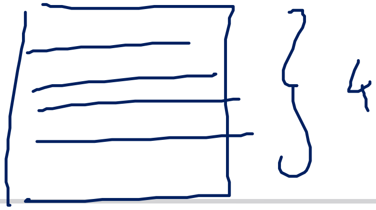
Split-Stream
Motivation

- **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

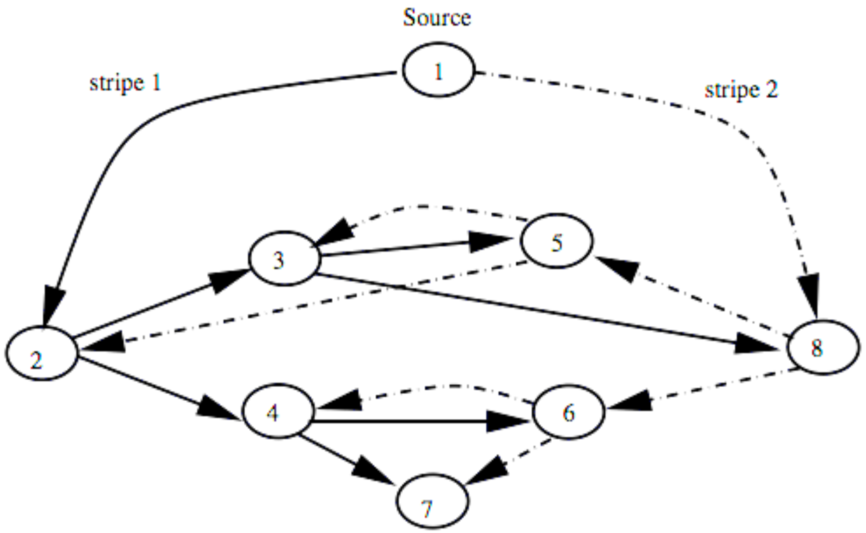



1
2
1
2

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 a most the download**

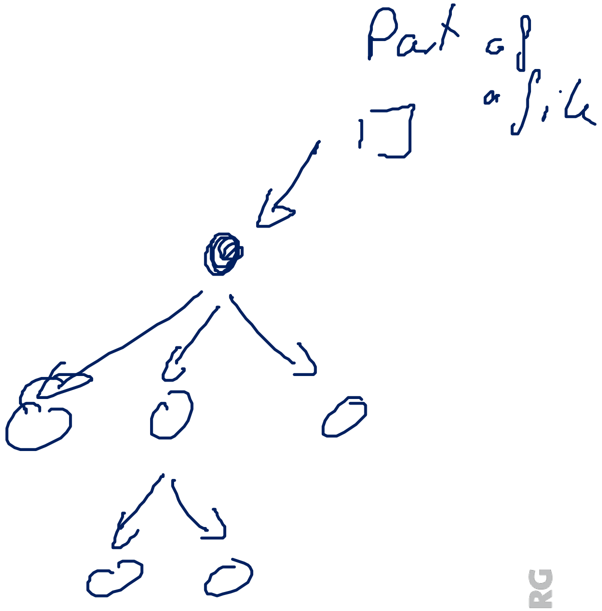


– churn

– coordination

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

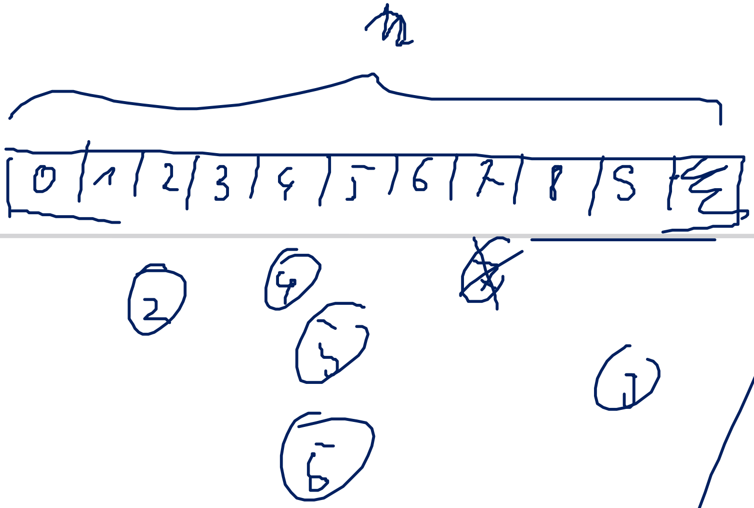
P2P Network

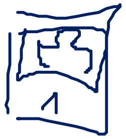
Kademlia ≈ Chord

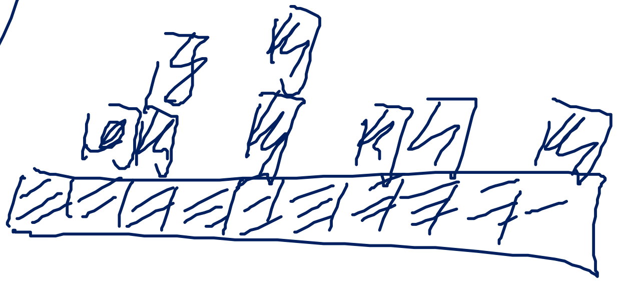
- **Central coordination (original implementation)**
  - by tracker host
  - for each file the tracker outputs a set of random peers from the set of participating peers
    - in addition hash-code of the file contents and other control information
  - tracker hosts to not store files
    - yet, providing 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







Coupon-Collector

$$\rightarrow \quad \Theta(n \cdot \underline{\log n})$$

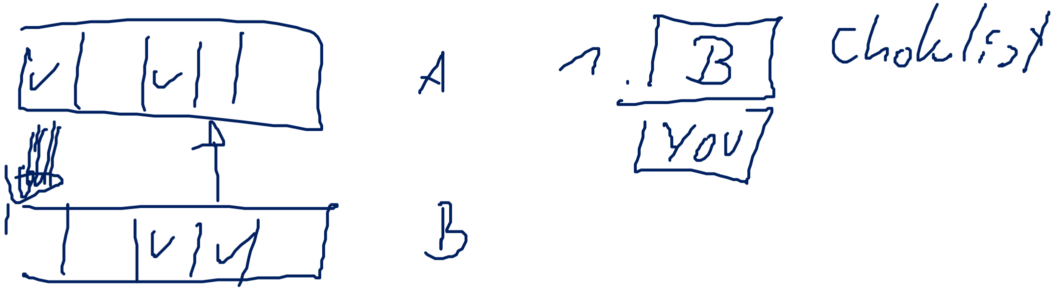
Bittorrent
Policy

- **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

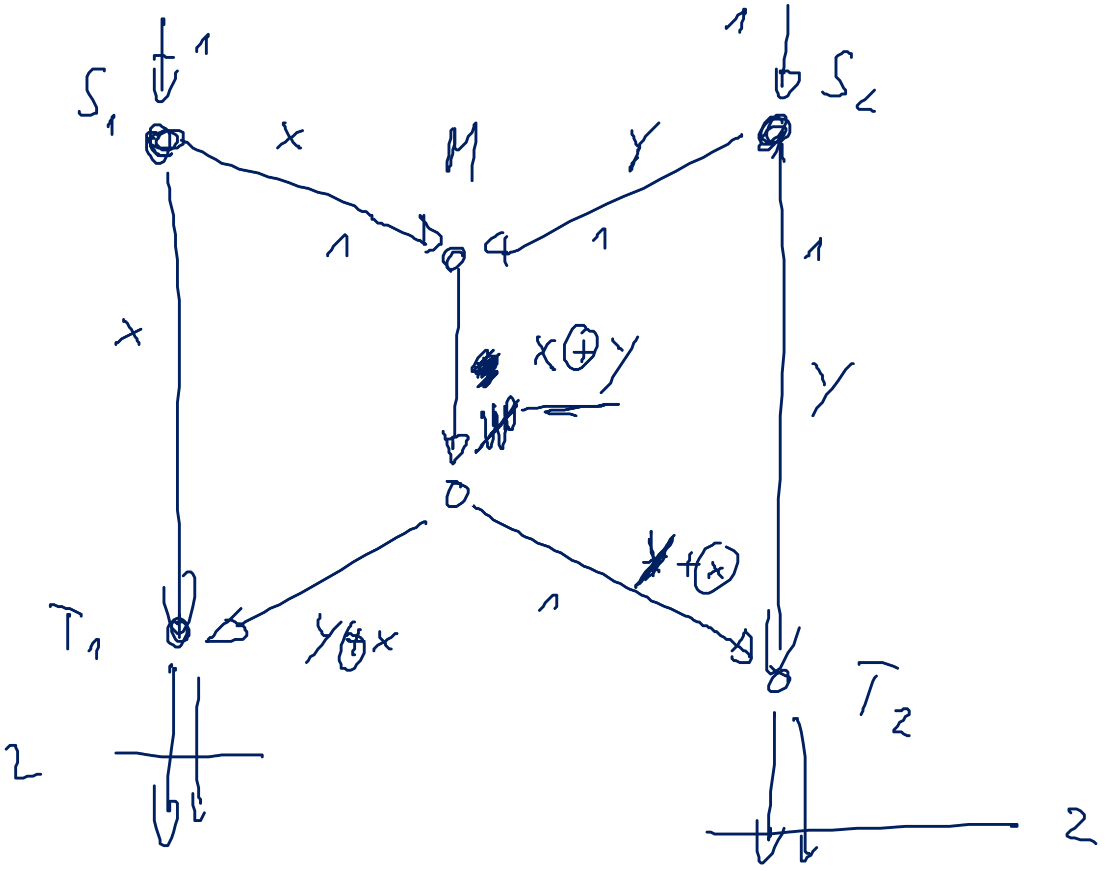
Bittorrent Choking

→ Game Theory

- **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



$x \in \{0,1\}$ $\quad\quad$ $y \in \{0,1\}$



xor	0	1
0	0	1
1	1	0

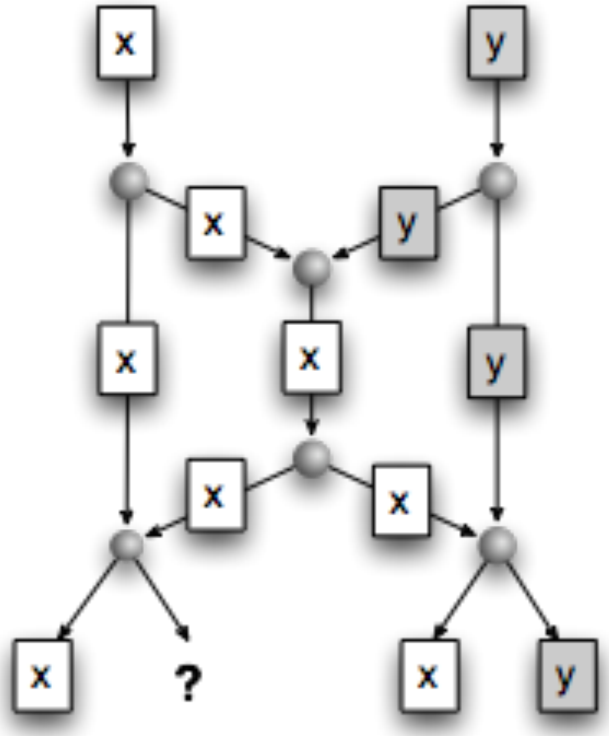
$x, \underbrace{x \oplus y \oplus x}_{y}$

$y, \underbrace{y \oplus x \oplus y}_{x}$

Network Coding

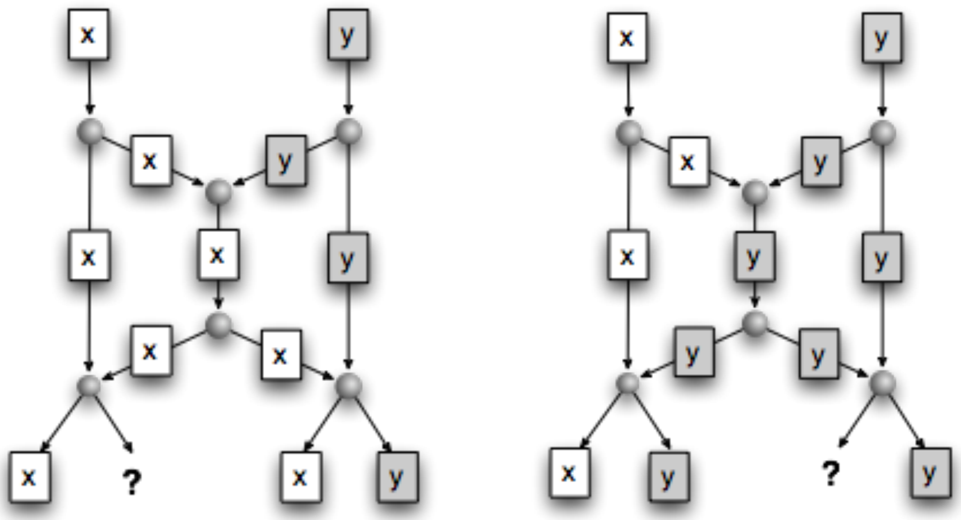
- R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow", (IEEE Transactions on Information Theory, IT-46, pp. 1204-1216, 2000)

- 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

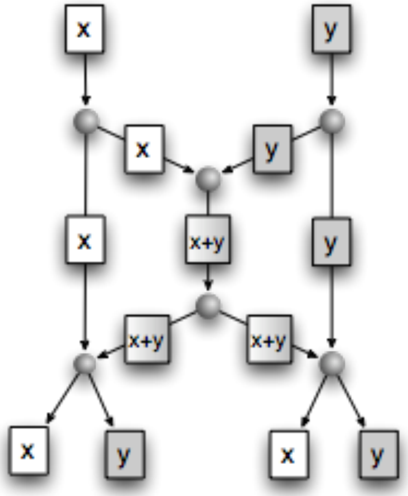


Network Coding → P2P ②

- 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


x
y
x
y
x
x+y
y
x+y
x+y
x
y
x
y