A churn-resilient replication strategy for peer-to-peer distributed hash-tables

ANR – SHAMAN – Kickoff - 27 January, 2009

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

INRIA

centre de recherche PARIS - ROCQUENCOURT

Sergey Legchenko, Sébastien Monnet, Pierre Sens

UPMC PARIS UNIVERSITAS

CNRS

LIP6

Outline

- Background and definition

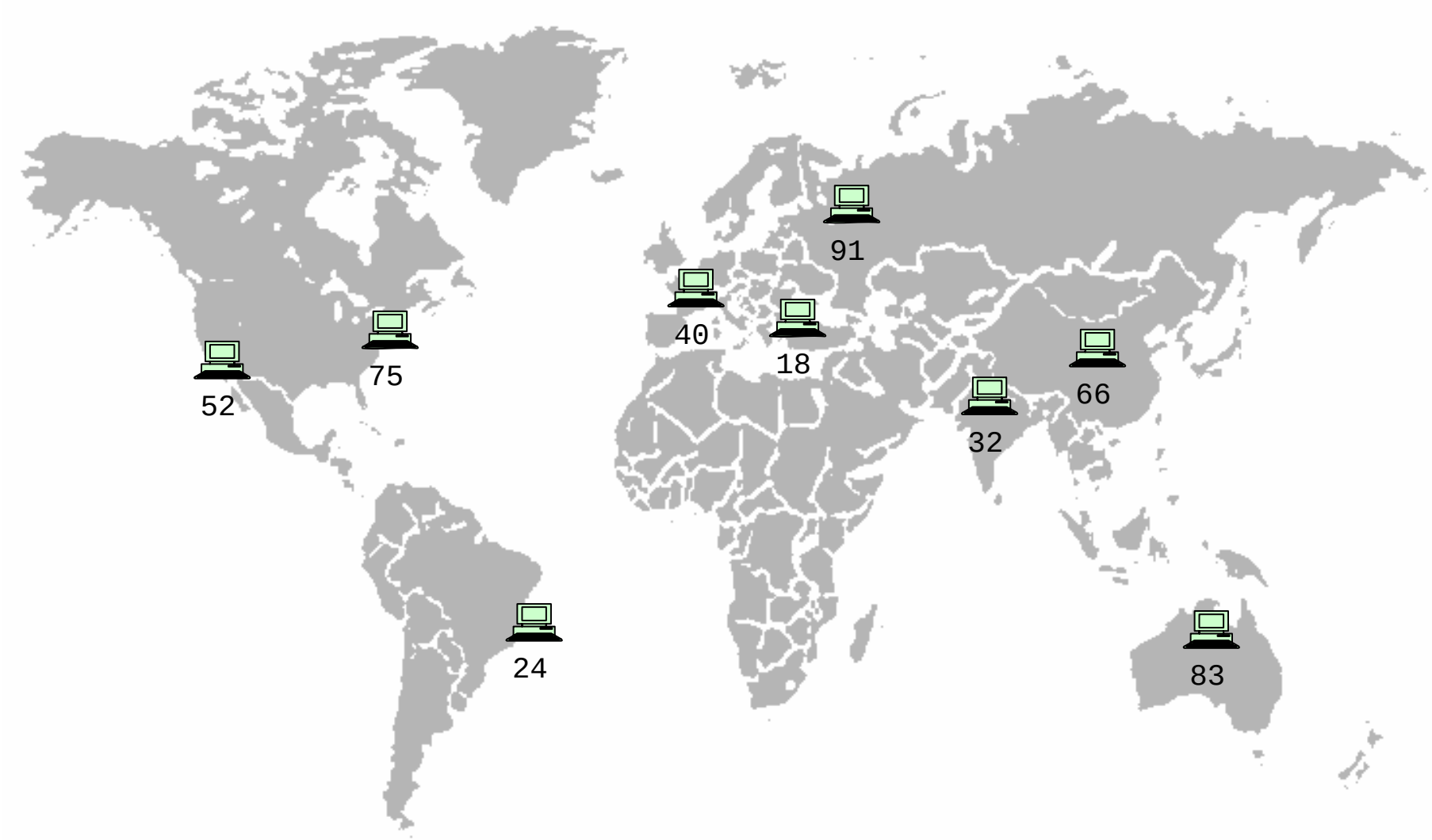
- Replication protocol in presence of churn

Background

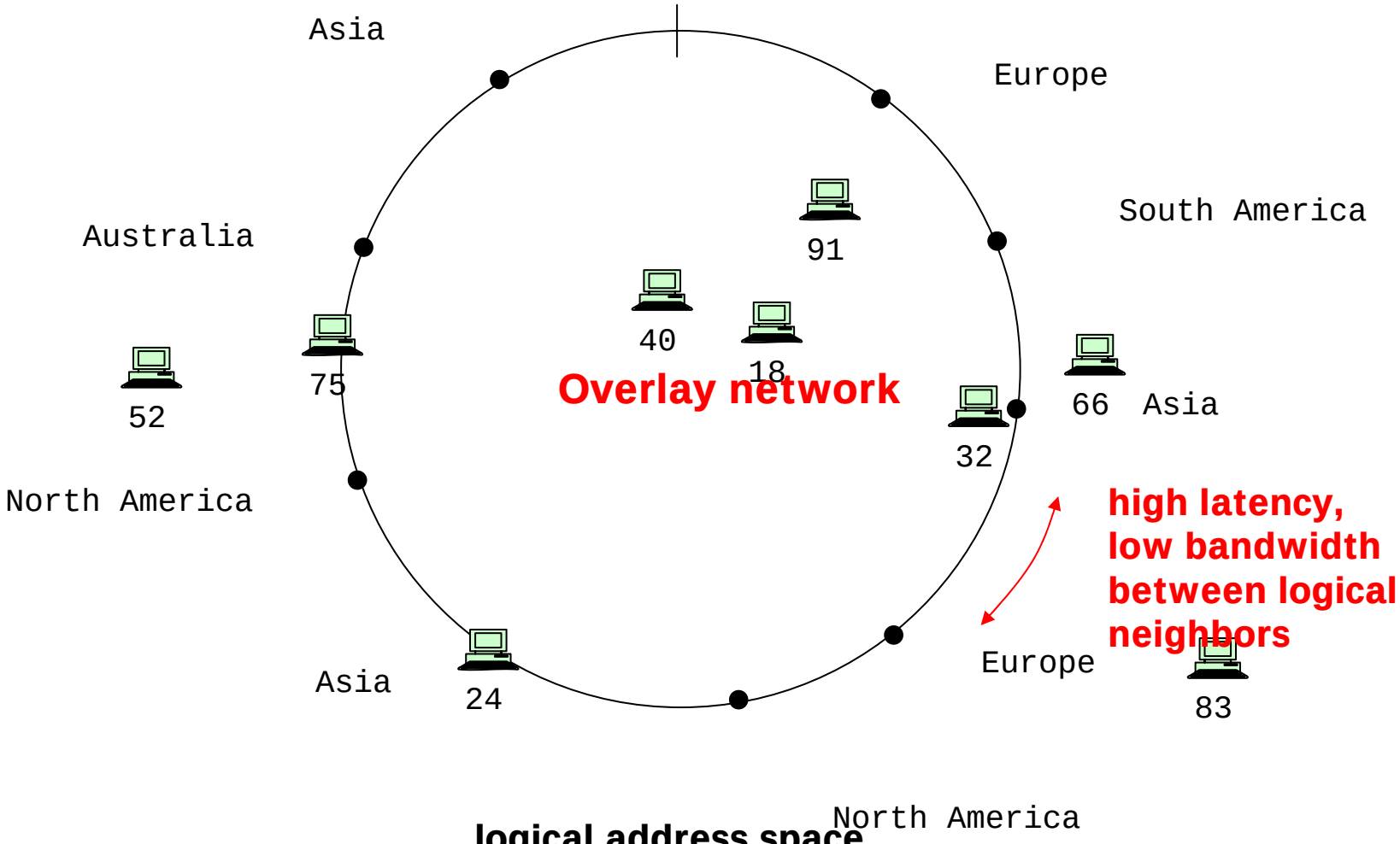
Peer-to-peer systems

- distribution
- symmetry (a node = a **peer** client and server)
- decentralized control
- dynamicity
- self-organization

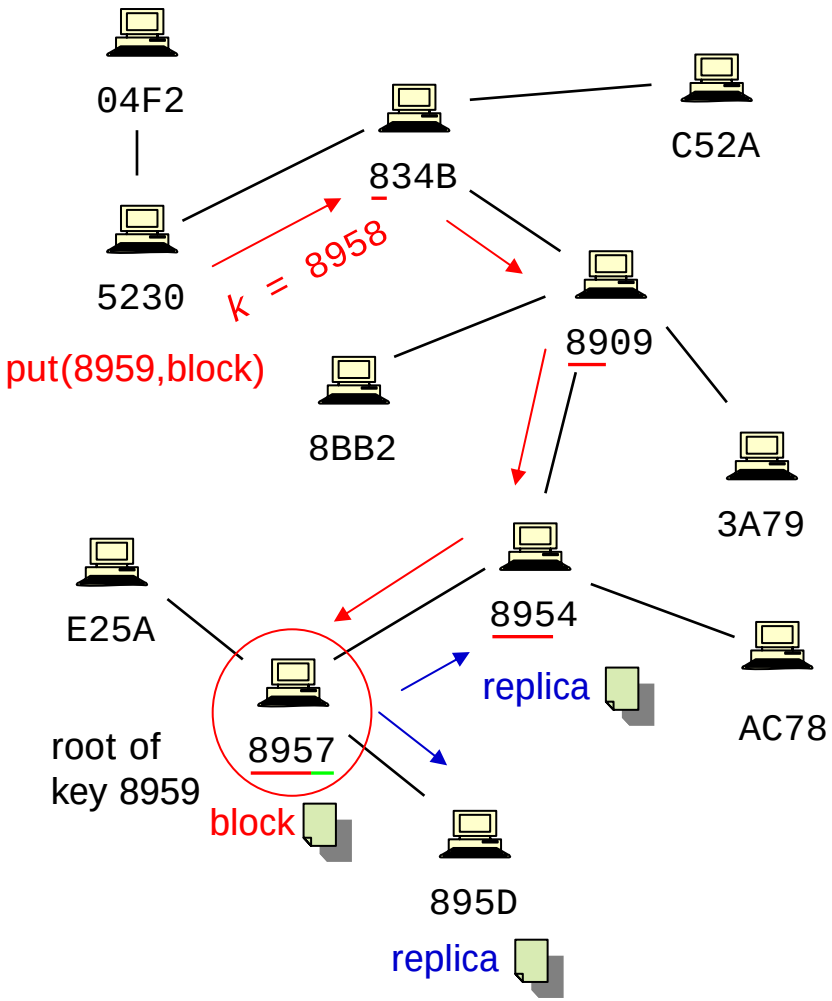
Distributed Hash Tables

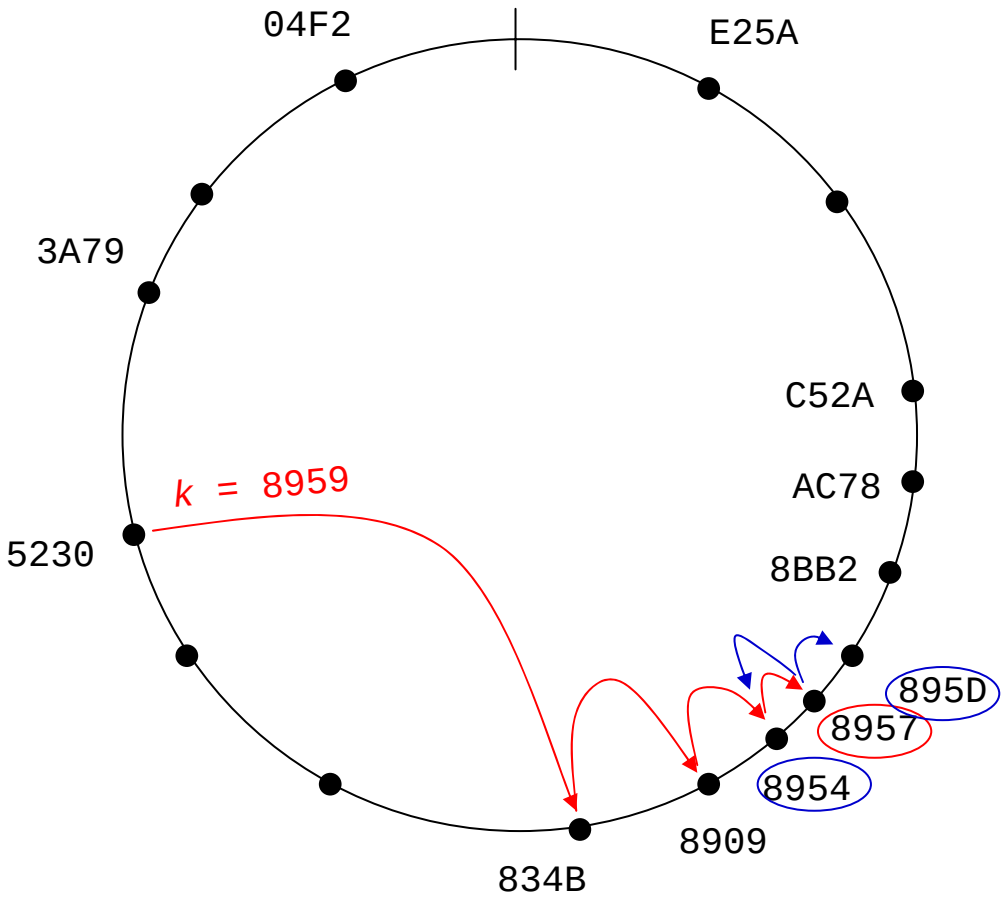


DHTs



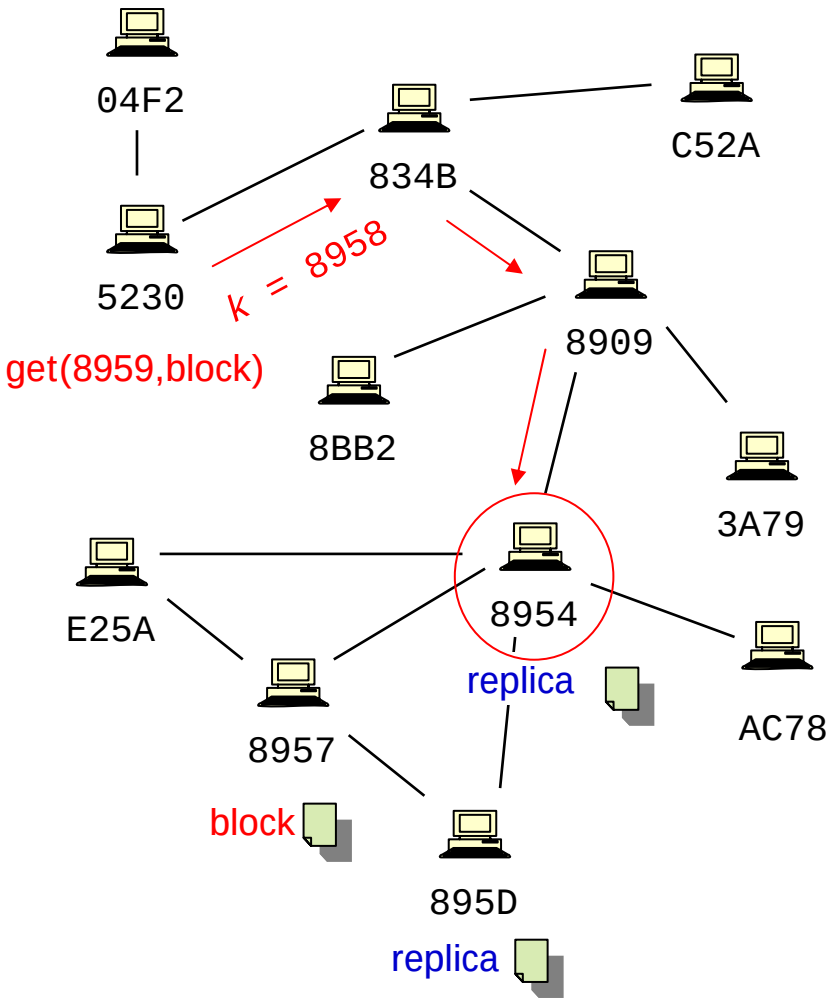
Insertion of blocks in DHT

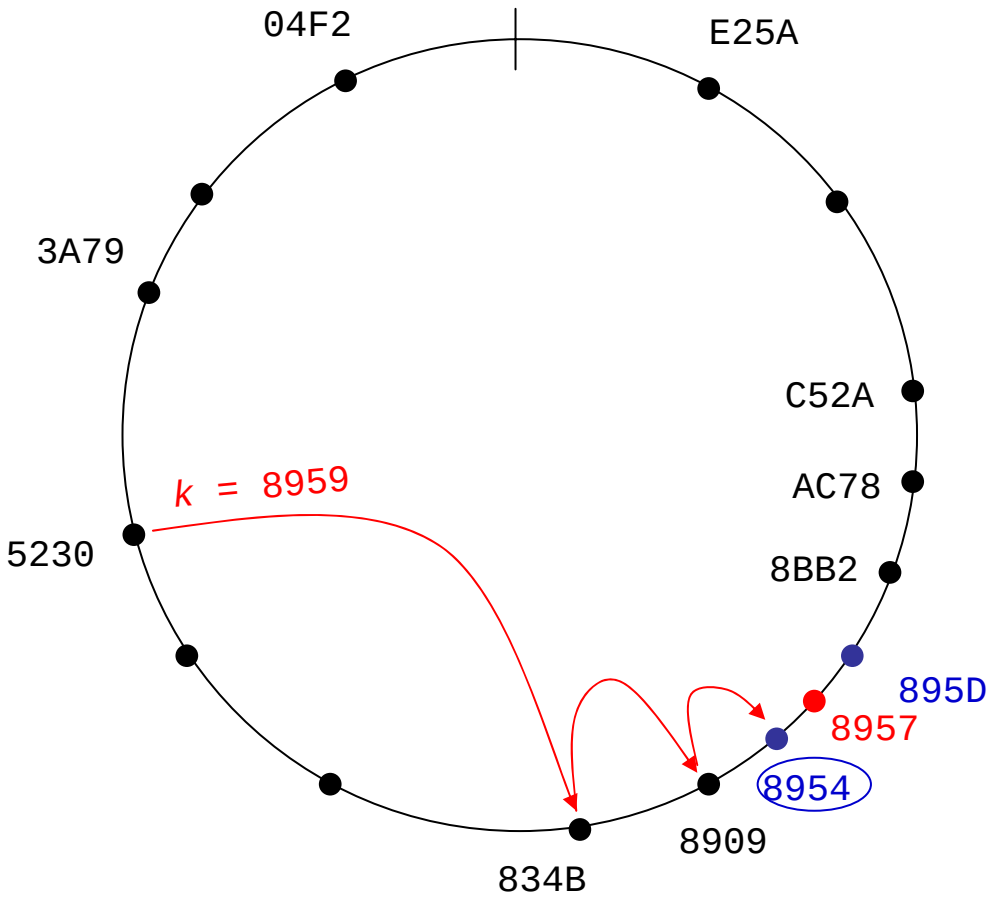




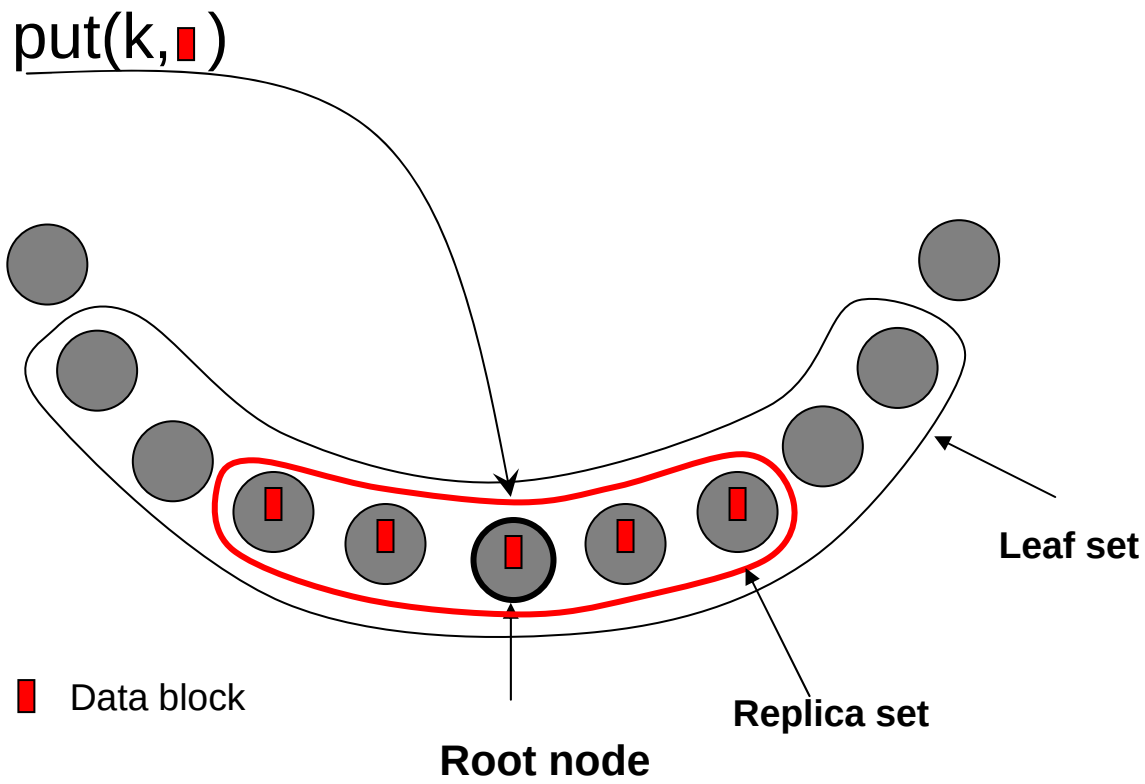
Address space

Insertion of blocks in DHT





Leafset-based replication



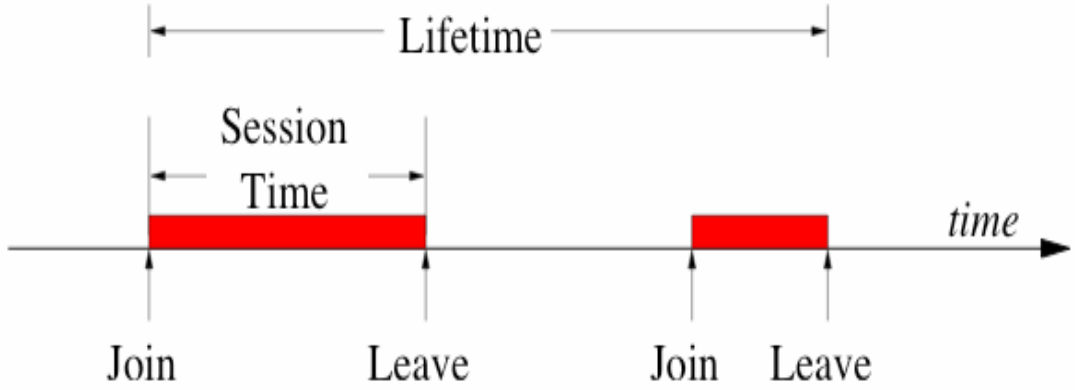
Impact of Churn

Churn = « the continuous process of node arrival and departure »

- Join
- Leave
- Crash

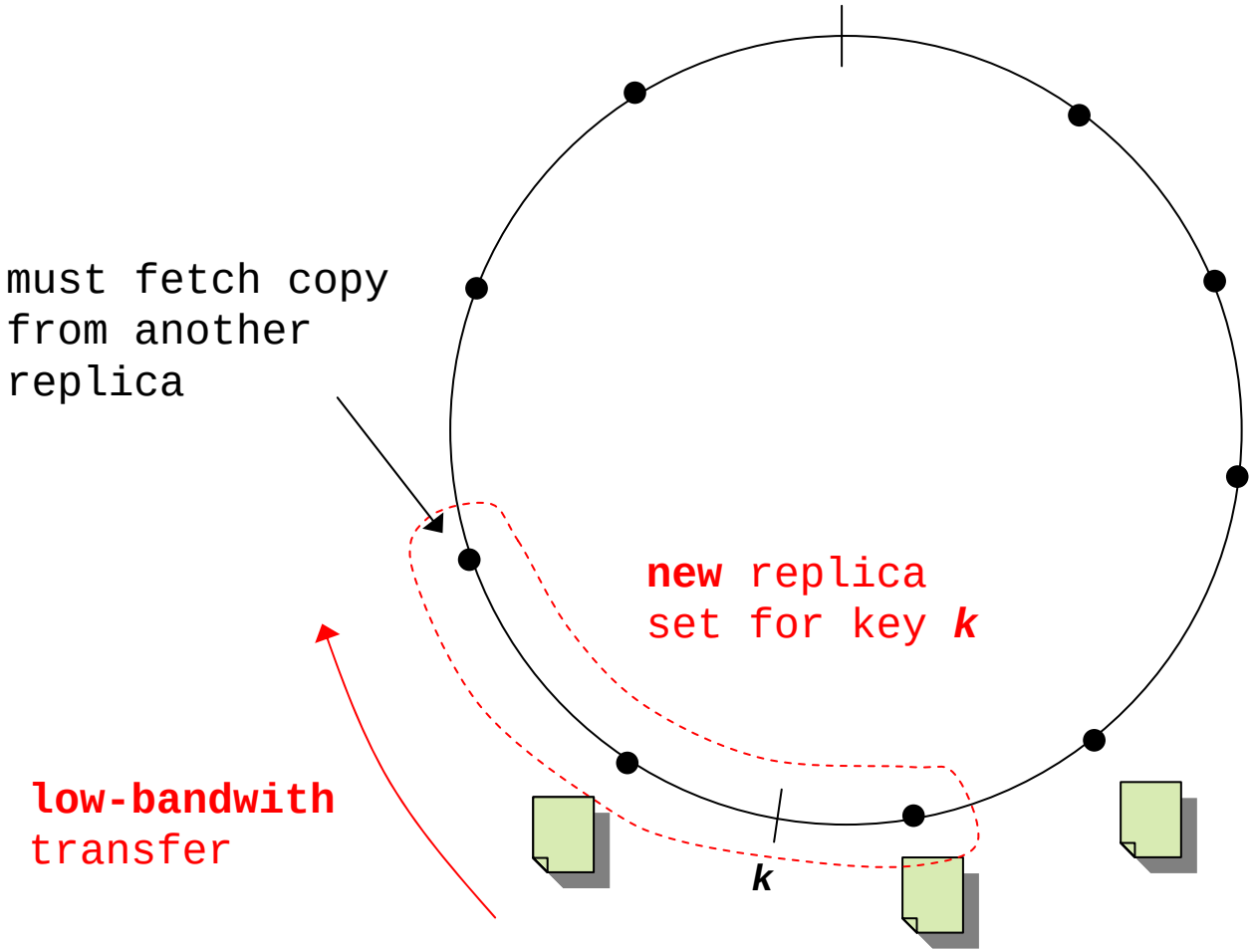
Metrics of Churn

[Rhea 04] Handling churn in DHT, Usenix Annual Technical Conf. 2004

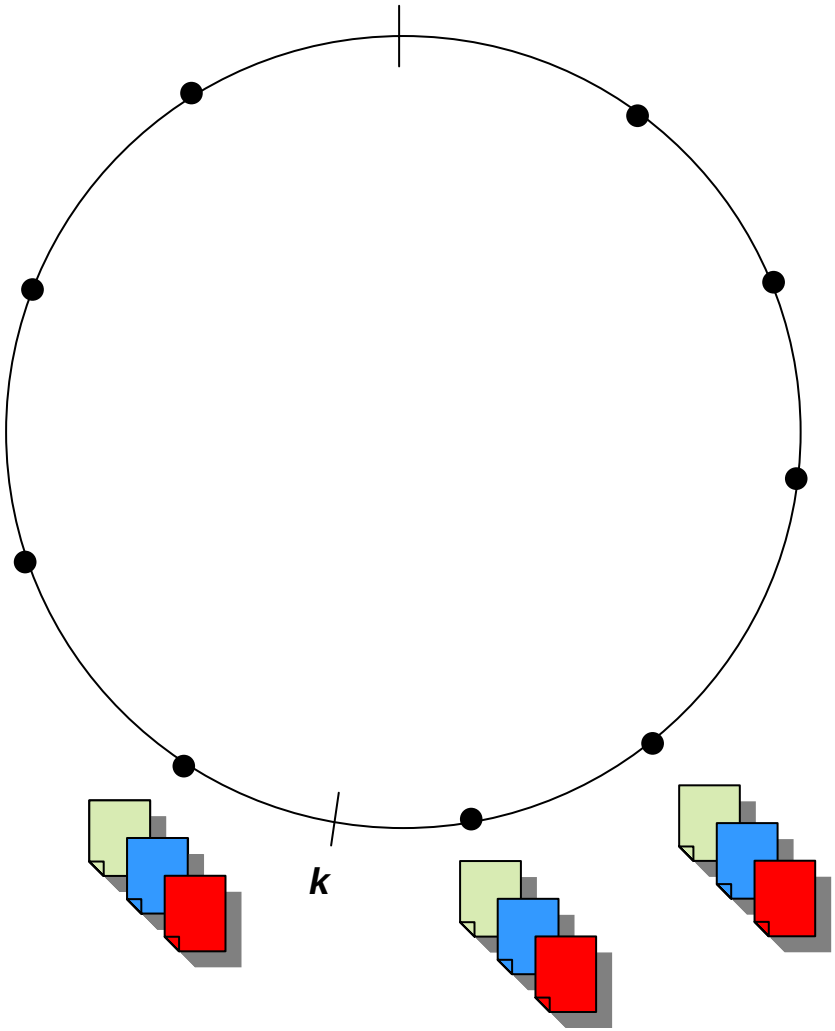


Authors	Systems Observed	Session Time
SGG02	Gnutella, Napster	50% < 60 minutes
CLL02	Gnutella, Napster	31% < 10 minutes
SW02	FastTrack	50% < 1 minute
BSV03	Overnet	50% < 60 minutes
GDS03	Kazaa	50% < 2.4 minutes

Churn in DHTs

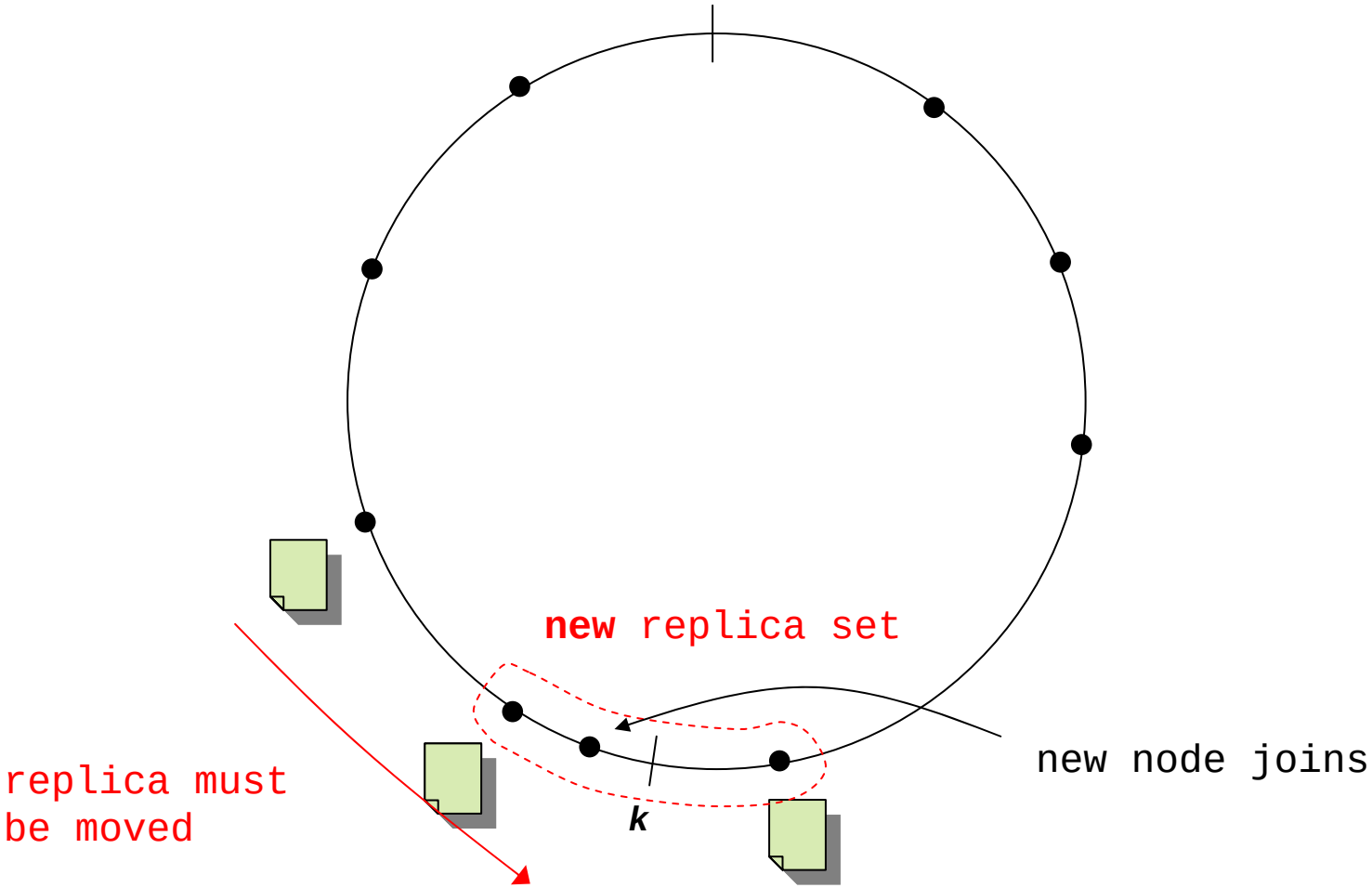


Churn in DHTs

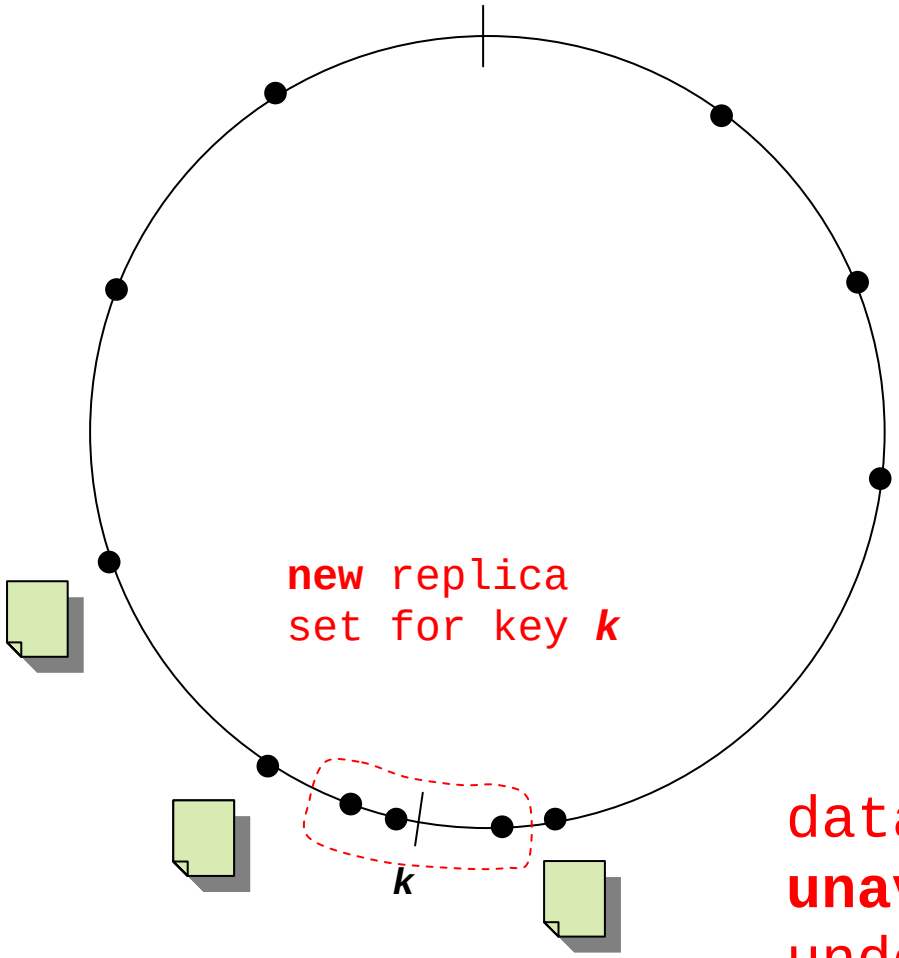


data may be **lost** under high churn

Churn in DHTs



Churn in DHTs



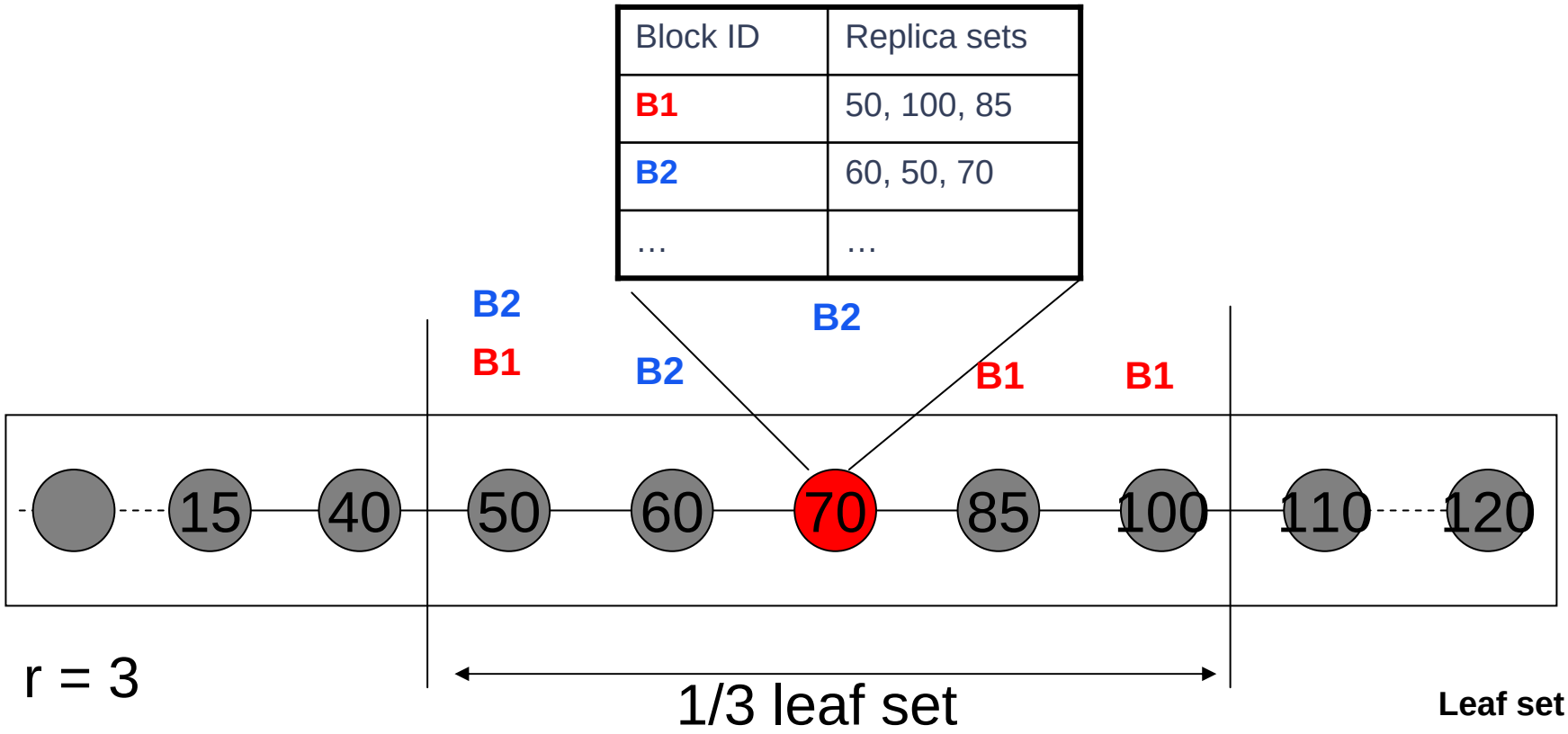
data may be **unavailable** under high churn

A churn resistant protocol

- Main idea : relax the DHT constraints
  - Allow uncontinuous replica-set in the replica set
  - Avoiding useless data movement

An enhanced replication protocol

- Root only stores meta-information of blocks is responsible (replica set)
- Insertion of new block : randomly choses r block in the middle of its leafset (1/3 in the middle)

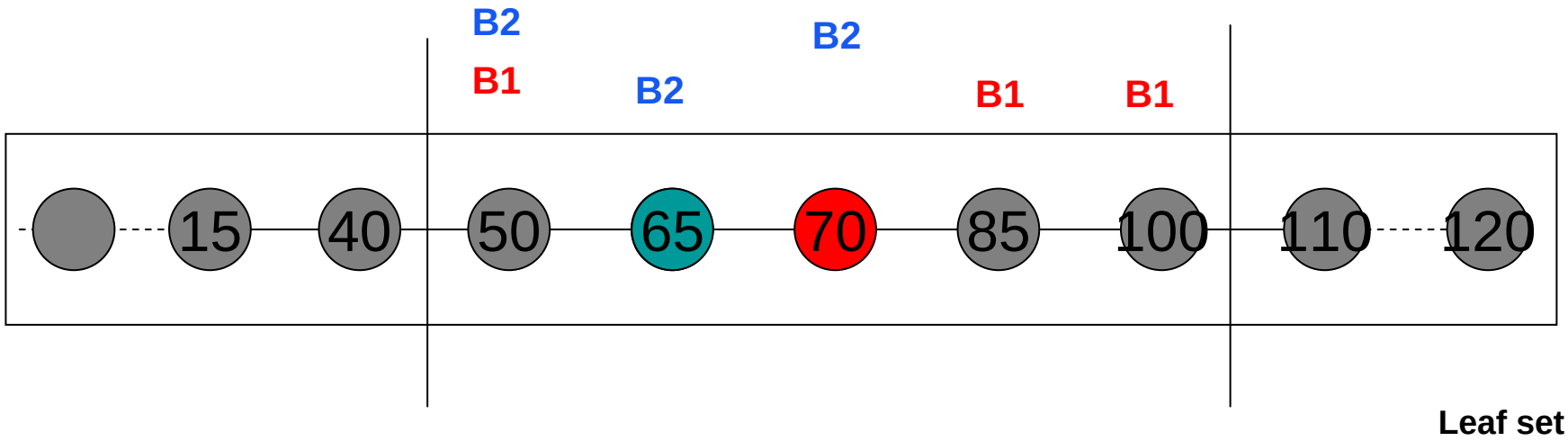


Maintenance protocol

- For each data-block for which the peer p is root :
  - Check if the **r** copies of the replica set are still in the leaf-set.
  - If some copies are missing randomly chose new ones in the middle (1/3) of the leaf-set
- For each data-block for which the peer p is hosting a copy:
  - Check if the known current root is still the current root. if not, notice the new root
  - The new root update its state

Replication protocol with churn

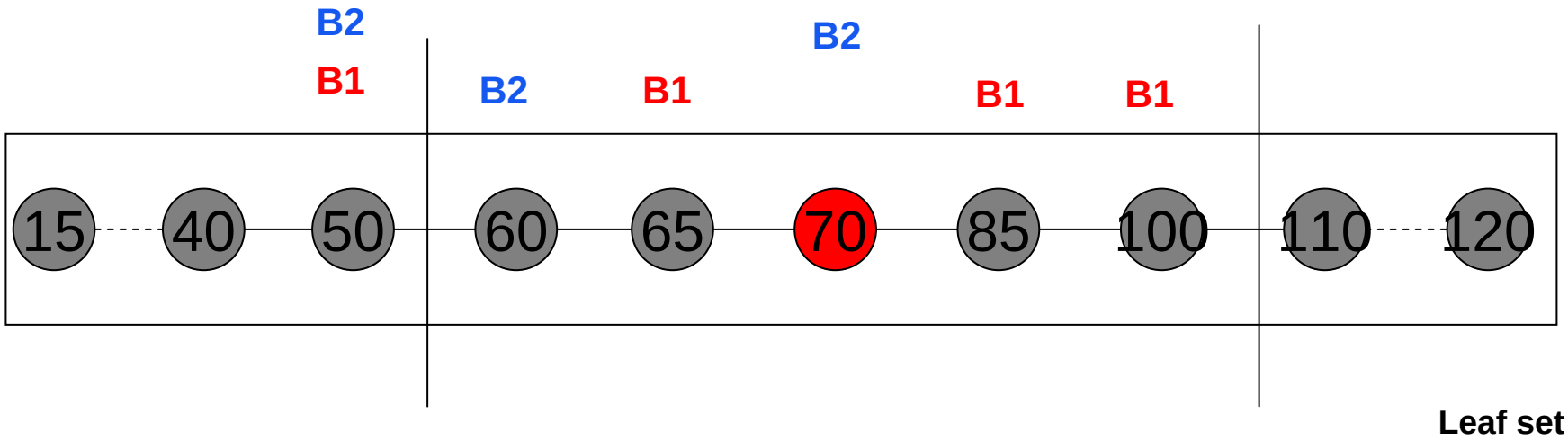
- Arrival of new node **65**



- **No data is moving**

Replication protocol with churn

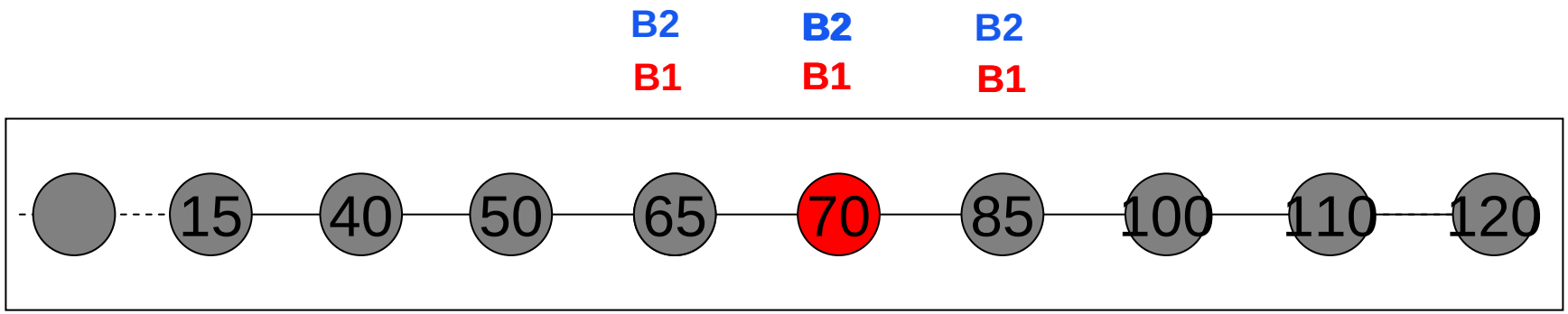
- Departure of node **85**



- **Replication degree is maintain, no useless data movement (only one message B1)**

Comparison with Past replication protocol

- Insertion of 65 : B1 and B2 move
- Leave of 85 : B1 and B2 move



Leaf set

4 blocks moving (Past) vs. 1 block (Enhanced Past)

Performance evaluation

- PeerSim simulation
- Fine grain simulation
- Implementation of Pastry KBR / Past DHT / Enhanced Past (Pasta)

- Comparison of 2 strategies
  - Contiguous placement of data block = **Past**
  - « Free » random placement (mobile position) = **Pasta**

- Preliminary results

Simulation parameters

- Network :
  - 100-peer network (N)
  - ADSL : 1 mbits/s for upload and 10 mbits/s for download
  - Latency between 80 and 120 ms

- DHT :
  - leaf-set size = 24
  - inter-maintenance time of 10 minutes at the DHT level (**replica set**)
  - inter-maintenance time of 1 minute at the KBR level (**leaf set**)
  - 10 000 data-blocks (files) of 10 000 KB
  - Replication degre = 3 (**r**)

Performance evaluation : Churn pattern

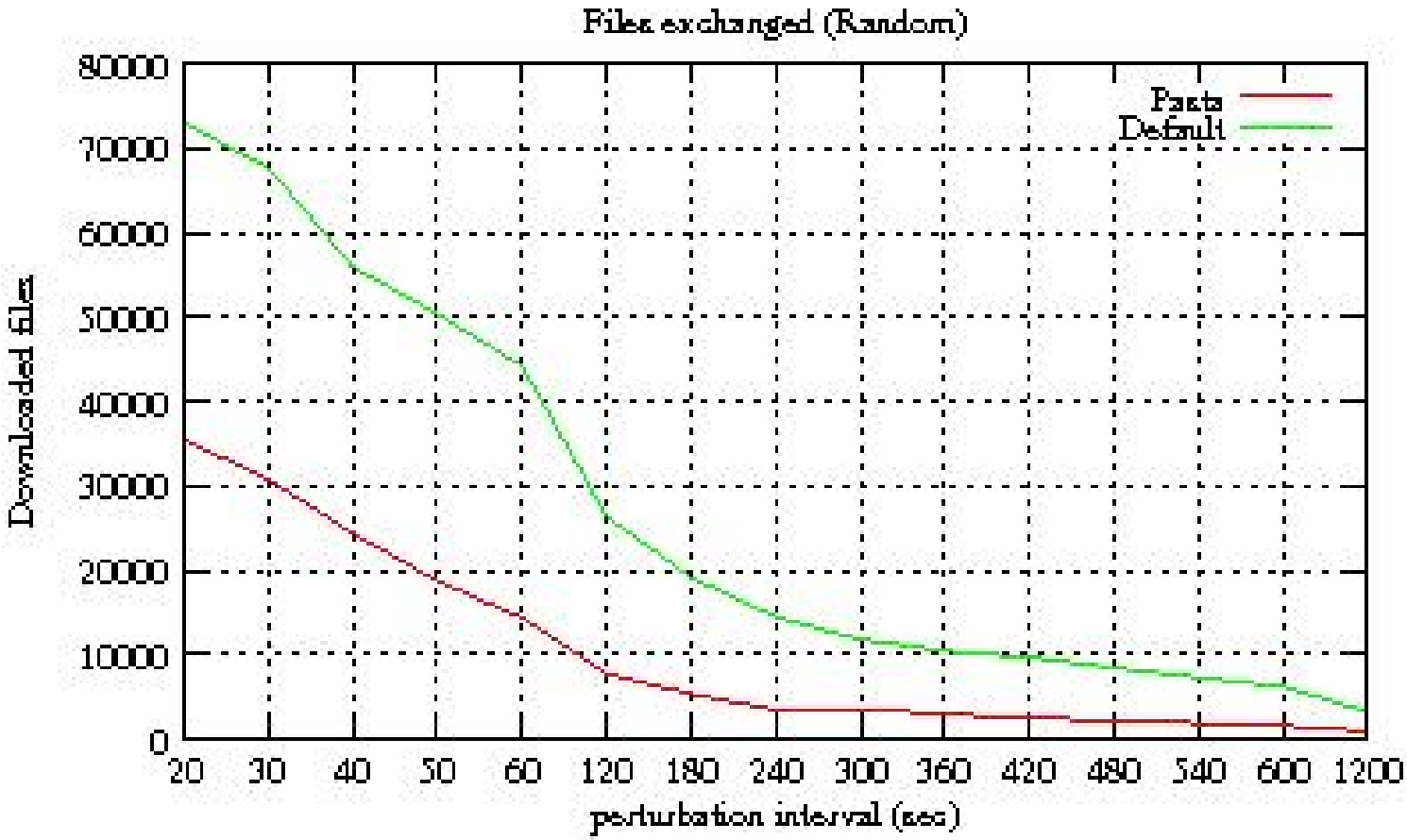
- Periodically insert join and leave

- 2 patterns :
  - One hour churn
  - Continuous churn (snapshot of the system after 5 hours of churn)

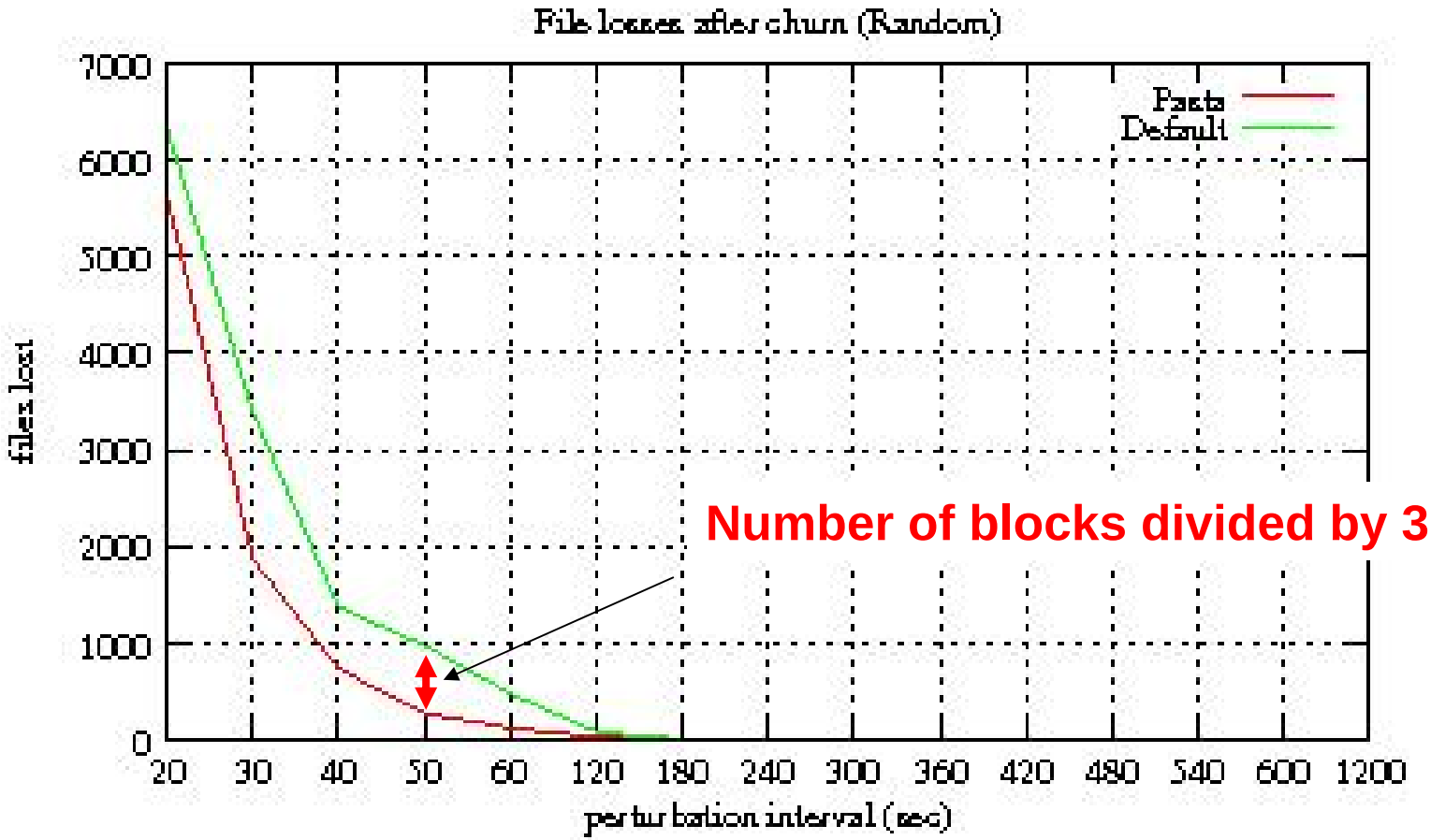
- 3 metrics :
  - Number data-blocks exchanged (bandwidth of the maintenance protocol)
  - Number of data-blocks lost
  - Stabilization time

One hour of churn: number of data-block exchanged

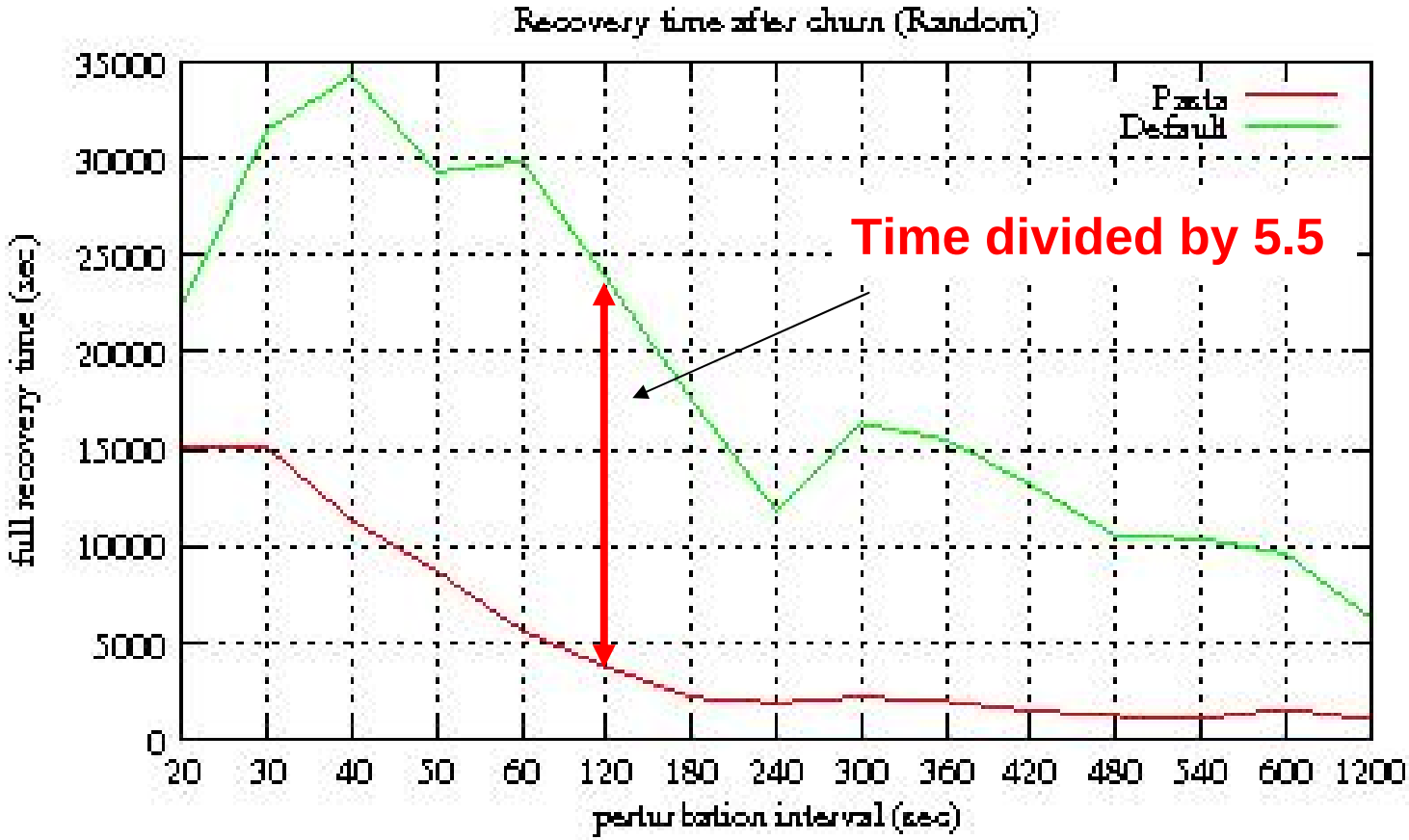
- number of exchanged blocks 2 times lower than in the standard solution



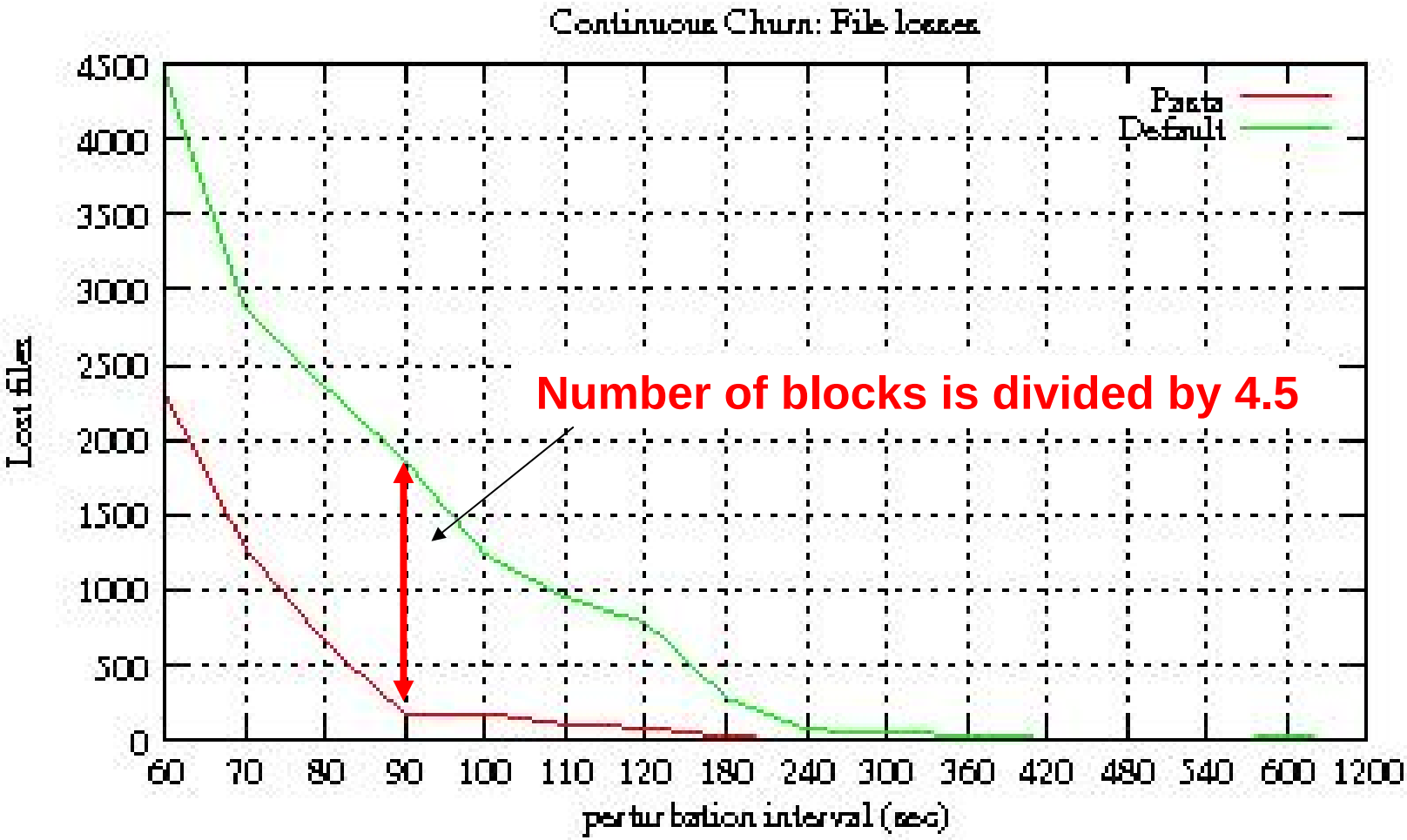
One hour of churn: number of block lost



One hour of churn : Recovery times



Continuous churn : block lost

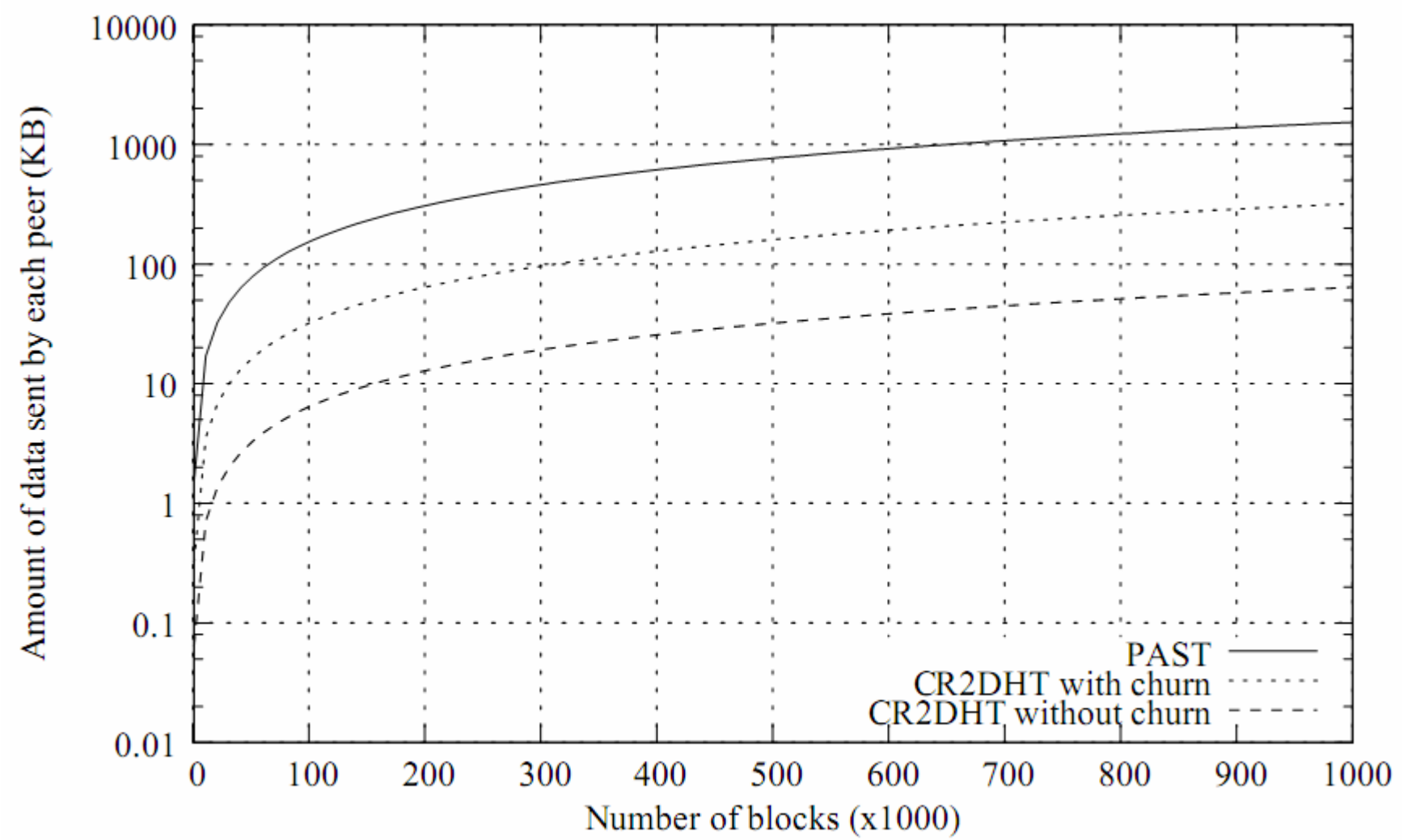


Recovery of a single failure

- Past : 4606 seconds to recover from a single failure

- Pasta : 1889 seconds to recover from a single failure

Maintenance protocol cost



Conclusions

- Current P2P DHT replication strategies does not support high churn levels

- Observations :
  - Many transfers goal is just to maintain location constraints
  - The network is quickly saturated (depends on the network, the amount of stored data and the churn rate)

- Our proposition relaxes location constraints

- Main benefits
  - Node contents are de-correlated => data transfer are more parallelized
    - **A single failure recovery is much faster**
  - Fewer (almost none) useless data transfer
    - **Less network congestion**

Better churn support : less lost data-blocks

Future works

- Placement strategy inside the leafset
  - Monitoring the stability of nodes (modification of leafset maintenance protocol), placement on the most **stable** node
- Study the impact of leafset size
  - Large leaf set (100 nodes)
- Real experimentation
  - Modification of FreePastry
  - Deployment o a distributed architecture