A Distributed and Adaptive Revocation Mechanism for P2P Networks

Thibault Cholez, Isabelle Chrisment et Olivier Festor
{thibault.cholez, isabelle.chrisment, olivier.festor}@loria.fr

LORIA - Campus Scientifique - BP 239 - 54506 Vandœuvre-lès-Nancy Cedex

March 4th 2008

Challenges of P2P networks

P2P network weaknesses : lack of central control and autonomous peer behaviour.
Malicious peer behaviour affects :

- Network security :
  - Peers trying to make attacks (don't respect the protocol).
  - Peers sharing malicious or illegal content (virus, malware).
- Quality of service :
  - Selfish behaviour (70% of users don't share anything, 50% of ressources shared by 1%).
  - Pollution phenomenon (50% of the content).

Problem statement

Our aim : to improve the quality of the network.

- Detect malicious behaviours.
- Revoke them from the network.

Difficulties to design a revocation mechanism :

- How to define a peer's reputation ? (storage, evolution)
- How to do the revocation ? (information, messages)
- How to ensure the mechanism security ?

Concerning the reputation

Classical P2P reputation : each peer stores locally the reputation of others.

- No a priori knowledge of another peer.
- Inefficient for large P2P networks (few peers known, few relationship with each one).

Centralised reputation : eBay.

- Feedbacks of the community create reputation ($\sim$ history).
- Weakness : provided by a central server.

Distributed accounting : PeerMint.

- Each peer has an account stored in the network (DHT).
- Solution with two advantages : global reputation management and adapted to P2P networks.

Concerning the revocation

Acces control system :

- Done by cryptographic mechanisms.
- Group agreement (different thresholds and signatures are possible).
- High cost, bad scalability.

Revocation with suicide :

- Detection and revocation done peer by peer (no consensus).
- A peer which revokes another suicides itself at the same time.
- Advantages : simple, fast, adapted to P2P, safe.
- Weakness : limited application (peers with no individual interest).

Contribution main idea

Pointed weaknesses :

- Revocation : group cryptography, individual action : not adapted.
- Reputation : inefficient mechanisms (no global reputation management).

Principle :

- Reputation of the peers is stored in the DHT (structured P2P network).
- Revocation mechanism based on the reputation (triggered by a threshold).

Studied P2P network : KAD

- Implementation of the Kademlia protocol in eMule and aMule.
- Widely deployed structured P2P network.

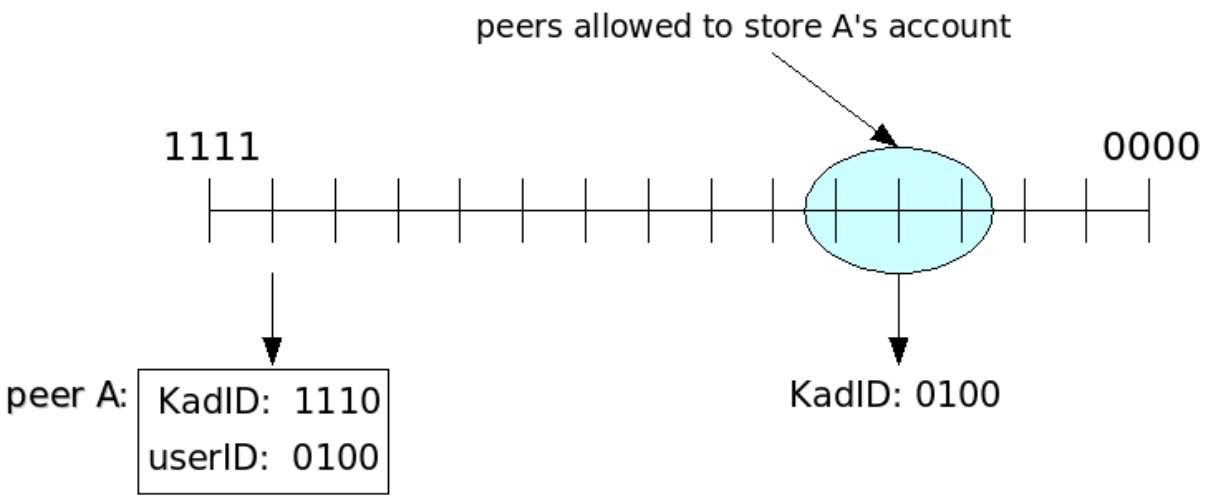
Distributed Accounts

Each peer has two identities (128 bits) :

- The address of the peer in the network (KADID).
- The address of its account in the network (userID).

An account stores public information concerning the peer :

- publicKey (128 bits) : to ensure who is the legitimate owner.
- trustRating (16 bits) : the peer's reputation.
- blackboard (few kBytes) : displays the current transactions of the peer.



Evolution of the reputation

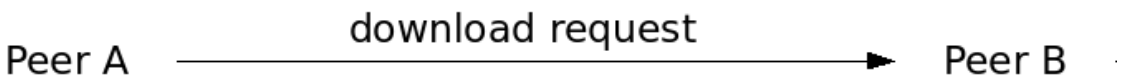
Reputation criteria : the way a peer contributes to the network.
Evolution of the reputation :

- Automatic updates related to peer contribution.
- After a transaction between two peers A and B, both reputations are updated.
- Real update if the transaction is displayed by both peers, with the same amount.

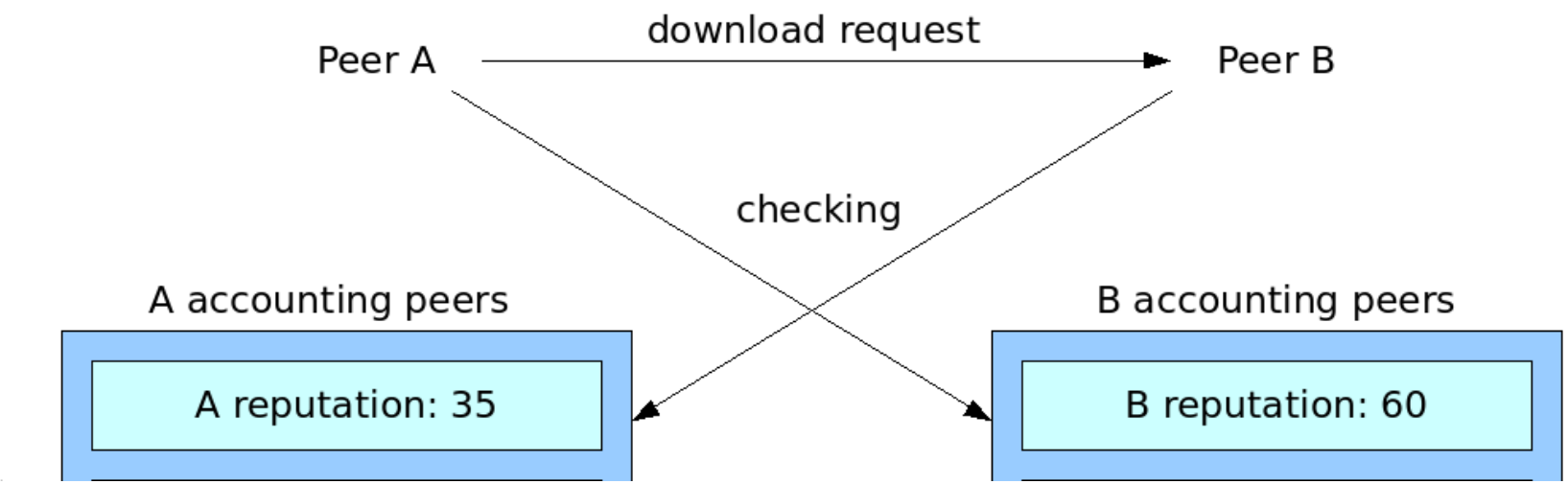
Properties :

- A peer can not directly change its reputation.
- Reciprocal control of both peers.

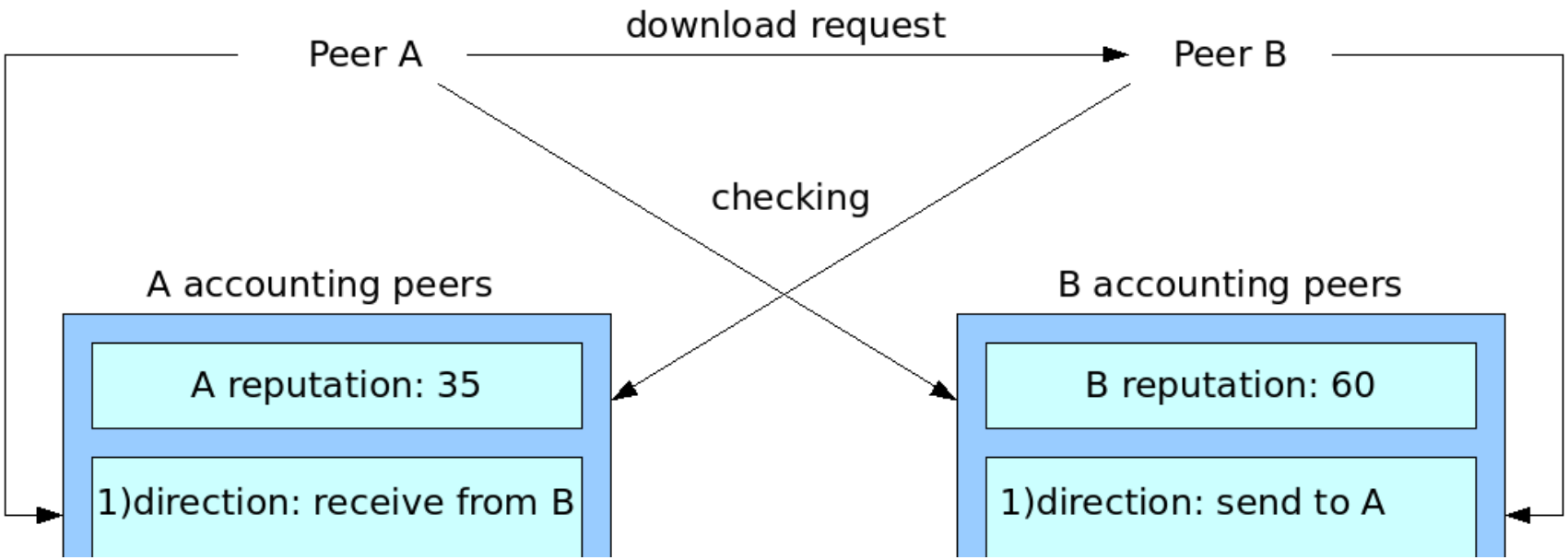
Blackboard usage



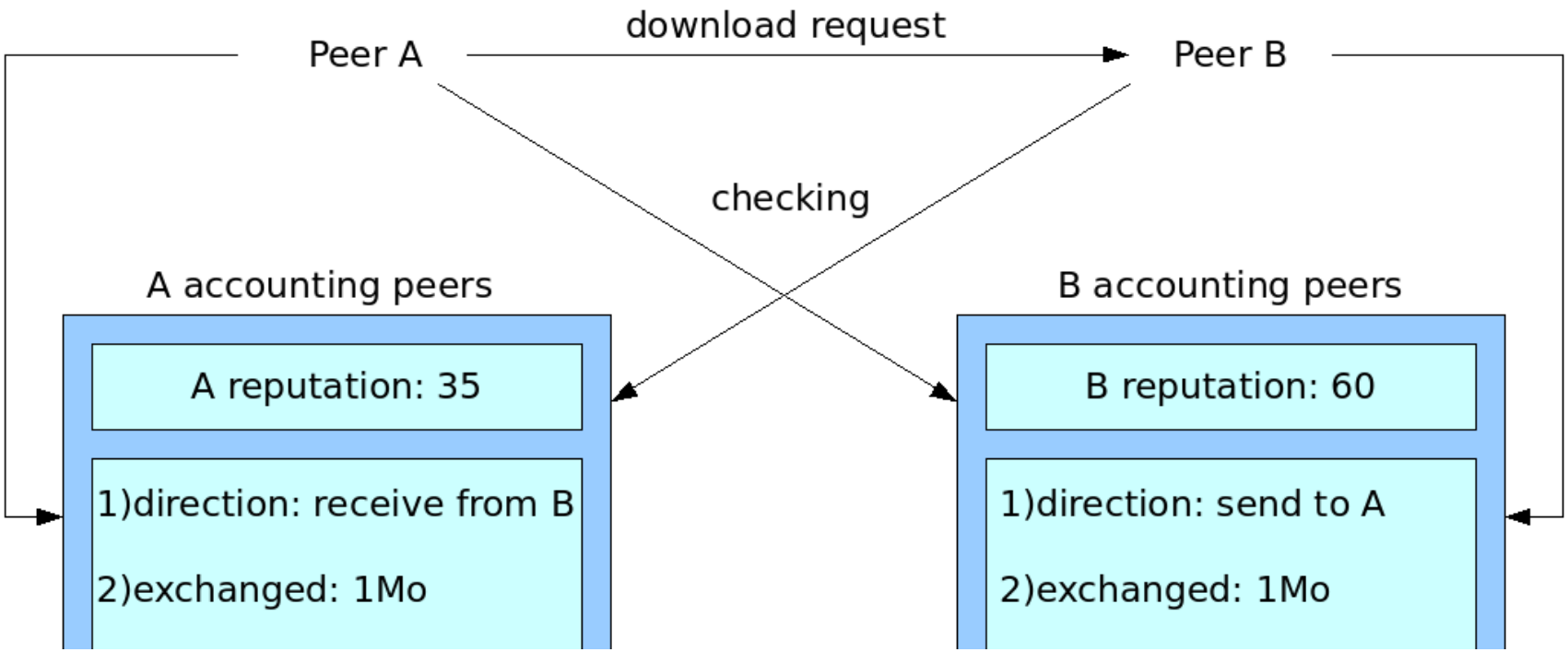
Blackboard usage



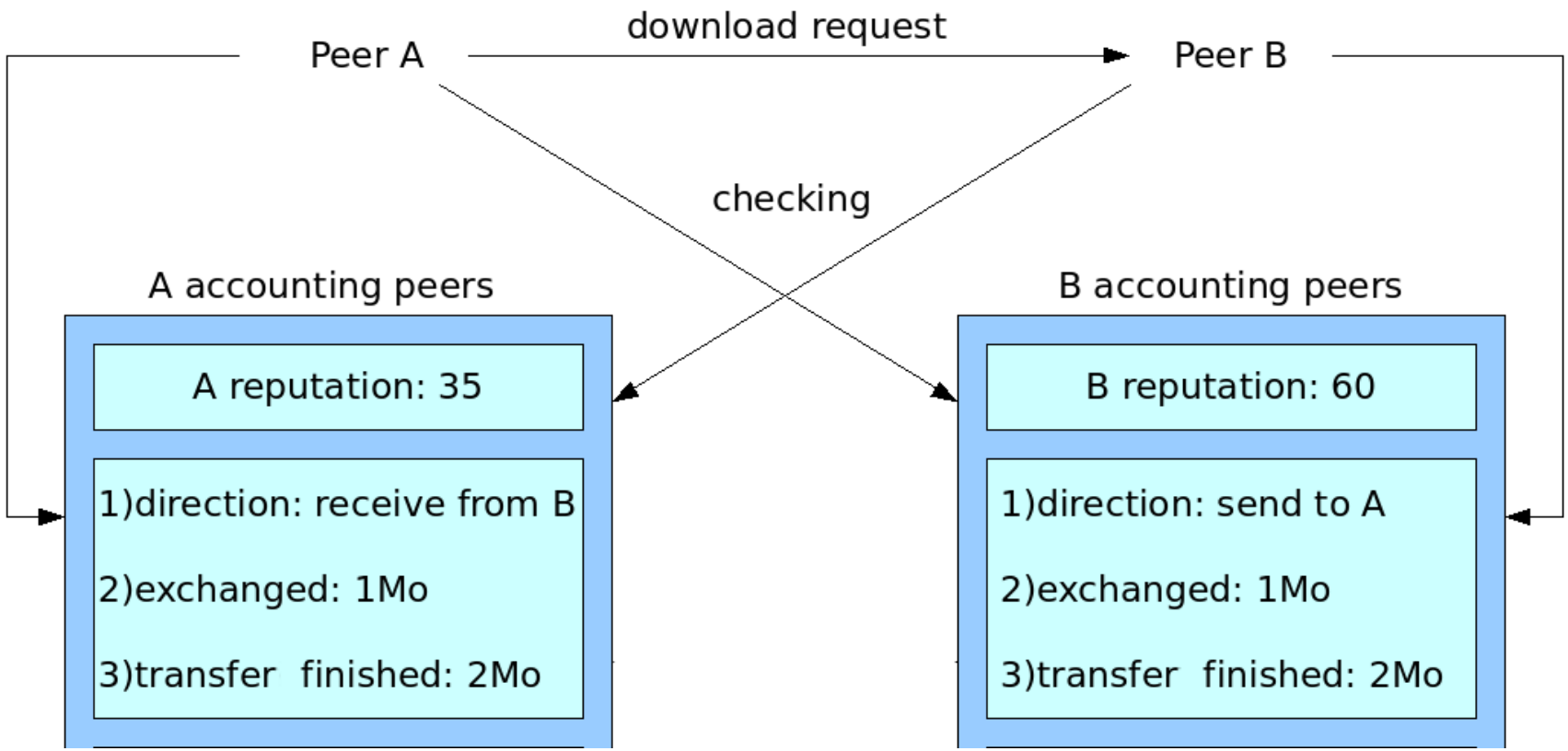
Blackboard usage



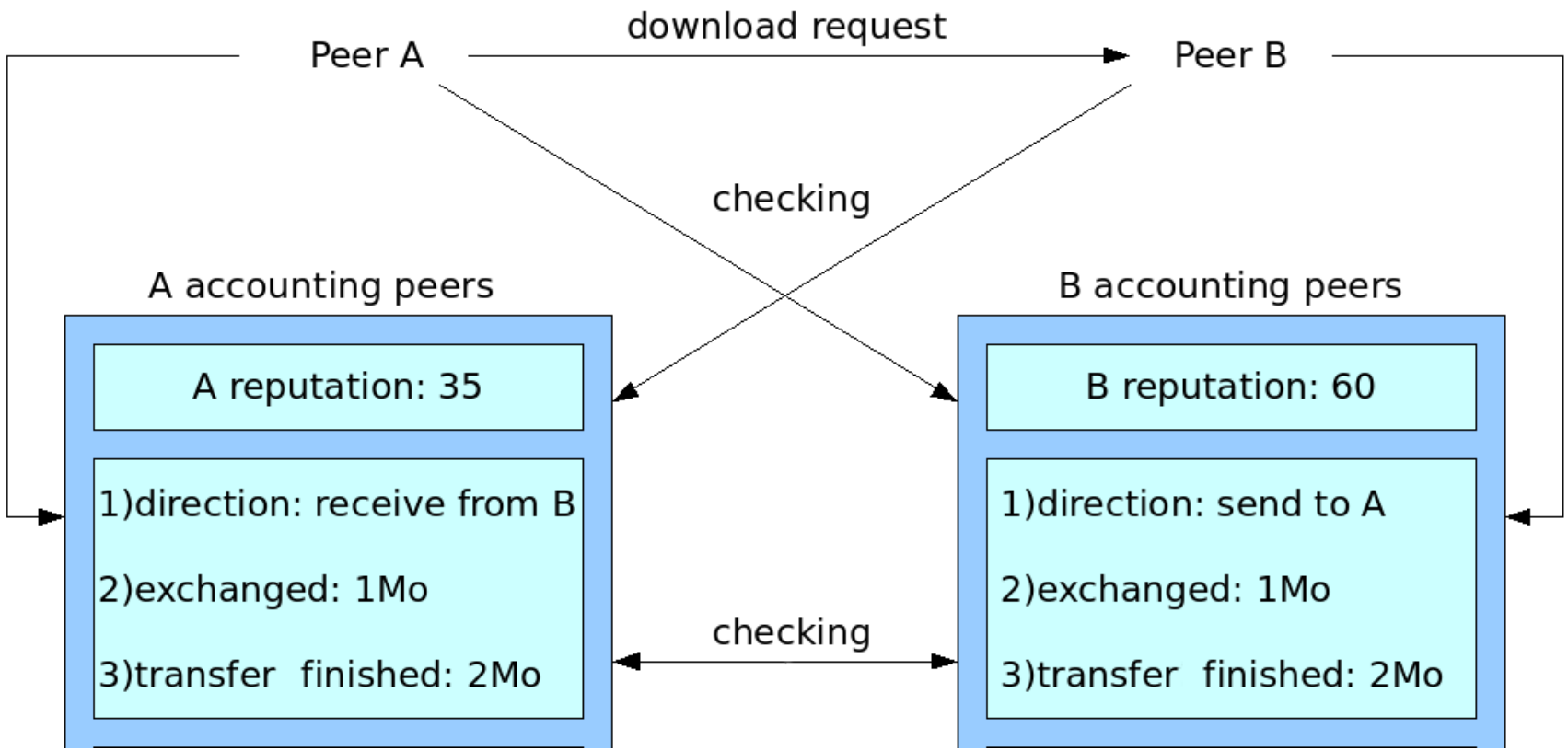
Blackboard usage



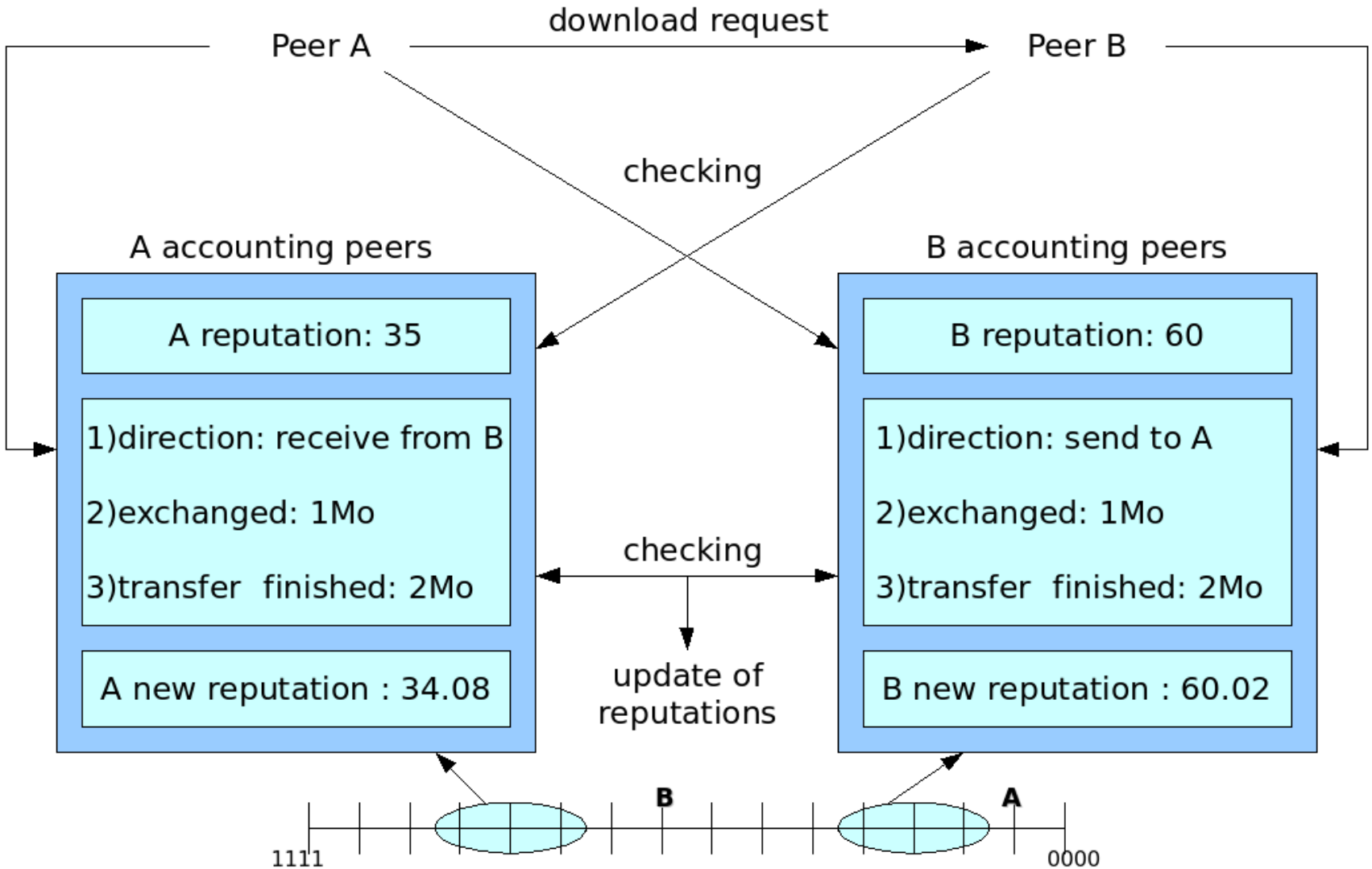
Blackboard usage



Blackboard usage



Blackboard usage



Revocation mechanism

A service-oriented revocation :

- Distributed revocation : each peer must check the reputation before providing services.
- Uses the reputation stored in the network.

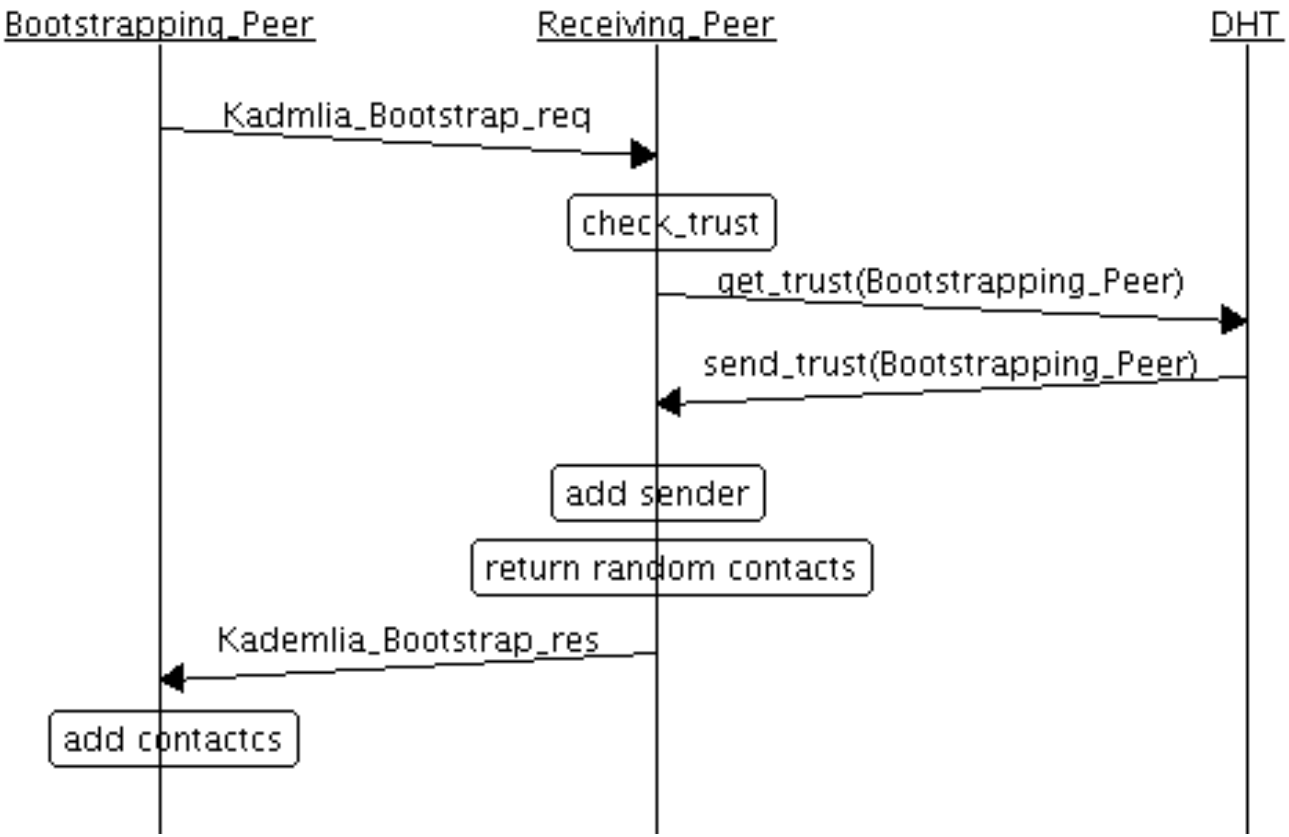
- Revocation inserted in the core of the protocol.
- Adaptive revocation : services are revoked independently according to the reputation criteria.

Revoked Services	Sharing	Security
bootstrap and routing table	No	Yes
publication and upload	No	Yes
download	Yes	Yes
search	No	No

Bootstrap control

First level of revocation $\sim$ acces control :

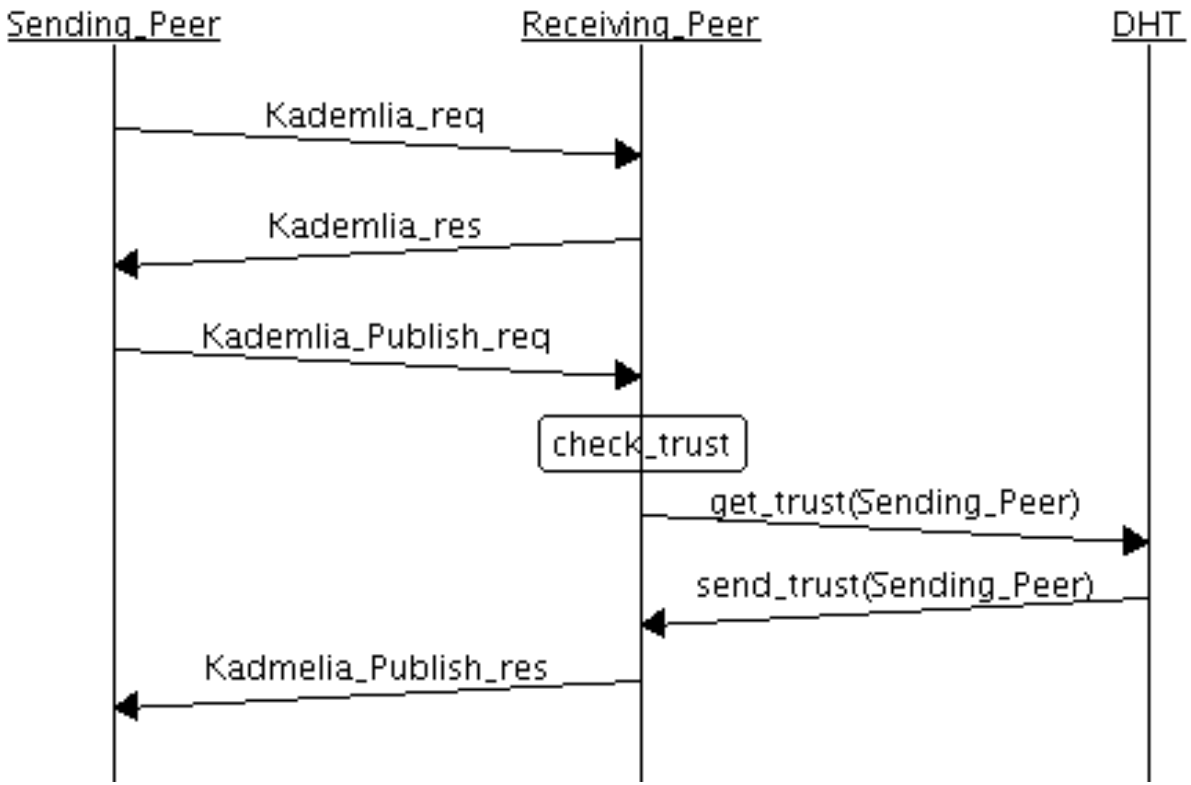
- Reputation checking before sending new contacts.
- Weakness : a malicious peer can share its contact list.



Services control

Services are achieved in the same way :

- 1) Generic Kademlia_REQ's are sent to find contacts in the tolerance zone.
- 2) Service specific requests are sent



Implementation

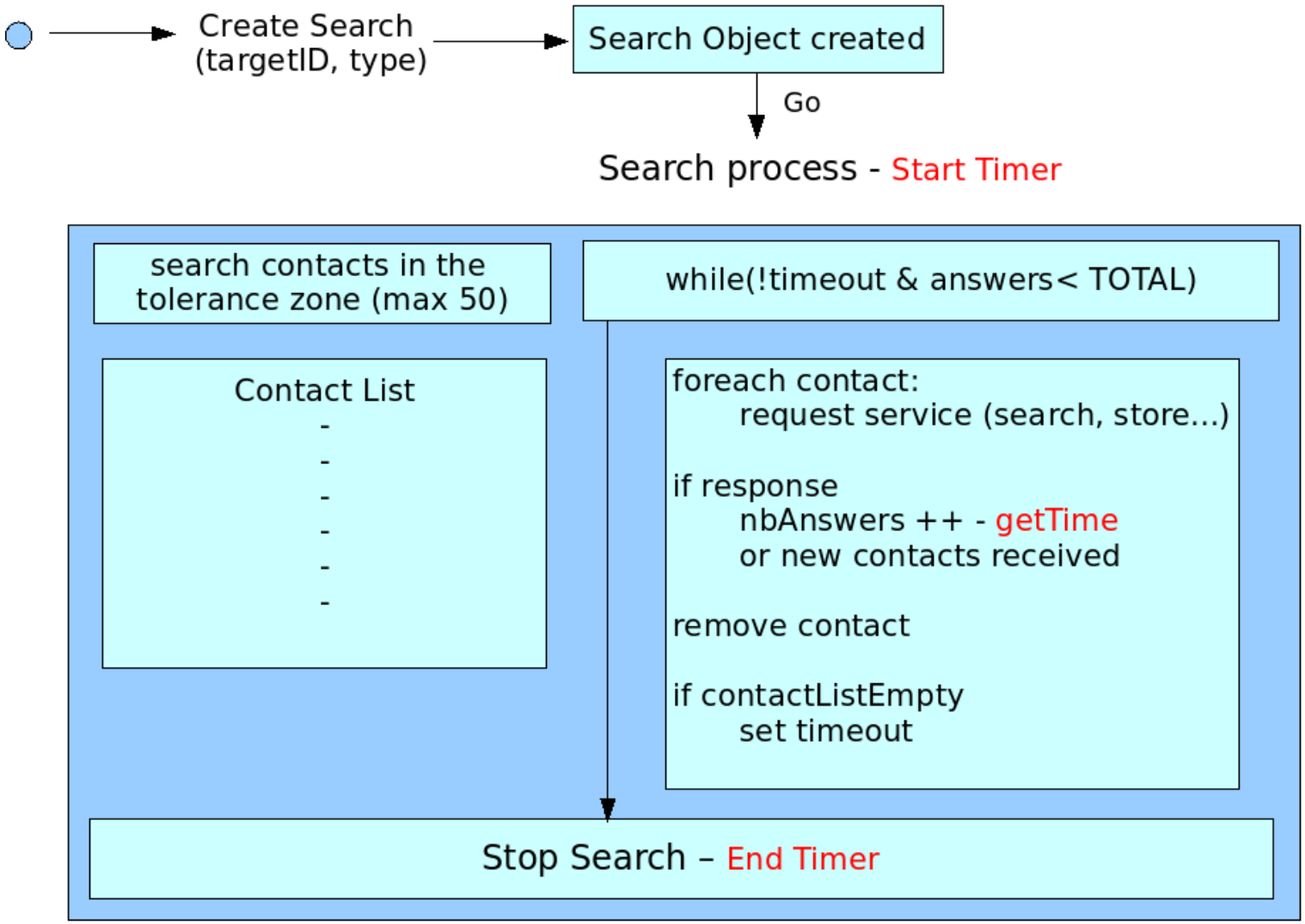
Modification of the KAD client aMule :

- Creation and management of a new kind of information "Account" (data structure, related requests).
- Modification of the class UDPListener : searches and checks the reputation before processing a request.

Delay measurement (reputation finding and storage) in progress on EmanicsLab.

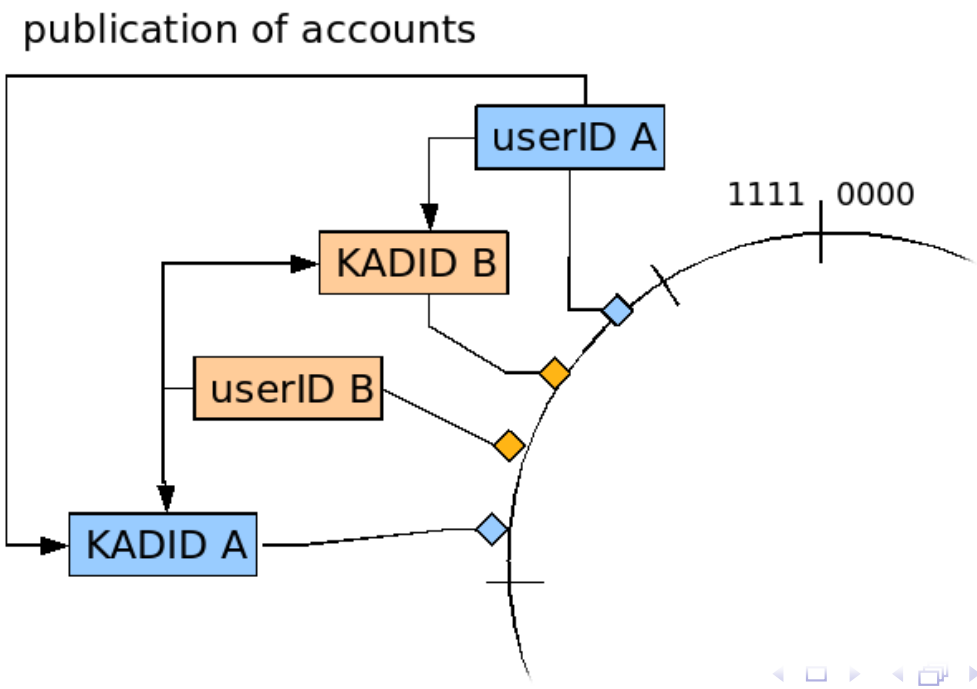
- To evaluate the cost of the mechanism.
- To find a compromise between delay and replication.

Search process and measurement



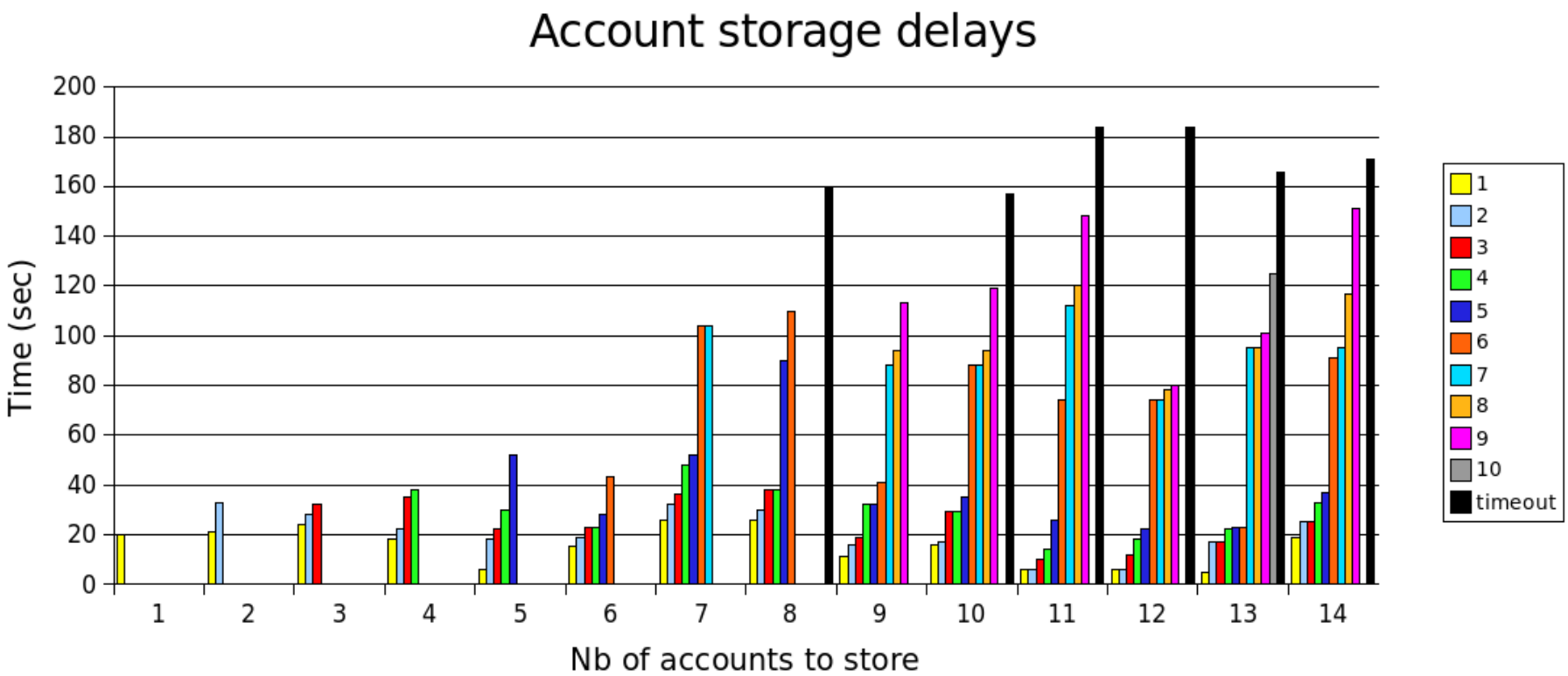
Deployment on EmanicsLab

- One full slice usage : inserting 14 modified clients in KAD.
- Compiling and installing "aMule deamon", "aMule command" and libraries in a static way.
- Deployment scripts :
  - install application on nodes
  - push parameters
  - get results



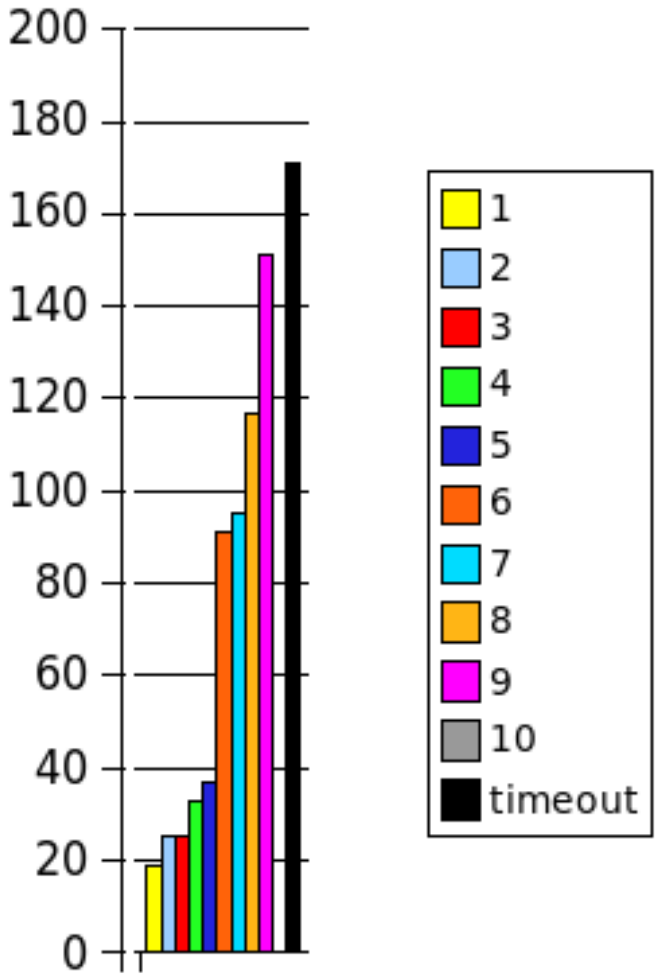
Performances evaluation results

- Trying to store X accounts on 14 possible.
- Simulate a search account process (few peers possible $\neq$ a full tolerance zone).



Performances evaluation results

- Trying to store X accounts on 14 possible.
- Simulate a search account process (few peers possible $\neq$ a full tolerance zone).



Performances evaluation analysis

According to our first experiment on KAD (with standard search parameters) :

- delays propotional to the number of accounts.
- all possible accounts are not found ($\sim 2/3$).
- delays limited by the search time.

Other parameters to study :

- size of the tolerance zone.
- size of the contact list.
- (timeout value).

Delays not sensed by users (no real-time services).

Security issues

After a transfer : modify the information displayed on the blackboard :

- Decreasing the amount of downloaded data : not suported by the protocol.
- Increasing the amount of uploaded data : disagreement between the blackboards.
- Solution : considering the amount displayed by the downloading peer (penalysed if increased).

Malicious peer lying when a reputation is requested :

- No consequence thanks to the replication.
- Majority decision.

Identity changing :

- Allows to retrieve a new reputation.
- Identity crysis : no perfect solution.

Security issues

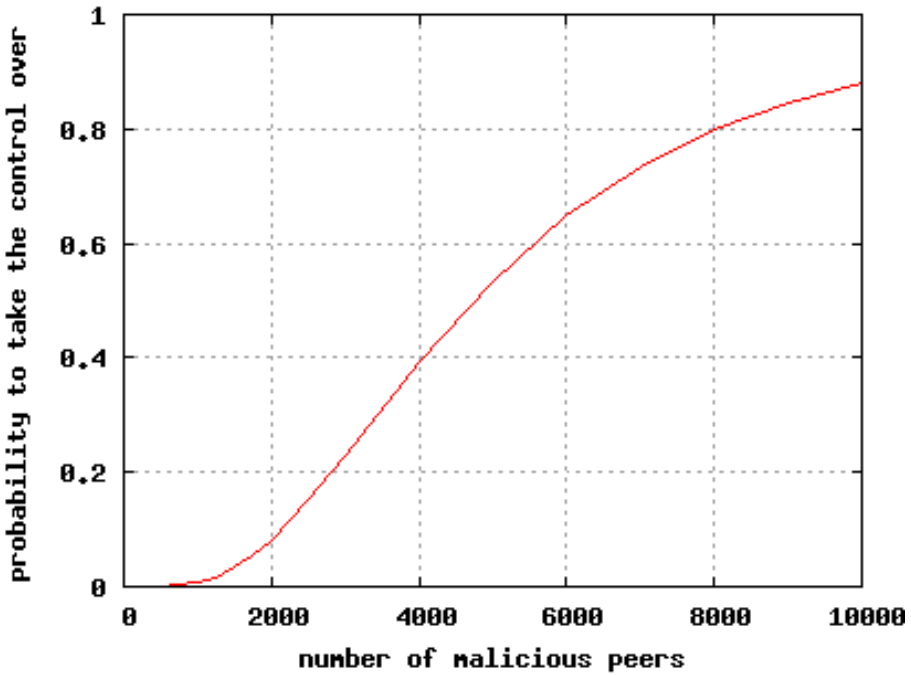
Malicious peers coalition : manage to take the control of at least $n/2+1$ replicated accounts of the target.

- Sybil attack : insertion of many fake peers to take the control over a part of the network.
- Allows victim revokation by the entire network.
- What is the probability of a successful attack ?

$$P(X=i) = \frac{C_x^i * C_{4000}^{10-i}}{C_{4000+x}^{10}} \quad (1)$$

$$P(X \geqslant 6) = \sum_{i=6}^{i \leq 10} P(X=i) \quad (2)$$

Security issues



KAD implementation is insufficient : large Sybil attacks are possible (2^{16}). How to secure the peer's ID :

- Central authority delivering KadIDs.
- Keypeer : distributed key delivering.

Conclusion and future works

In summary :

- Overall reputation mechanism, based on distributed accounts, 1st criteria : contribution of a peer.
- Revocation mechanism service-oriented, distributed, adaptive.
- Design, implementation and experimentation on KAD.
- Safe with a strong peer ID.

Current work : continuing performance evaluation on EmanicsLab.
Future work :

- New criteria : evaluate the quality of the shared content.
- Prevent and detect attacks to the mechanism.