University of Würzburg
Informatik III (Distributed Systems)
Prof. Dr. P. Tran-Gia

Comparison of Crawling Strategies for an Optimized Mobile P2P

Tobias Hoßfeld, Andreas Mäder, Kurt Tutschku

www-info3.informatik.uni-wuerzburg.de

Mobile Peer-to-Peer (MoPi) Project

- from 10/2003 – 09/2004

SIEMENS

Siemens AG Com
Frank-Uwe Andersen, Dr. Cornelia Kappler

University of Würzburg, Department of Distributed Systems
Prof. Dr. Phuoc Tran-Gia, Dr. Kurt Tutschku, Tobias Hoßfeld, Andreas Mäder

UNIVERSITÄT PASSAU
Fakultät für Mathematik und Informatik

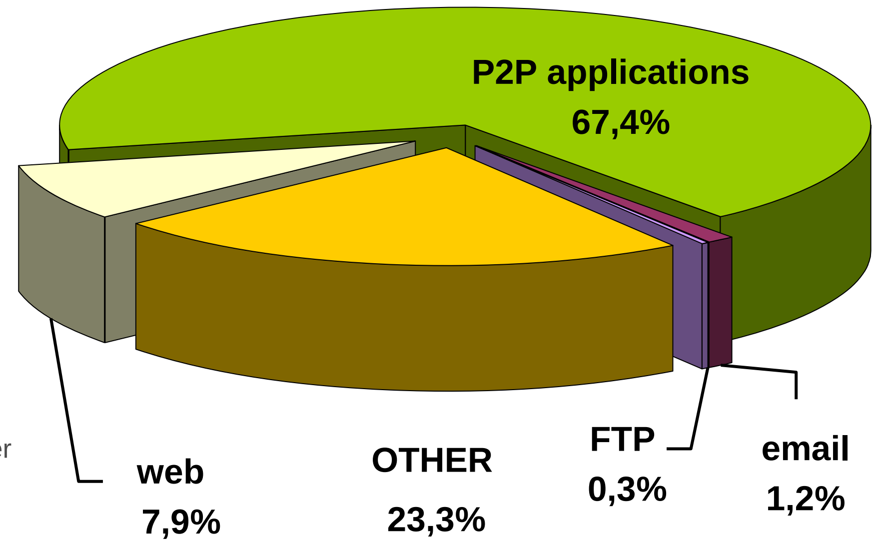
University of Passau, Computer Networks & Communications
Prof. Dr. Hermann de Meer, Jens Oberender, Ivan Dedinkski

- http://www3.informatik.uni-wuerzburg.de/staff/mopi/

Contents

Motivation

- P2P applications are highly popular in today's Internet
- UMTS operators are searching for new packet-oriented applications which…
  - exploit the potential of UMTS
  - motivate users to adopt the new technology



source:
Telefonica 2004
Jose Enriquez
COST 279, Rome: traffic observed in a transit router

- **Solution:** operator supported mobile P2P file-sharing network

... so far in MoPi

- Which P2P architecture is suited for an operator to **add value**?
  - **Hybrid architecture** like eDonkey
- Is P2P file-sharing **feasible** in a mobile environment?
  - **Yes**, however with some restrictions
- What **performance** will we achieve?
  - **GPRS** for instant-messaging mode
  - **UMTS** enables P2P file-sharing
- How can we **improve** the system's performance?
  - seamless enhancement of P2P network w/o protocol modification
  - **caching peer** stores popular resources, i.e. speeds up downloads
  - **crawling peer** locates resources, i.e. saves signalling traffic

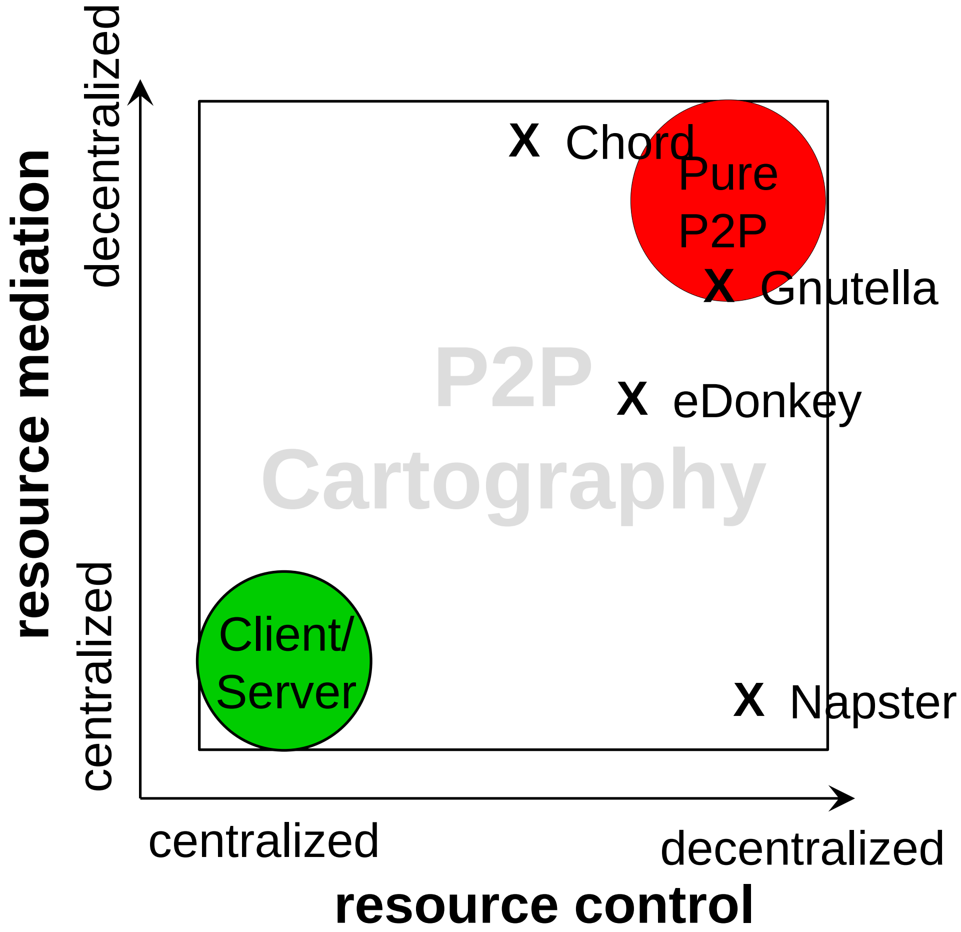
Contents

P2P Cartography

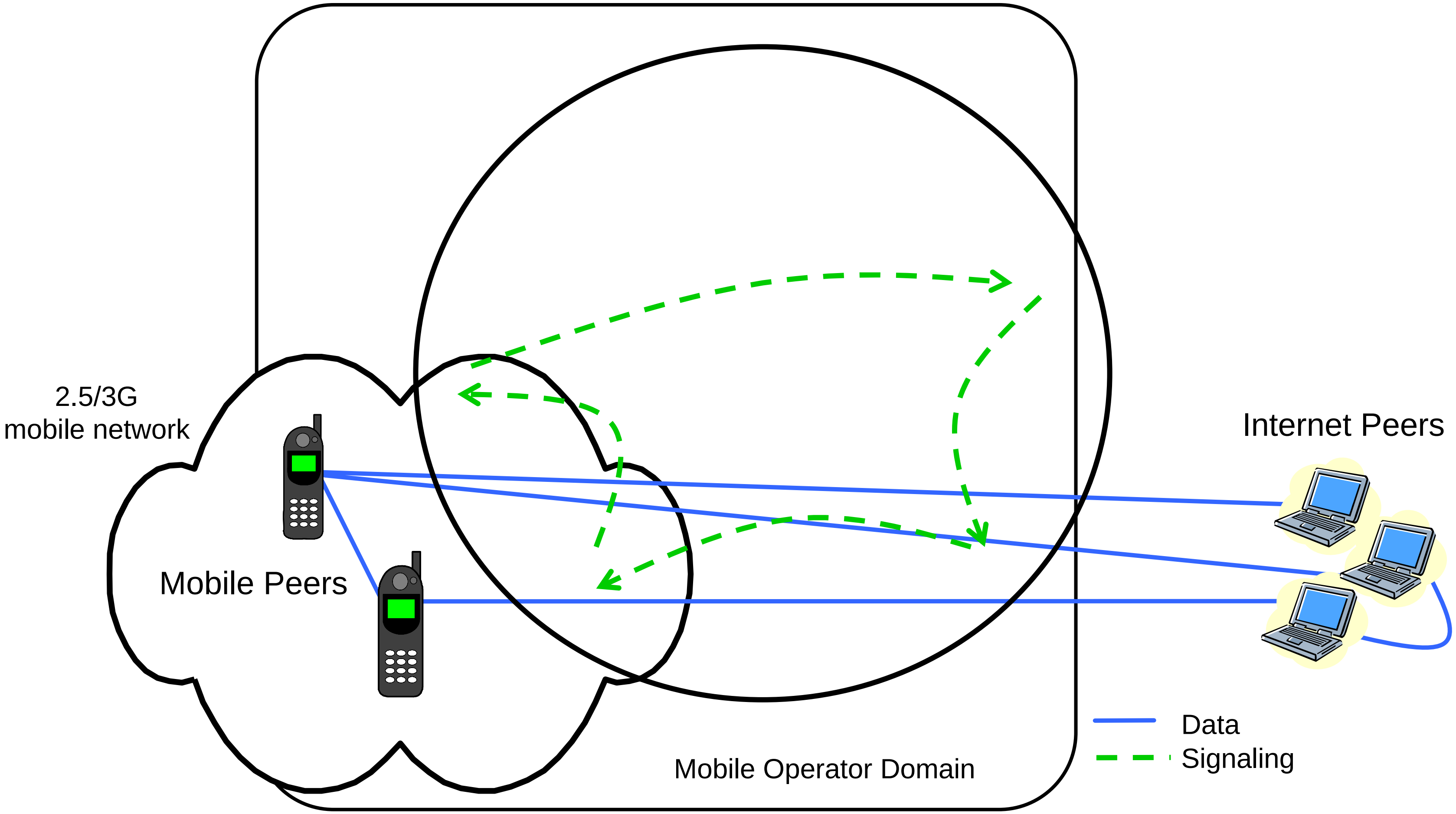
- Peers share resources
  i.e. storage, CPU cycles

- Wide range between Pure P2P and Client/Server
- Parameters
  - Resource Mediation
    (how are resources located)
  - Resource Control
    (who may access and when)

- Current State-of-the-Art
  - Gnutella
  - Napster
  - eDonkey

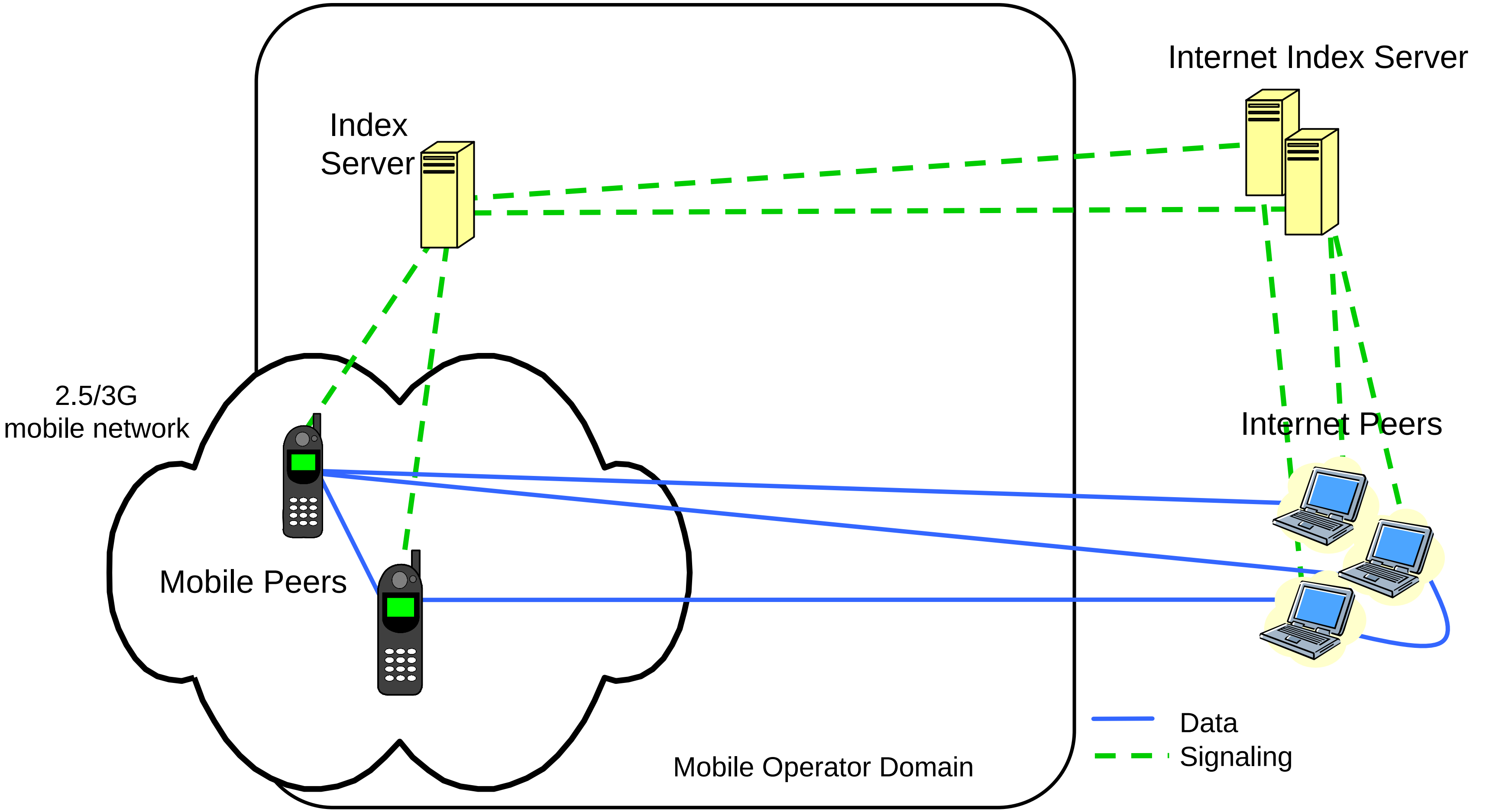


Resource Mediation in Mobile P2P Networks



Structured P2P Network

Resource Mediation in Mobile P2P Networks

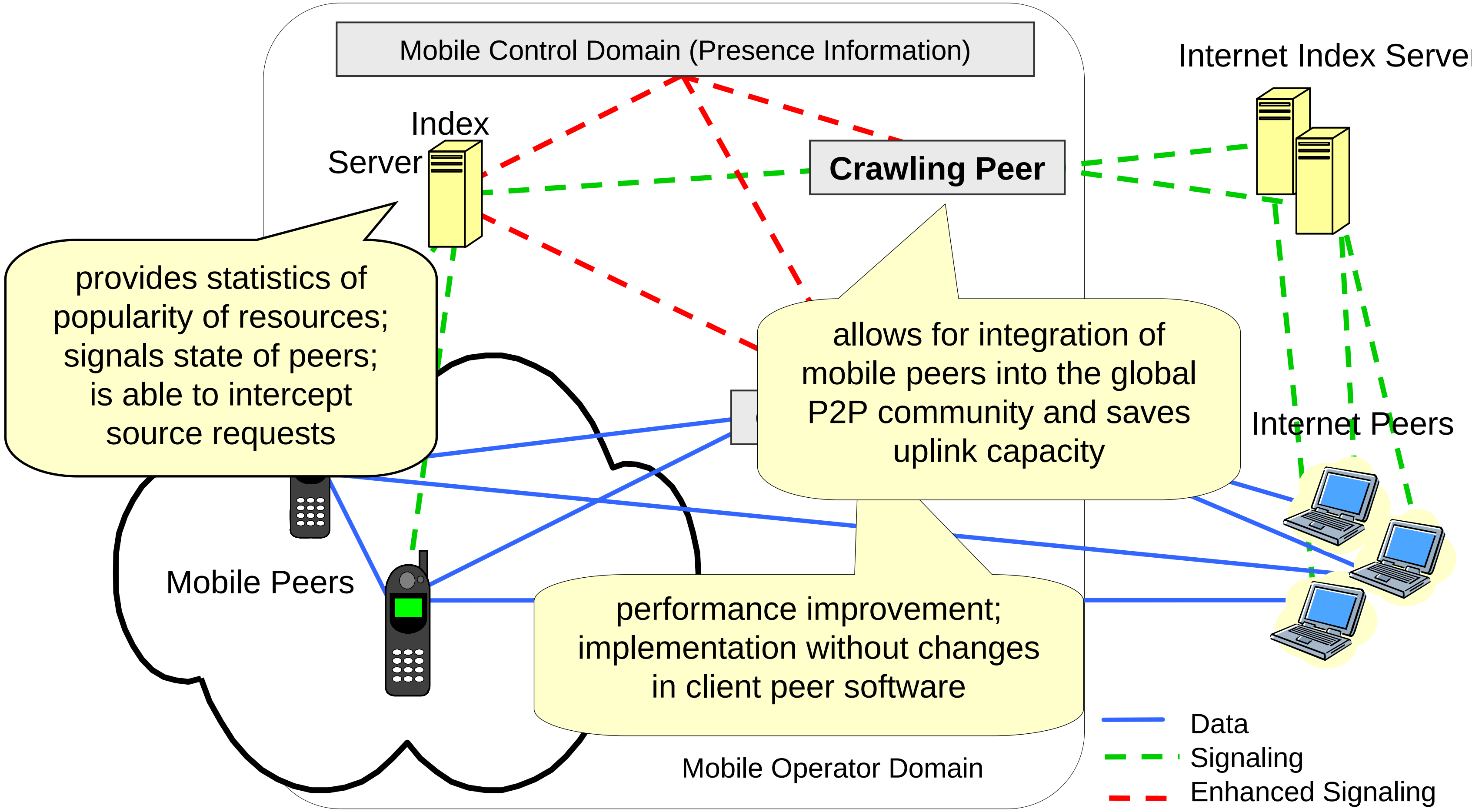


eDonkey-based architecture

Hybrid P2P Network

- with centralized elements as resource directory
- provider-operated P2P entities within mobile domain

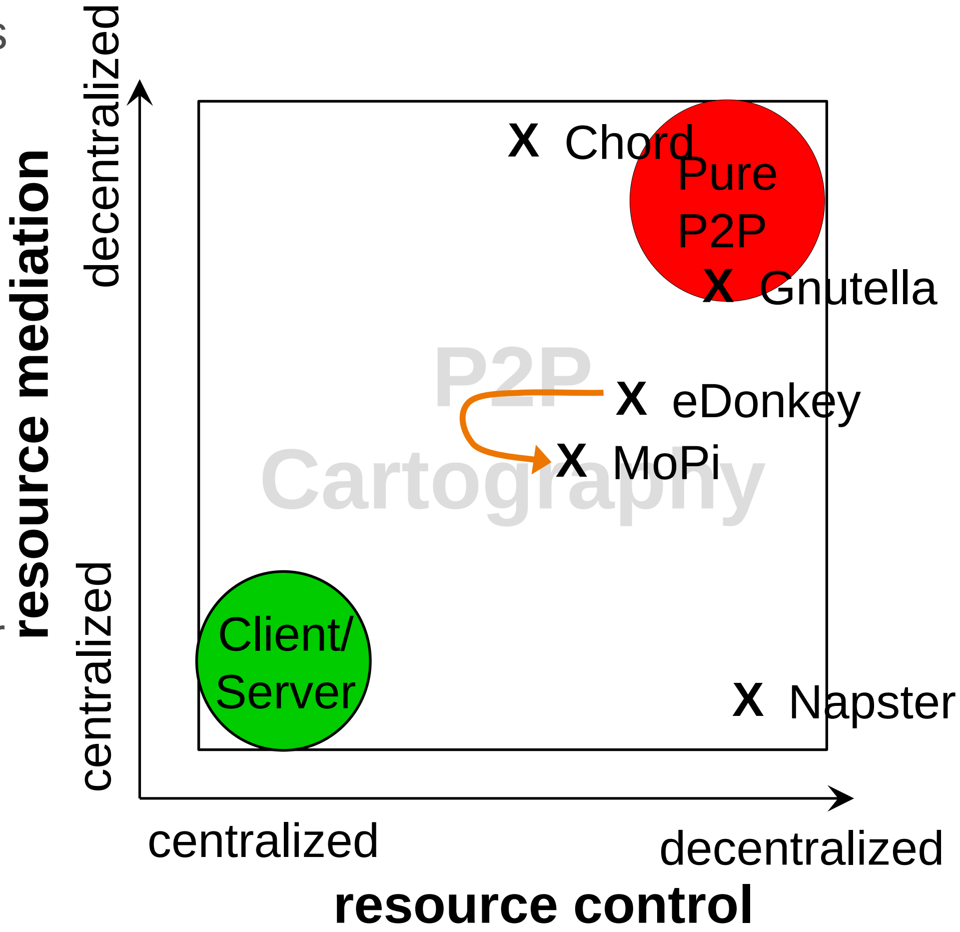
Mobile P2P Architecture



- ***eDonkey-based architecture*** with provider-operated P2P entities
  - indexing server: resource directory
  - caching peer: stores popular resources
  - crawling peer: outside interface

Advantages of Hybrid MoPi Architecture

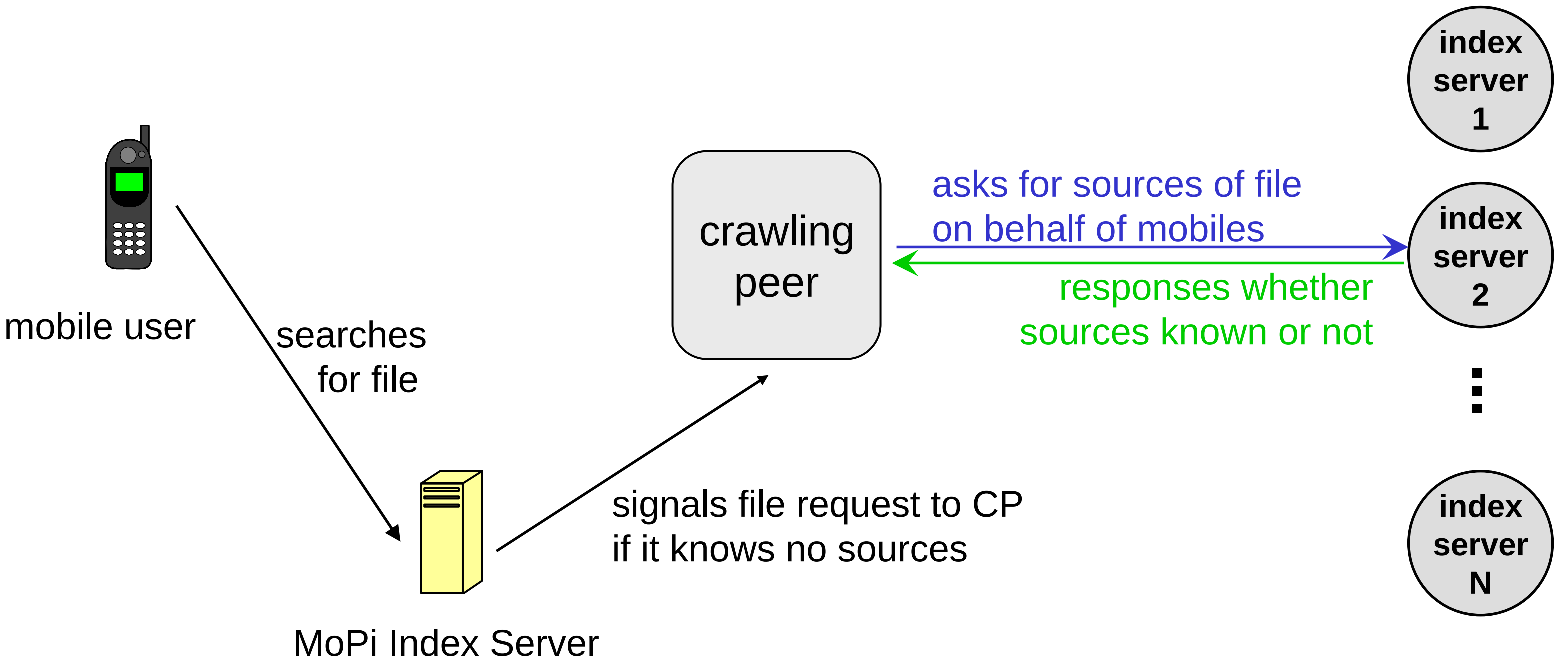
- Give control to operator
  - Sell operator-provided services
- Realize user preferences
  - Based on eDonkey = high user acceptance
- Replace M2M transmissions
  - Check what data can be provided from within the operator domain
- Crawling Peer
  - Shift of signaling traffic from air interface to wired part
  - Problem of searching unique filenames is avoidable
  - Searching is even possible when mobile subscribers are offline



Contents

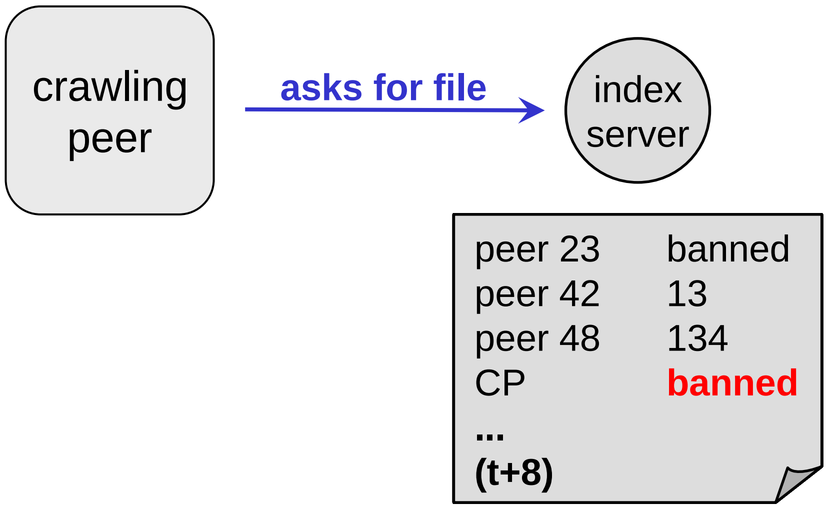
How works the crawling peer?

- Allows for integration of mobile peers into the global P2P community, saves uplink capacity, and shifts traffic to wired part
- Can also locate contents when mobile user is offline
- Interaction: MoPi -mobile control domain (presence information)



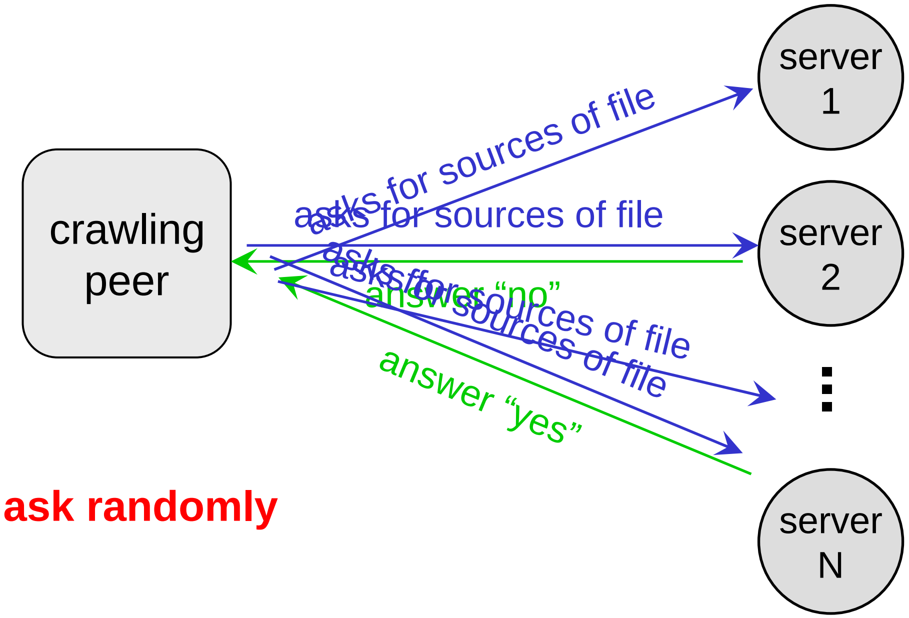
Banning and Credit Points

- An index server has credit points for each peer
- After each second, the credit points are incremented by one
- A file request costs 16 credit points
- If credit points are below zero, the peer is banned at server
- Banning means no response to source requests



Comparison of Crawling Strategies

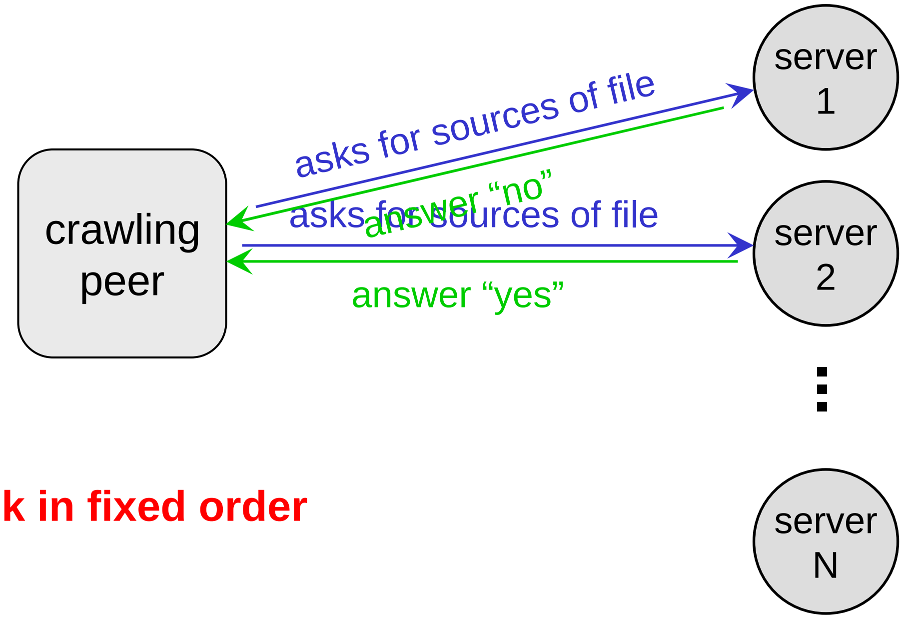
- Strategy of crawling peer
  - randomly requesting servers – RaRe strategy
  - optimizing success probabilities – Psi strategy
  - smart requesting without banning – NoBan strategy
- Impact of parallel requests



RaRe strategy: ask randomly

Comparison of Crawling Strategies

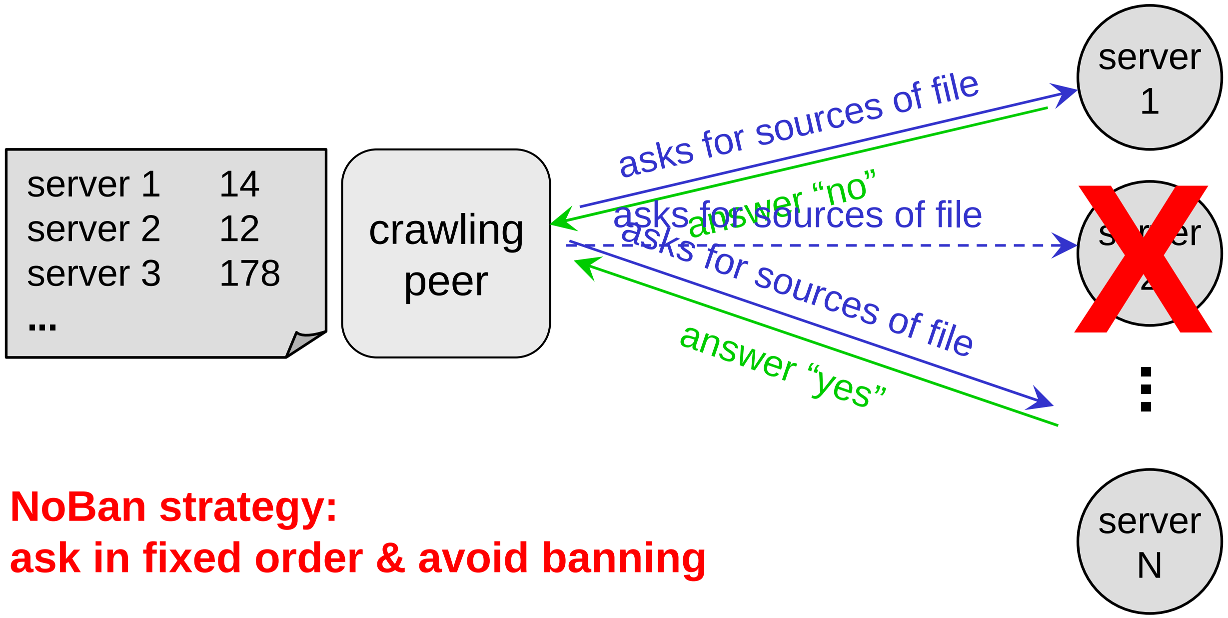
- Strategy of crawling peer
  - randomly requesting servers – RaRe strategy
  - optimizing success probabilities – Psi strategy
  - smart requesting without banning – NoBan strategy
- Impact of parallel requests



Psi strategy: ask in fixed order

Comparison of Crawling Strategies

- Strategy of crawling peer
  - randomly requesting servers – RaRe strategy
  - optimizing success probabilities – Psi strategy
  - smart requesting without banning – NoBan strategy
- Impact of parallel requests



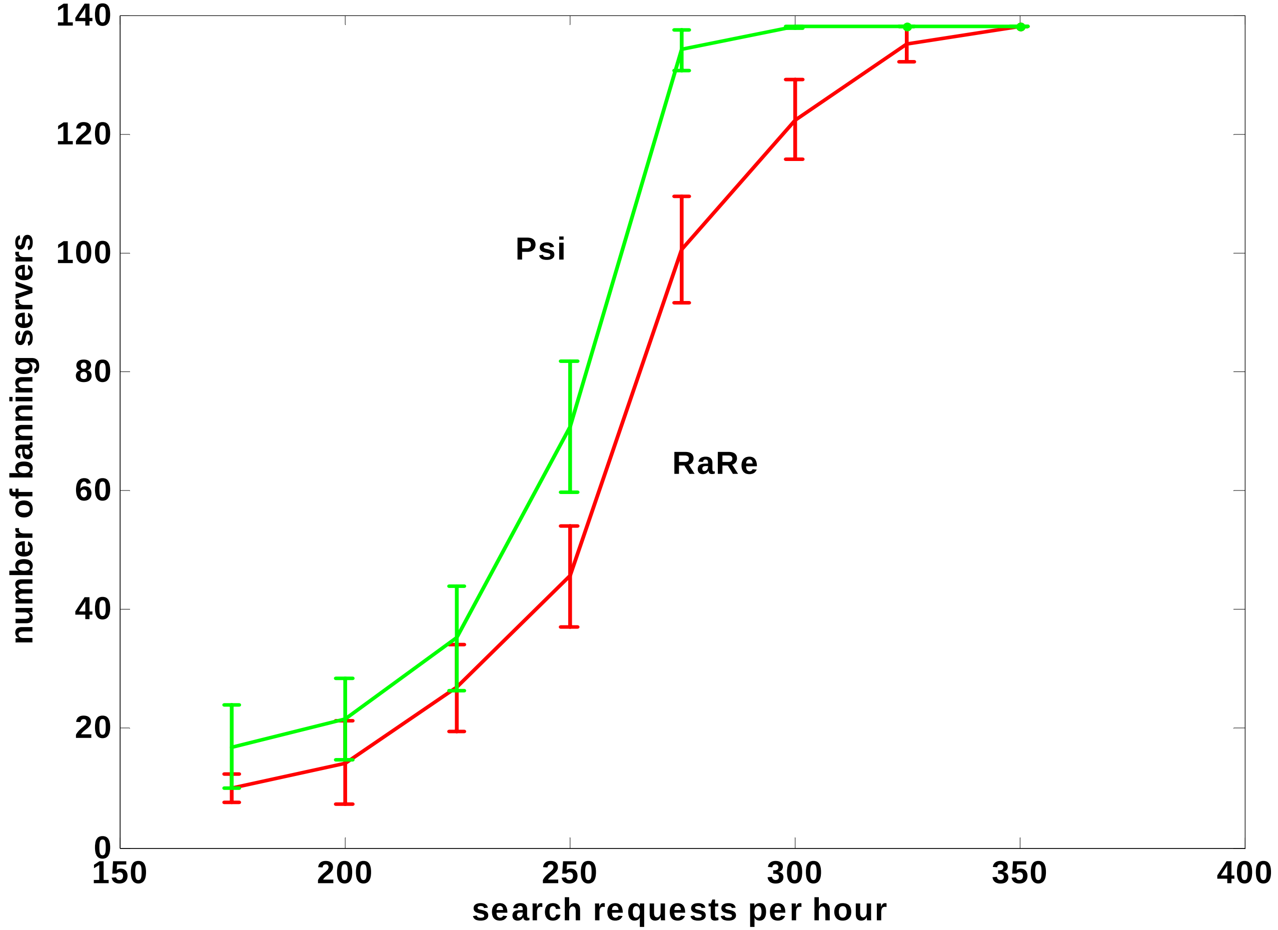
NoBan strategy:
ask in fixed order & avoid banning

$$T_i = RTT_i + DB_i$$

Contents

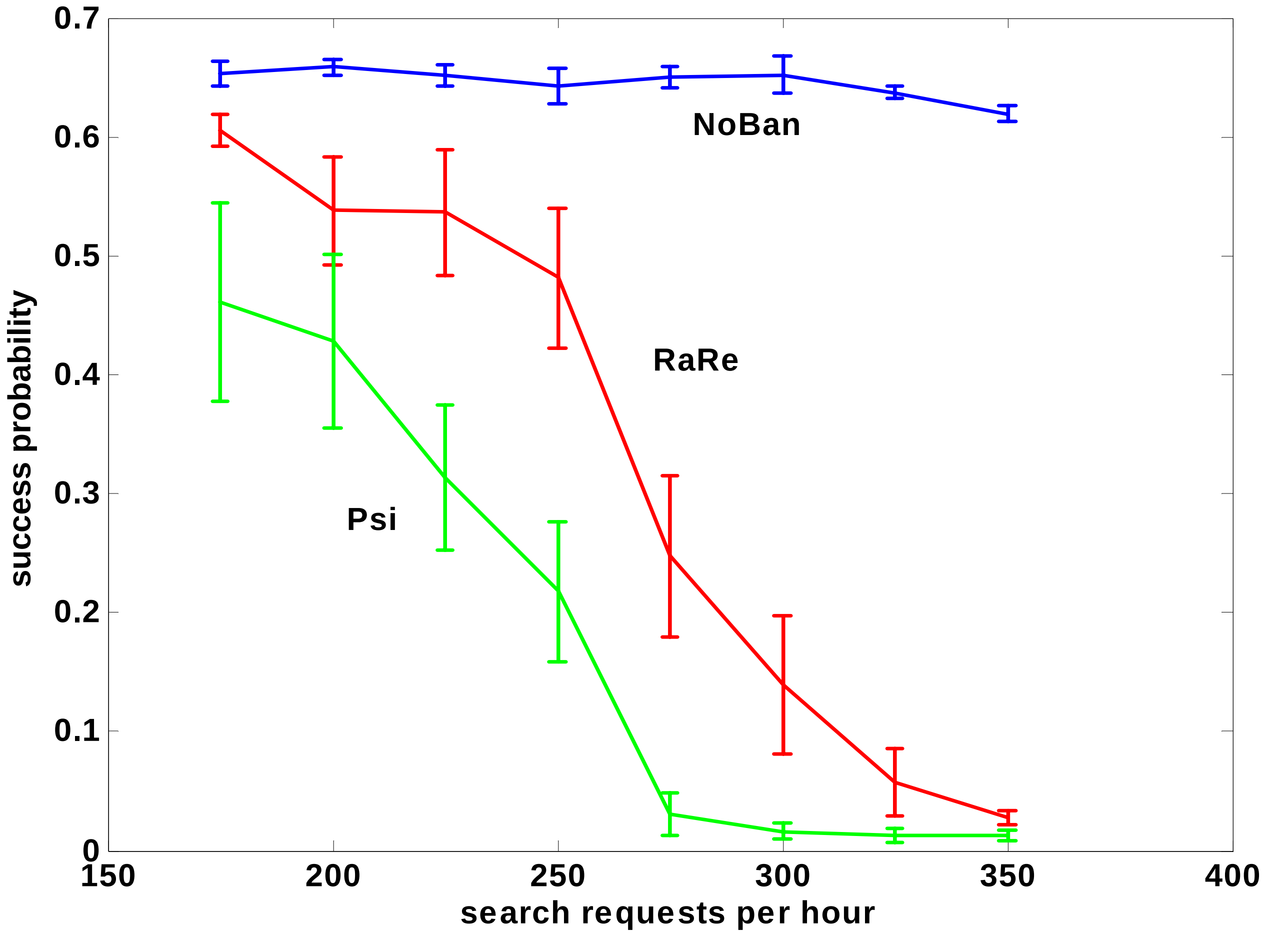
Number of Banning Index Servers

- For high search request rates, CP is almost banned from every index server



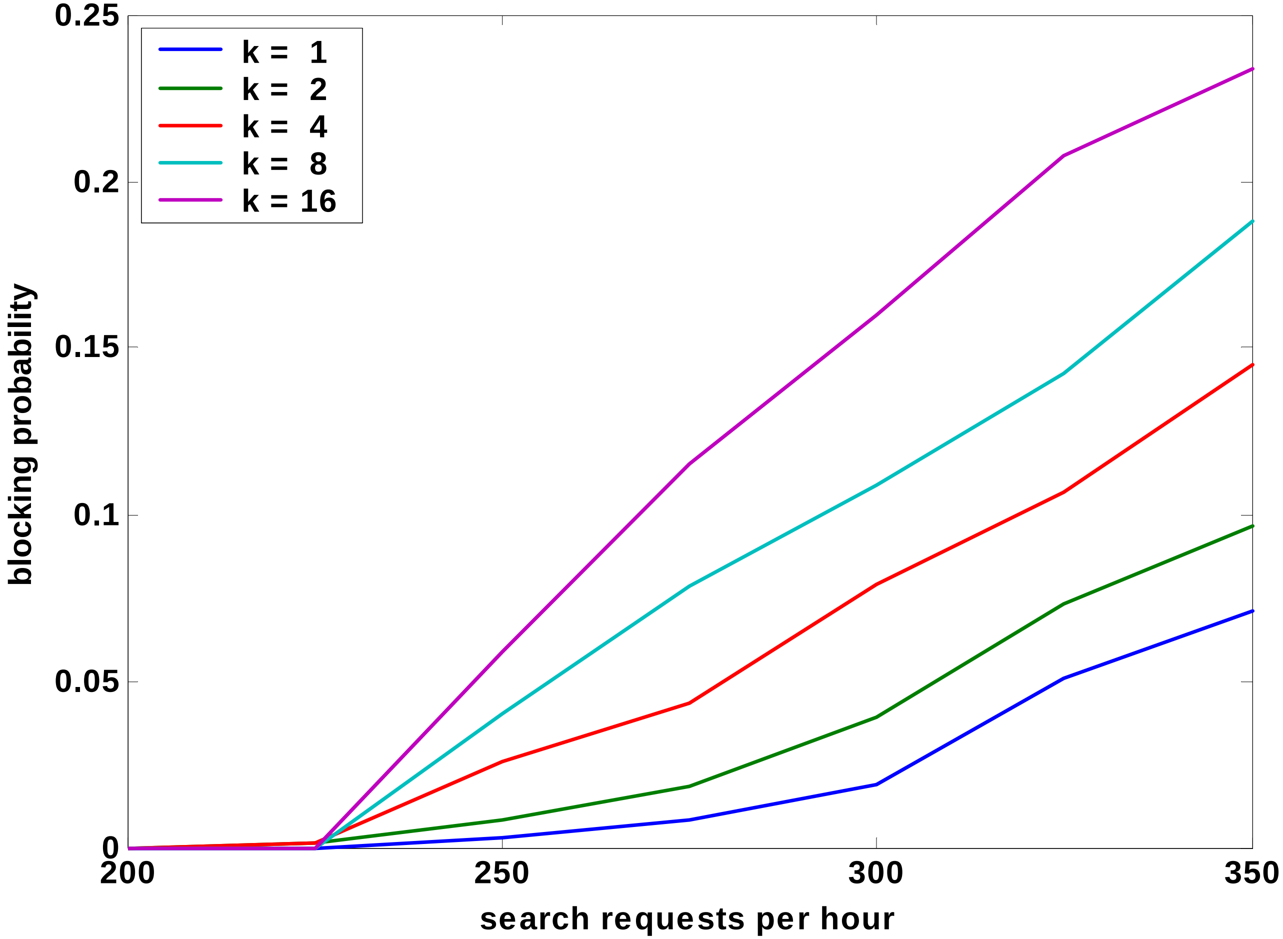
Success Probability of Search Requests

- Success probability to find a file tends toward zero for RaRe/Psi
- NoBan strategy ranges near maximal success probability



NoBan Strategy – Blocking Probability

- Number of parallel requests decreases response time at the cost of increased blocking probability, i.e. less success



Contents

Conclusions and Outlook

Crawling peer

- different query strategies considered: NoBan performs best
- optimizes the resource mediation mechanism in a mobile P2P file-sharing architecture; independent of churn behavior
- saves signaling traffic and shifts traffic from air interface to wired part of the network
- analytical approach enables parameter sensitivity studies

Current and future work

- enhance and finalize analytical approach
- investigate structured P2P approaches for locating contents
  - overhead
  - response time

Backup-Slides

www-info3.informatik.uni-wuerzburg.de

Investigation of Crawling Peer

- Crawling peer optimizes resource mediation, i.e. location of contents, in a mobile P2P file-sharing architecture

- Comparison of **different strategies** of the crawling peer
  - randomly requesting servers – RaRe strategy
  - optimizing success probabilities – Psi strategy
  - smart requesting without banning – NoBan strategy

  to be presented at **ITC19**, Beijing, China, September 2005

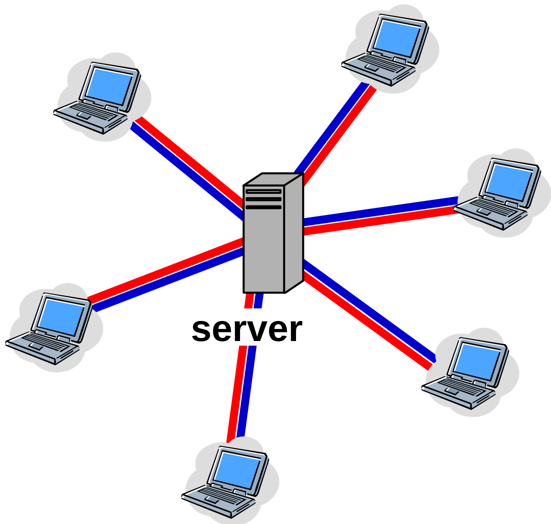
- **Analytical performance evaluation** of crawling peer
  - investigation of different scenarios
  - enabling of parameter-sensitivity studies
  - further optimizations of the strategy

  2nd **EuroNGI** work-shop WP IA.8.2, Como, Italy, July 2005

ITG, 5. Würzburger Workshop

Client/Server vs. eDonkey Network

- Conventional client/server



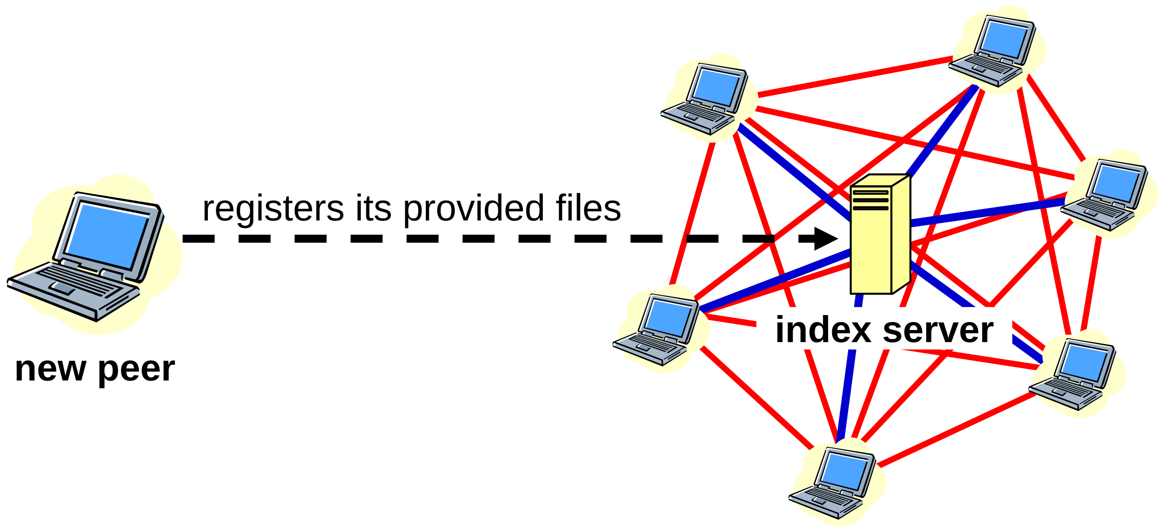
- Server is central entity and the only information provider
- Client is requestor
- File requests are managed by the server

- Hybrid eDonkey P2P architecture

- Central server is used as index database for resource location
- Files are shared between peers
- Peers directly exchange files
- Peer is provider and requestor

Participation in the eDonkey Net

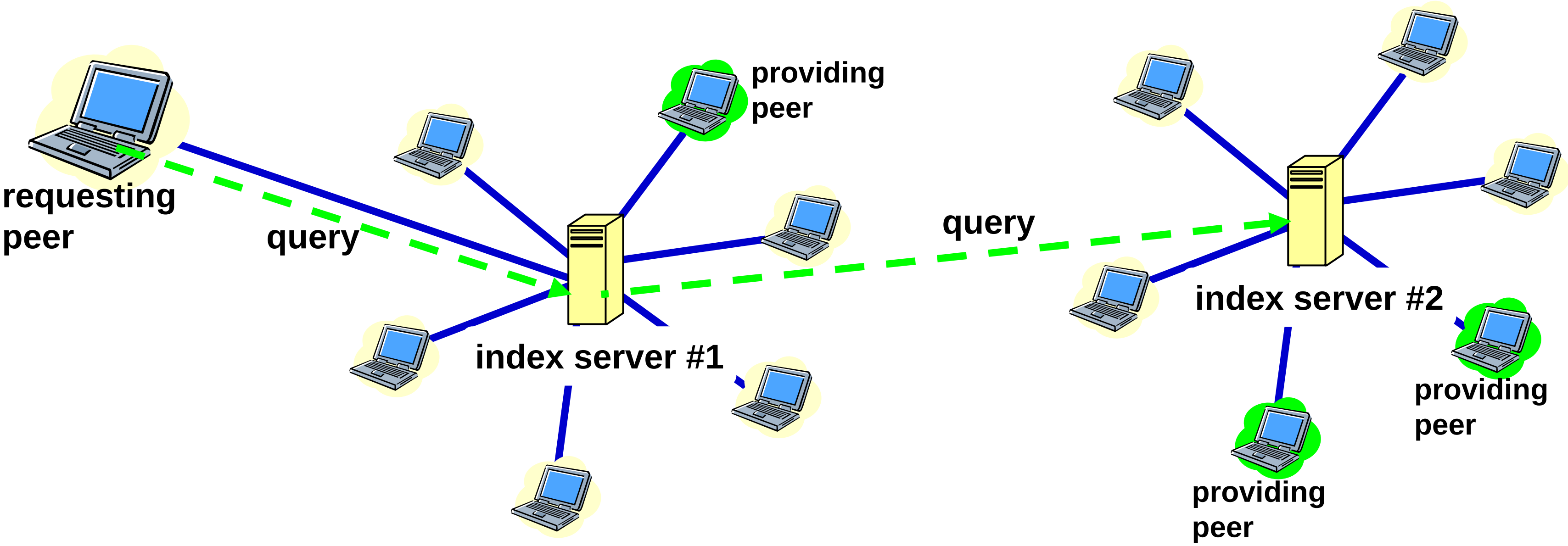
- Peer wanting to join the network registers at an index server
- Index server knows all files shared by its connected clients



- All peers in the eDonkey network may download the provided files from the new peer.

Searching of Files

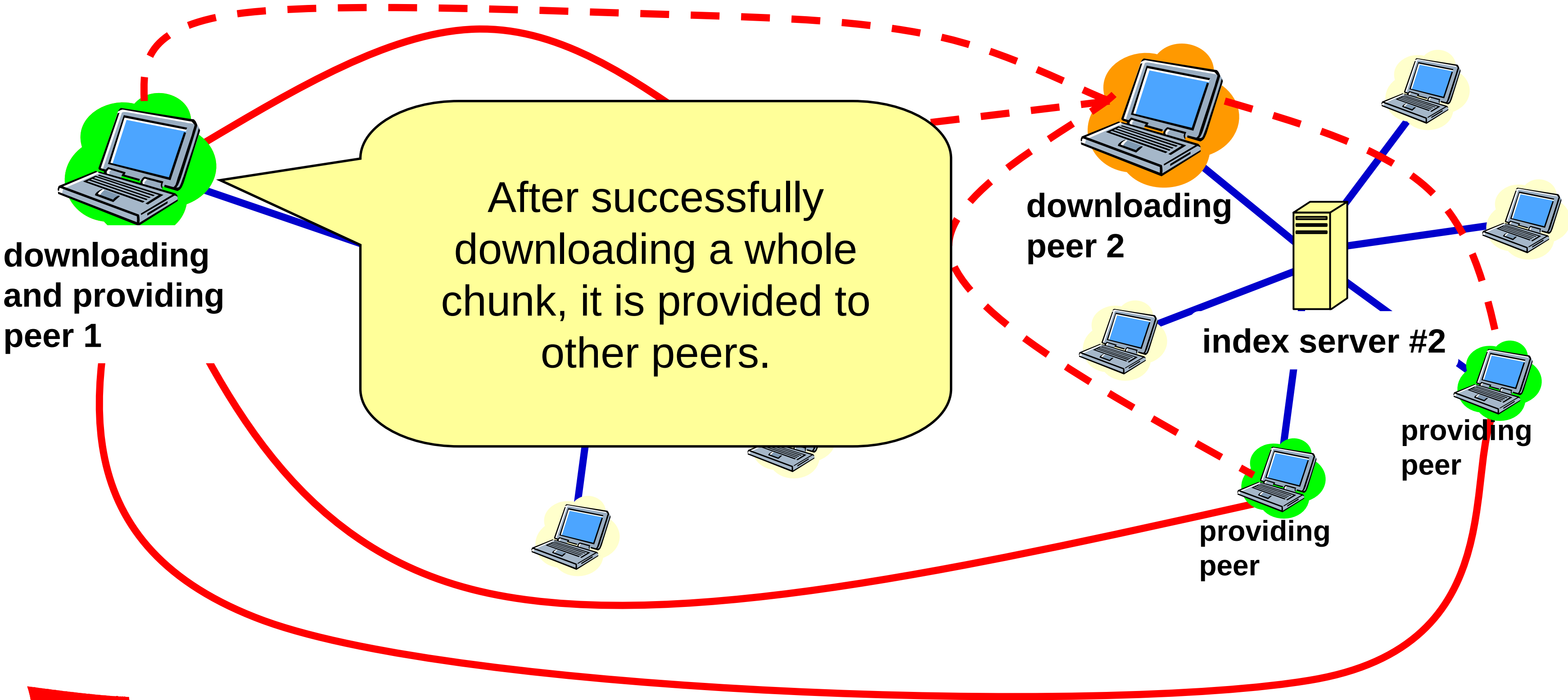
- File requesting peer sends a query to its index server.
- Index server returns the list of all providing peers.



- If none or an insufficient number of matches is returned, the client may resubmit the query to another index server.

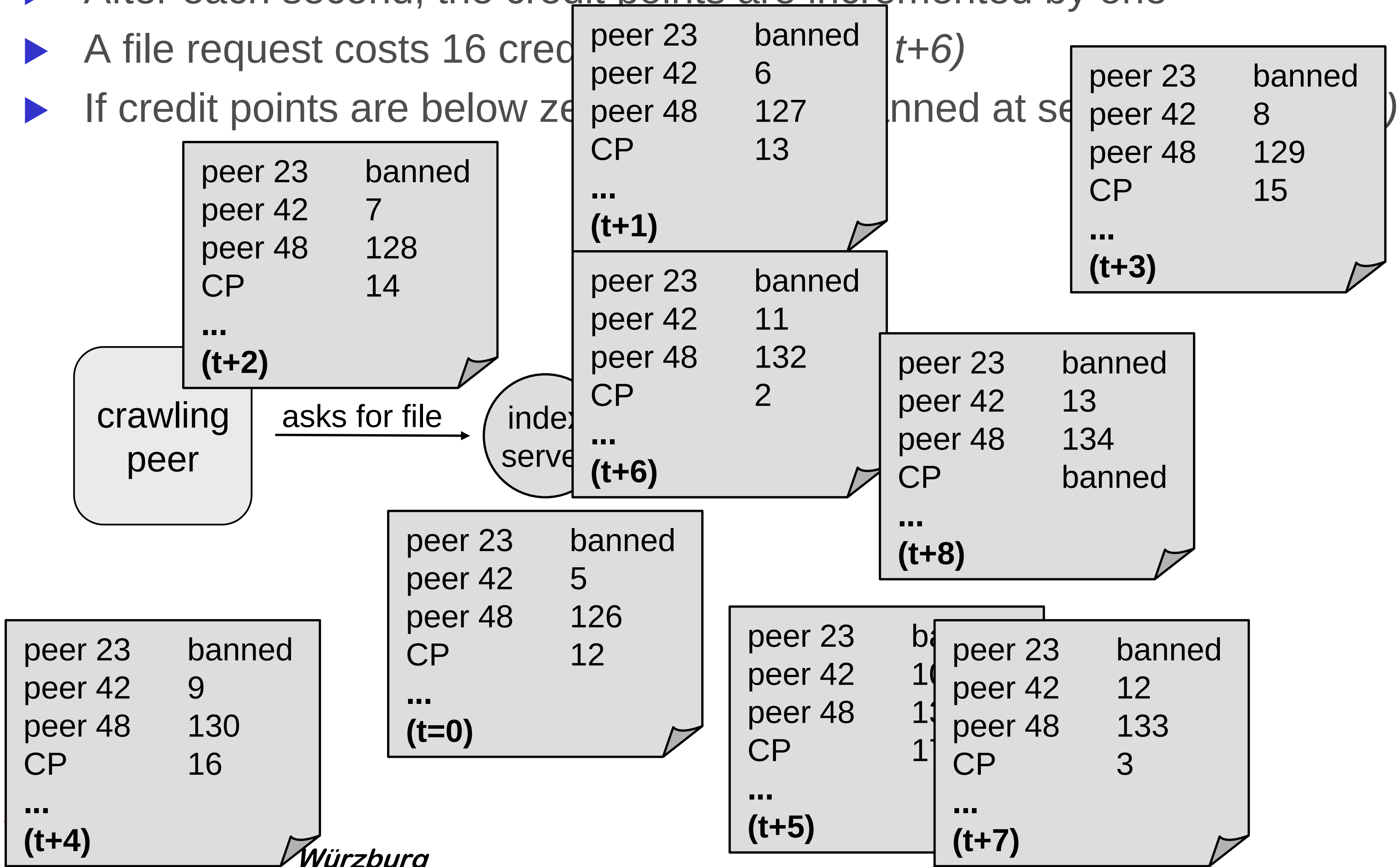
Downloading of Files

- Main feature of eDonkey is **multiple source download**.
- Peers issue several download requests for the same file to multiple providing peers in parallel.
- Providing peers serve the requesting peers simultaneously.



Banning and Credit Points

- An index server has credit points for each peer
- After each second, the credit points are incremented by one
- A file request costs 16 cred … t+6)
- If credit points are below ze … nned at se …)



What means blocking?

- For each file request x
  - a list L of all index servers exists
  - which denotes if server y was already requested for request x
  - $L(y)=1$: server y requested; $L(y)=0$: server not yet requested
- A request x is blocked if no more server y in $S=\{y:L(y)=0\}$ can be contacted, i.e. credits $c_y < 16$
- it is $S \neq \{\ \}$, otherwise the search was unsuccessful

- We have assumed this kind of blocking in order to avoid a waiting queue for requests at any index server
- The reason is that newly arriving request would then be blocked if the waiting list is not empty; then the waiting queue grows and grows and grows...

What means successful?

- A request is successfully answered if there is any index server y which has this file registered
- The probability that server y has the file is f_y

- A request is called unsuccessful, if the request is issued to every available index server, i.e. all N=138 index servers have answered “File not known”
- The probability for unsuccessful requests is

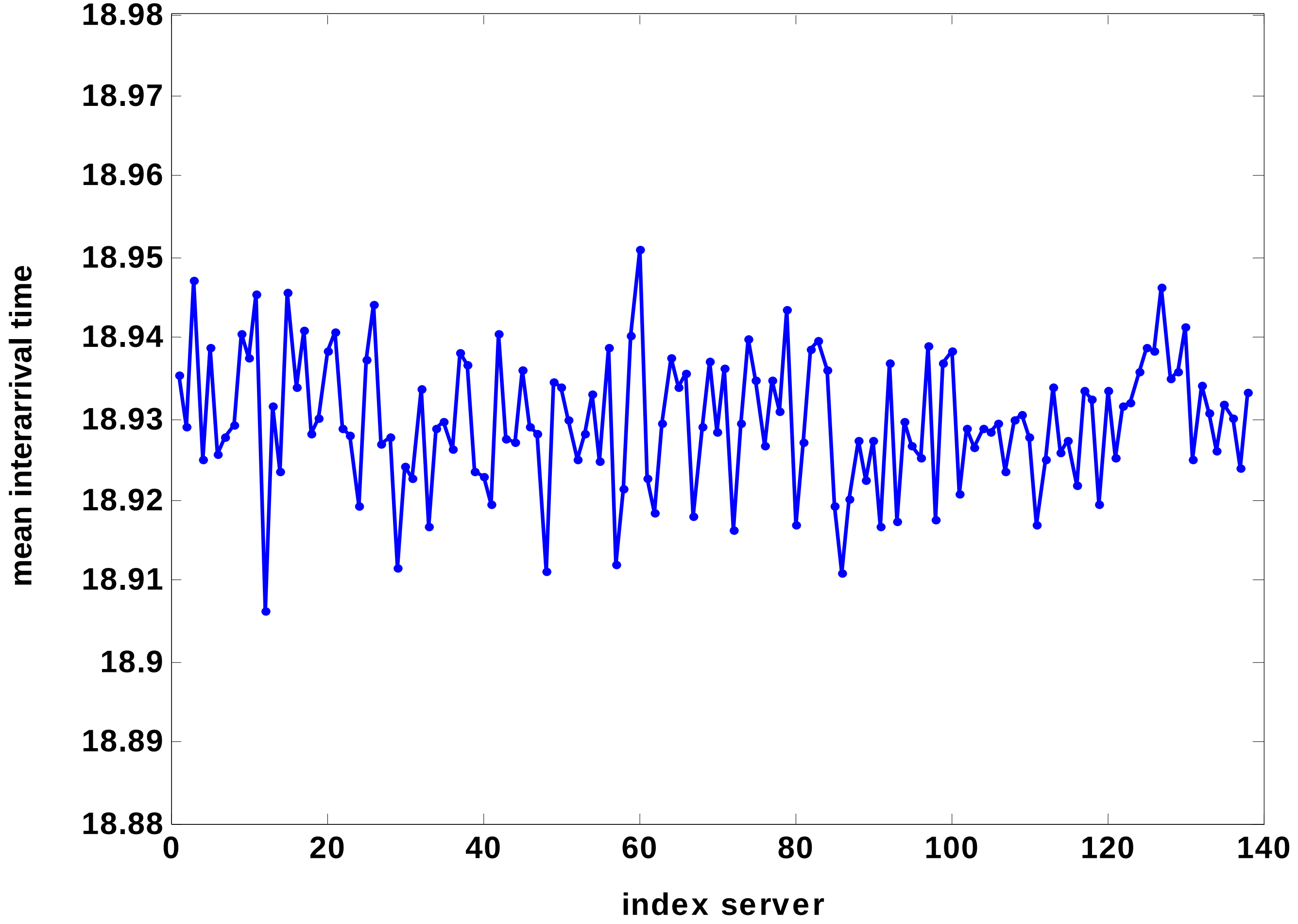
$$\prod_{i=1}^{N}(1-f_i)$$

Which servers are contacted?

- Consider file request x
- List of servers is L whereby
  - L(y)=1: server y requested; L(y)=0: server not yet requested
  - List of servers is sorted by success probability f_y of each server y (i.e. number of registered files at server y)
- Next server y to be contacted fulfills
  - L(y)=0: not yet asked server
  - c_y > 16: enough credit points available
  - y = random(i: L(i)=0 & c_i>16): ask randomly servers (with equal probability)

Mean interarrival time

- consider 200 simulation runs

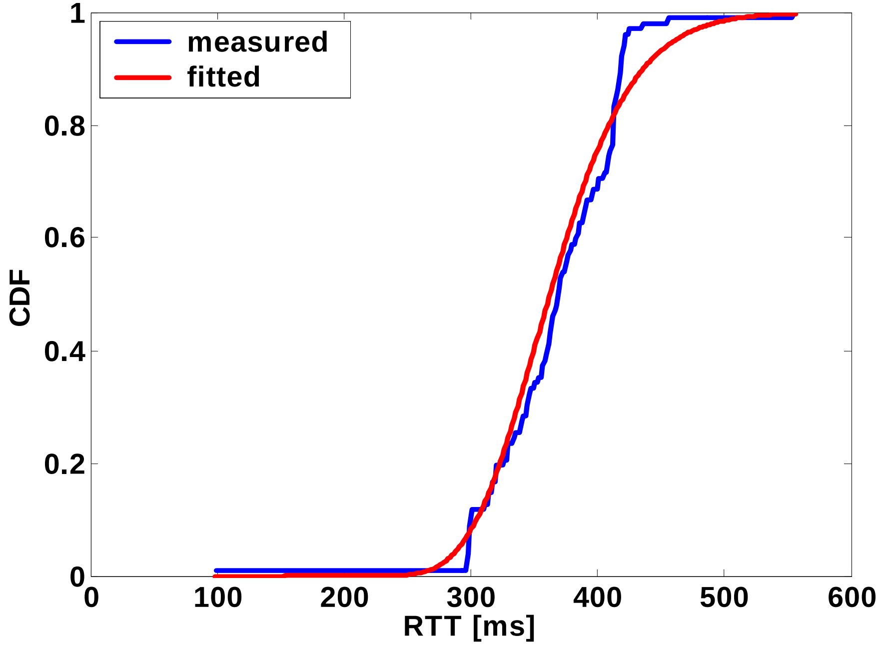


Blocking Probability

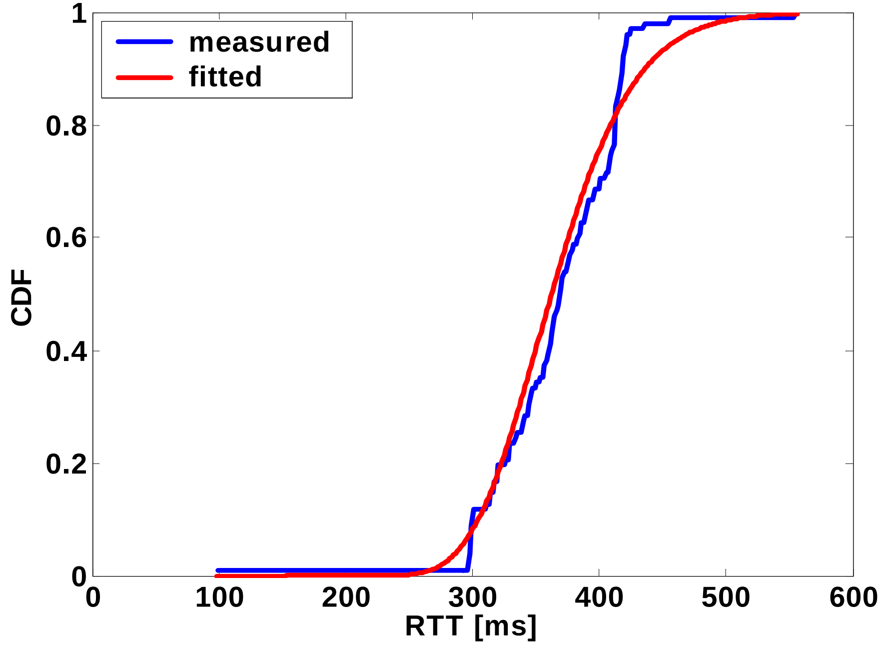
- Total number of index servers is N
- File request arrival rate at crawling peer is λ
- Success probabilities p_s are equal on each index server
- $\lambda_I = \lambda N \left(1 - p_{b,i}\right)$
- Unsuccessful search is forwarded to next server $\lambda_{i,j} = \lambda_I \left(1 - p_s\right)$
- Obtained rate at index server i $\lambda_i^* = \lambda_I + \sum_{j \neq i} \lambda_{j,i}$
- Blocking probability is $p_{b,total} = \prod_{i=1}^{N} p_{b,i}$
- Number of credit points at each index server using Power method
- Problem: Computation of $p_{b,i}$

Measurement of RTTs

- Largest eDonkey server with ID1



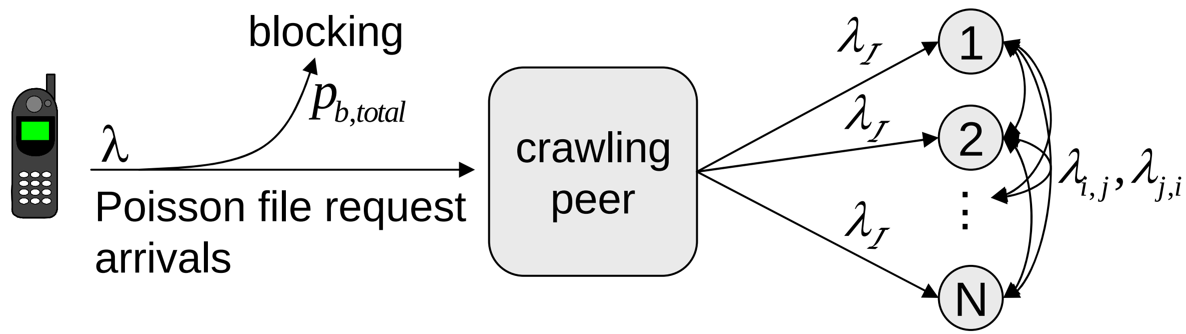
- Largest Chinese server with ID 48



- Fitting of measured round trip times $\tilde{R}_i$
- $R_i = \mathrm{DET}(d) + \mathrm{LOGN}(m,s) = \min\left(\tilde{R}_i\right) + \mathrm{LOGN}\left(\mu\left(\tilde{R}_i - \min\left(\tilde{R}_i\right)\right), \sigma\left(\tilde{R}_i\right)\right)$
- one third of the index servers could not be pinged, i.e. 46 of 138

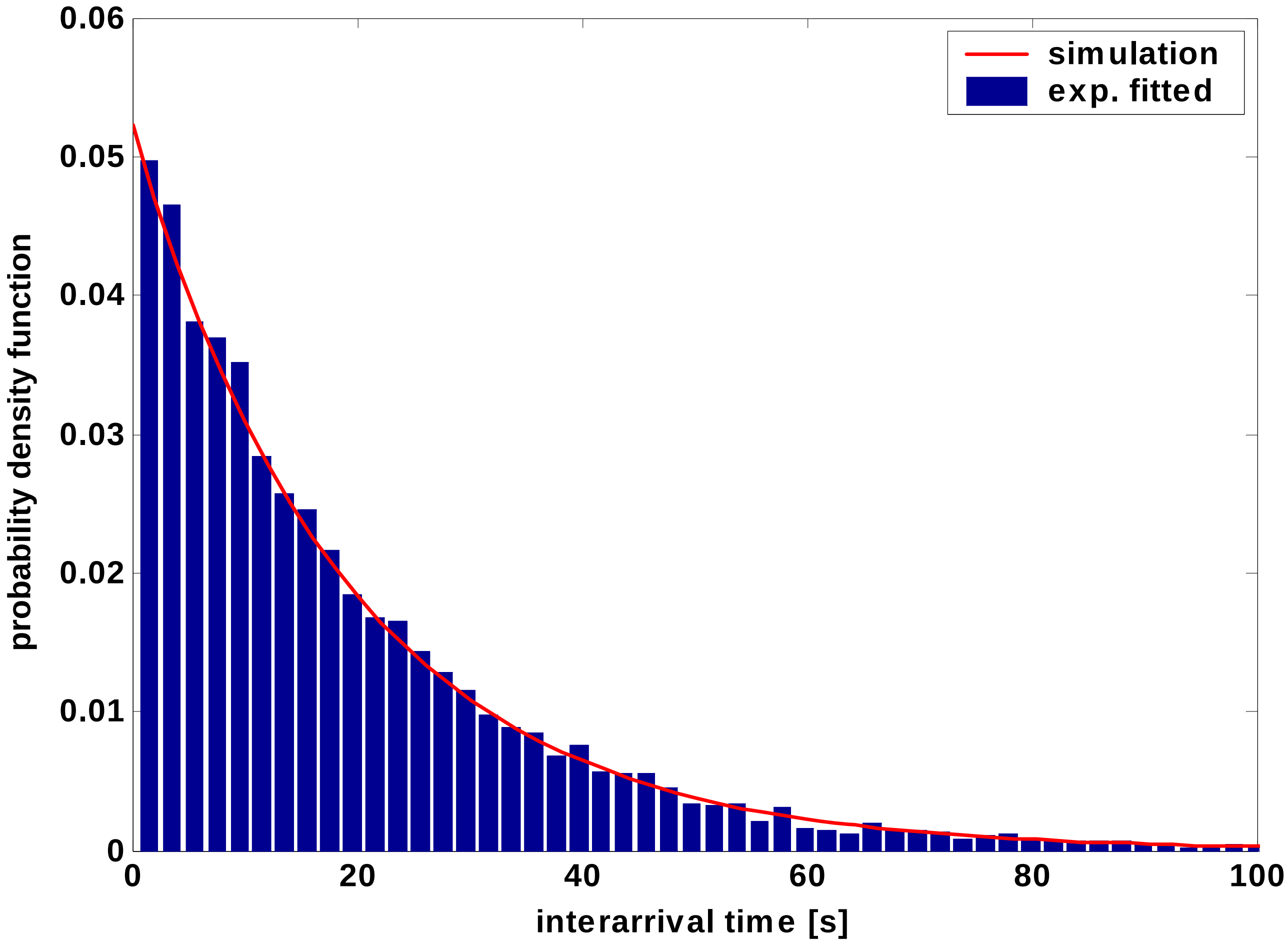
Analysis Model

- We assume the same probability for each index to know a file
- Poisson file request arrivals are split equally among N servers
- For each file request x
  - a list L of all index servers exists
  - which denotes if server y was already requested for request x
  - L(y)=1: server y requested; L(y)=0: server not yet requested
- A request x is blocked if no more server y in S={y:L(y)=0} can be contacted, i.e. credits c_y < 16



Observed interarrival times by simulations

- File request arrivals follow a Poisson process
- Observed arrivals at index server i still follow a Poisson process



NoBan Strategy – Mean Response Time

- asdf