

# BP for $k$ -partitioning and Local Churn Estimation on Structured Overlay Networks

**Boris Mejías** Danny Bickson

`boris.mejias@uclouvain.be`

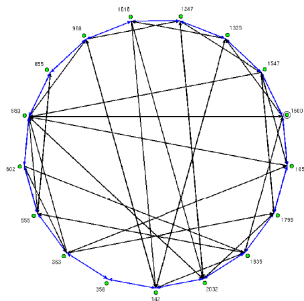
Université catholique de Louvain (UCL)

## Menu of the day

- ▶ Very short intro to P2PS and P2PKit
- ▶ Using BP for  $k$ -partitioning
- ▶ Demo (up to Murphy)
- ▶ Churn Estimation (*Work in progress*)

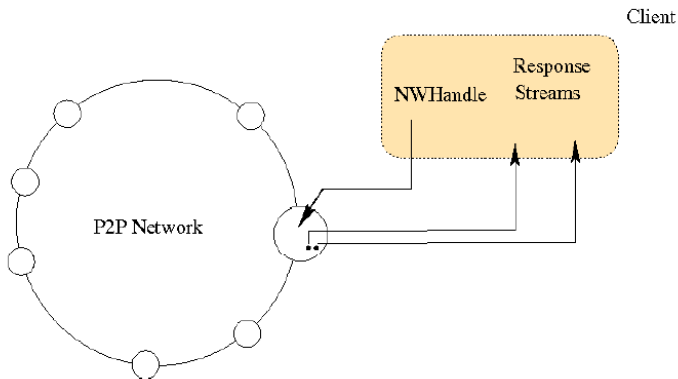
## P2PS - Structured peer-to-peer network

- ▶ Self organizing Chord-alike structured overlay network organized by successor, predecessor and finger-table
- ▶ Currently working on P2PSv3 with Relaxed-Ring structure for fault tolerance to link and process failures



## P2PKit - Framework for P2P Applications

- ▶ Clients connect to existing peer in the network
- ▶ Service oriented architecture
- ▶ Dynamic installation/upgrading of services



## Using Belief Propagation for $k$ -partitioning

- ▶ Based on “*Indexing data-oriented overlay networks using belief propagation*” by Danny Bickson *et al.*
- ▶ Partition of the network in  $k$  sets for load-balanced data retrieving
- ▶ A node cannot be in the same set as its neighbour
- ▶ Edge potentials: matrix  $k \times k$  with 0's in the diagonal
- ▶ Each belief is a vector (size  $k$ ) obtained from

$$bel_i^{(t+1)}(x_i) = \alpha P_i(x_i) \prod_{j \in N(i)} \max_{x_j} P_{ji}(x_j, x_i) bel_j^{(t)}(x_j)$$

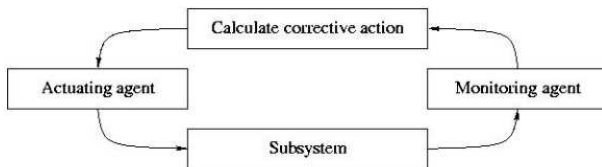
## Using Belief Propagation for $k$ -partitioning

- ▶ Adapted to network topology of P2PS resulting in a directed graph
- ▶  $N(i)$  determined by successor, predecessor and fingers, and thus, the amount of neighbours is dynamic
- ▶ We decided to use a timeout to finish every iteration step
- ▶ Implemented as a P2PKit service

# Time to defy Murphy's law

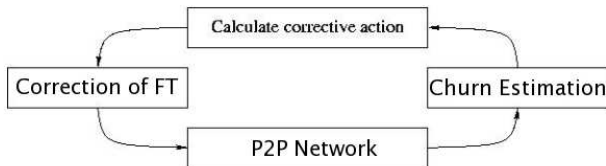
## In the context of Self-managing systems

- ▶ Software can be built based on feedback loops for self-adaptability



## Churn Estimation

- ▶ Ratio to measure nodes joining and leaving the network
- ▶ Local Churn Estimation as monitoring component to trigger correction of finger tables



## Churn estimation

- ▶ Successors list is maintain for fault tolerance
- ▶ Every join/leave/crash triggers an update of the successors list. This is the first source for local churn estimation
- ▶ Propagation of the belief to pred, succ and fingers.
- ▶ Based on “*Peer-to-Peer Rating*”, the goal is to minimize the cost function for churn estimation define as

$$\min_x \left[ \sum_i (x_i - y_i)^2 + \beta \sum_{j \in N(i)} w_{ij} (x_i - y_j)^2 \right]$$

$$\text{with } w_{ij} = \frac{1}{\text{dist}(i,j)}$$

## Churn Estimation

- ▶ Equivalent to

$$\max_x [e^{-(\sum_i (x_i - y_i)^2 + \beta \sum_{j \in N(i)} w_{ij} (x_i - y_j)^2)}]$$

- ▶ Apply Consensus Propagation (based on Gaussian BP) in order to solve the minimization of the cost function
- ▶ Input consist on a set  $W$  defining weight relationship, and the initial churn estimation  $x_0$
- ▶ Rest of the story to be learned during this workshop