

# **Principles of DHTs for P2P Systems**

Pierre Fraigniaud

Master MPRI

07/10/06

<http://www.lri.fr/~pierre>

# SUMMARY

- Overlay networks for P2P systems
  - Semi-decentralized systems
  - Decentralized systems
    - Non-structured networks
    - Structured networks

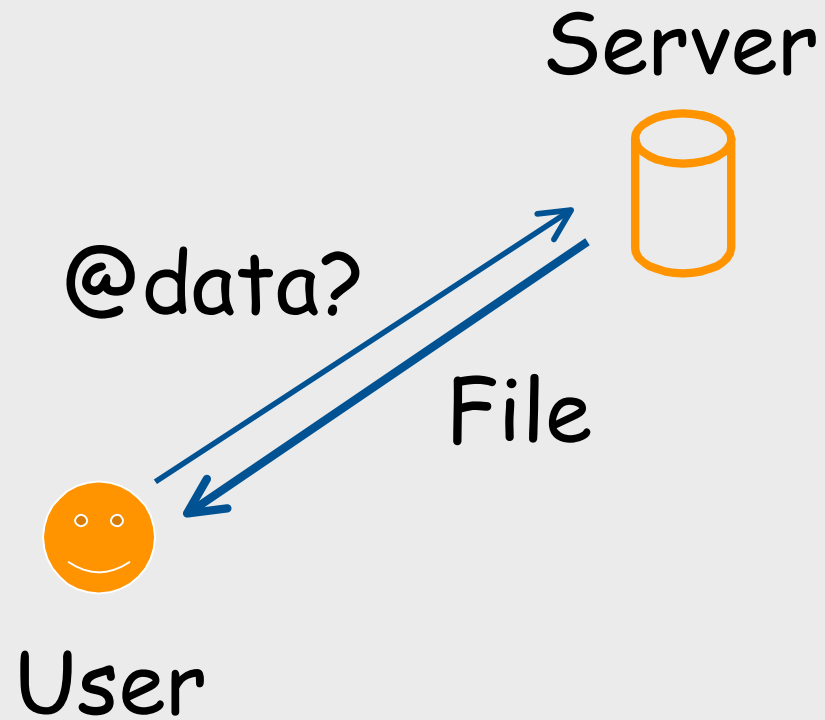
# PEER-TO-PEER (P2P) PARADIGM

- Opposed to the master-slave paradigm
- A group of users share a common space in a decentralized manner, all playing the same role
- Objectives:
  - Share data (music, movies, etc.)
  - Share resources (computing facilities)
- Functionalities:
  - Publish
  - Search

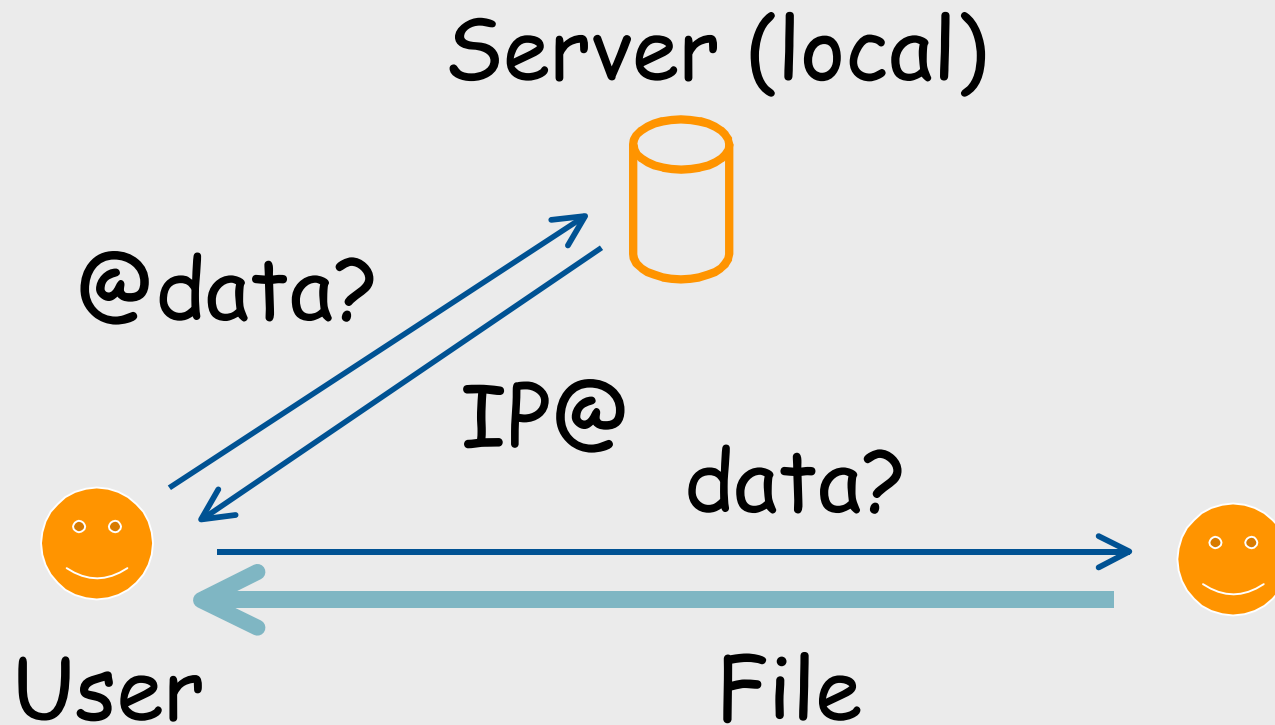
# MAIN (IDEAL) CHARACTERISTICS

- No central server
- Self organization
- Users can join and leave the system at any time
- Fault-tolerance
- Anonymity (?)

# CLIENT-SERVER



# SEMI-DECENTRALIZED SYSTEMS



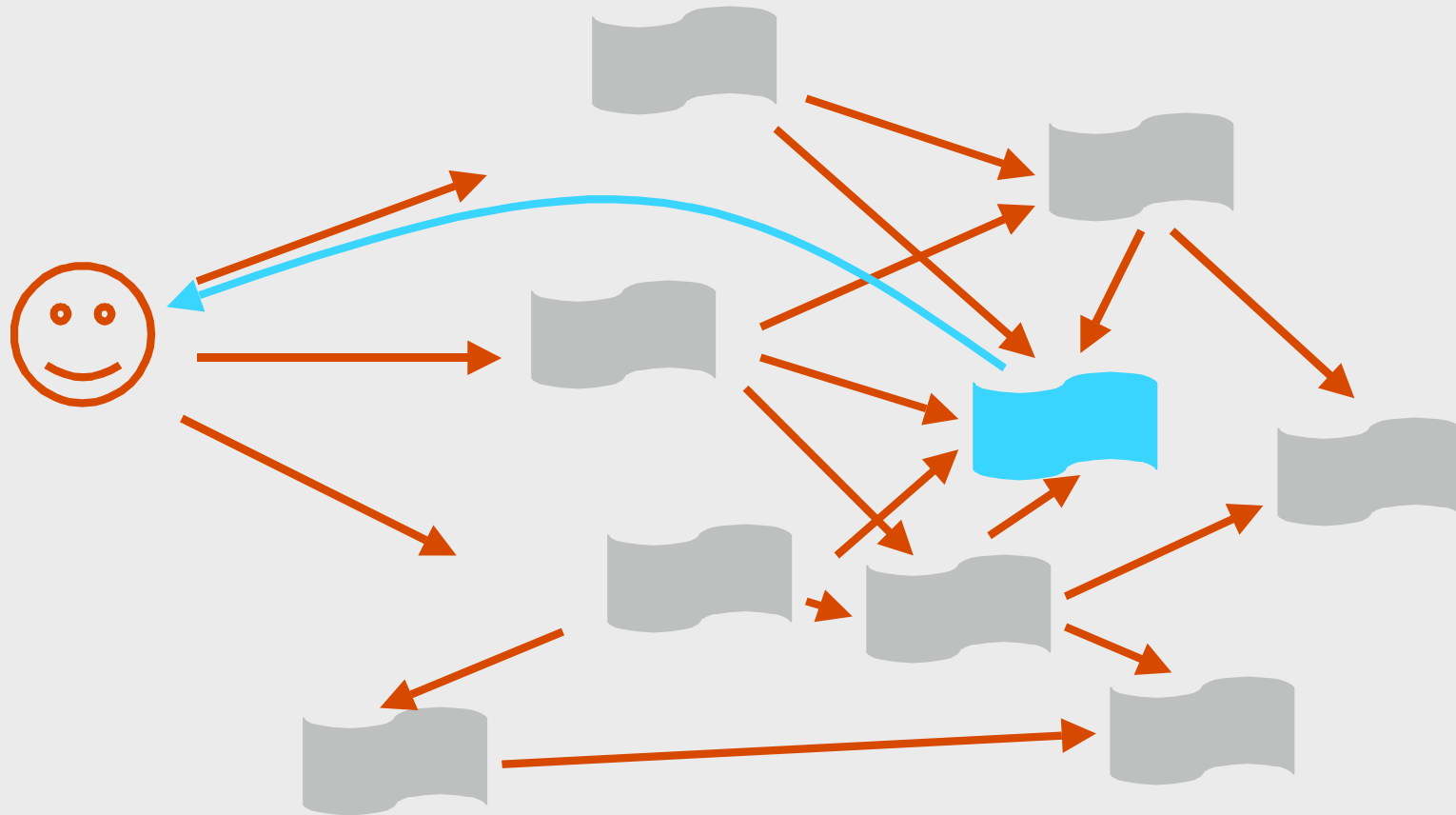
# DISCUSSION

- Pro:
  - Quick access
  - Enable sophisticated types of requests
- Con:
  - Expensive (computation and storage)
  - Bottleneck (high congestion)
  - Single point of failure

# DECENTRALIZED SYSTEMS

- Nodes are connected by a logical overlay network, deployed over the Internet
- Link  $(u,v)$  means  $u$  knows the IP@ of  $v$
- Structure of the overlay:
  - Non structured:  
nodes connect to arbitrary nodes
  - Structured:  
nodes are connected to specific nodes  
for maintaining a specific topology

# NON STRUCTURED NETWORKS



# DISCUSSION

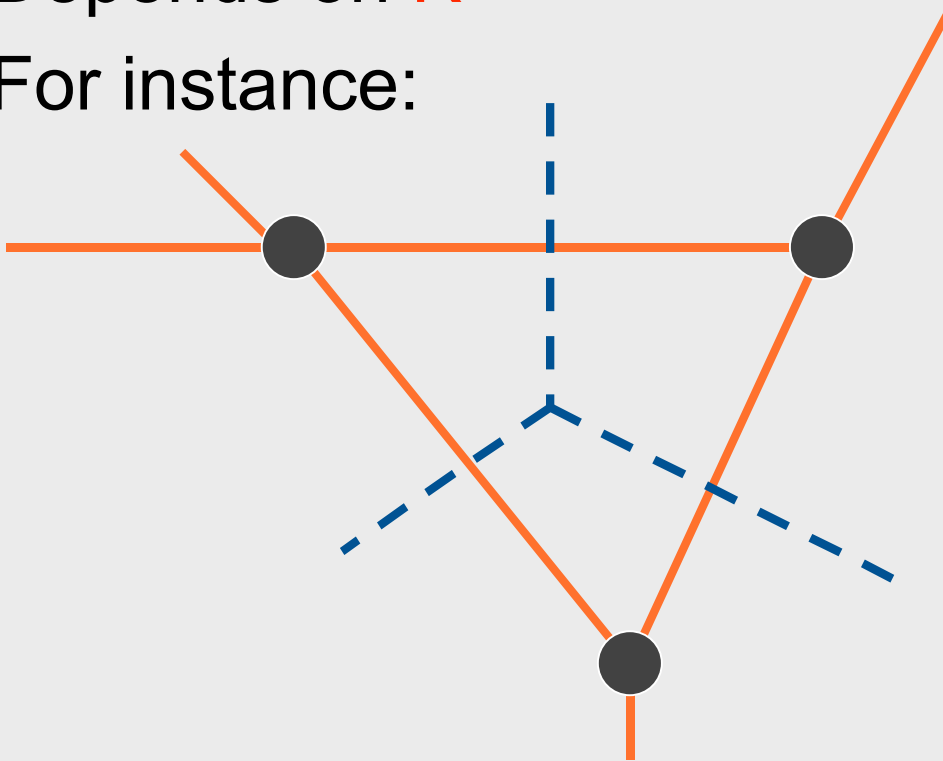
- Pro:
  - Easy to implement, and cheap!
- Con:
  - High traffic load (if flooding)
  - Non exhaustive (if search is bounded)
  - Routing is hazardous

# STRUCTURED NETWORKS

- Principles:
  - Let  $K$  be a metric space (e.g.,  $[0, 1[$ )
  - Assign a label to every node
$$\text{label} : \{ \text{IP@} \} \rightarrow K$$
  - Assign a key to every resource
$$\text{key} : \{ \text{resources} \} \rightarrow K$$
  - The resource  $r$  is published at the node  $u$  such that  $\text{dist}(\text{label}(u), \text{key}(r))$  is minimal.

# PRINCIPLES (CONT)

- Connections:
  - Depends on  $K$
  - For instance:



# ROUTING

- Key-based routing (Content Addressable Networks)
- Greedy routing to  $\kappa$  at current node  $u$ :
  - $N(u) = \{ \text{neighbors of } u \}$
  - Let  $v$  be a node such that:
$$\text{dist}(\text{label}(v), \kappa) = \min_{w \in N(u)} \text{dist}(\text{label}(w), \kappa)$$
  - Route to  $v$

# RESOURCE PUBLICATION

- Node  $u$  aims at publishing resource  $r$ 
  - Node  $u$  computes  $\kappa = \text{key}(r)$
  - Node  $u$  informs the node  $v$  in charge of  $\kappa$  that it is storing  $r$
  - Node  $v$  stores the IP@ of  $u$  in its lookup table
- The second phase is based on the routing procedure

# SEARCHING FOR RESOURCES

- Node **u** search for resource **r**
  - Node **u** computes  $\kappa = \text{key}(r)$
  - Node **u** contacts the node **v** where  $\kappa$  is published to get the IP@s of all nodes storing **r**
  - Node **v** sends these IP@s to **u**
- The second phase is based on the routing procedure

# DYNAMICS

- Node **u** leaves:
  - Reallocation of keys to **u**'s neighbors
  - Update connections between **u**'s neighbors
- Node **u** joins:
  - Connection to an entry point
  - Label computation (hash function:  
 $\text{label}(u) = \text{hash}(\text{IP}@(u))$ )
  - Setting of **u**'s connections
  - Reallocation of keys from **u**'s neighbors

# DISCUSSION

- Structured networks are based on the Distributed Hash Table (DHT) paradigm
- Pro:
  - Fully distributed
  - Low traffic and load balancing
  - Exhaustive search
- Con:
  - The dynamics is a bit complex
  - Basic requests (key-based)

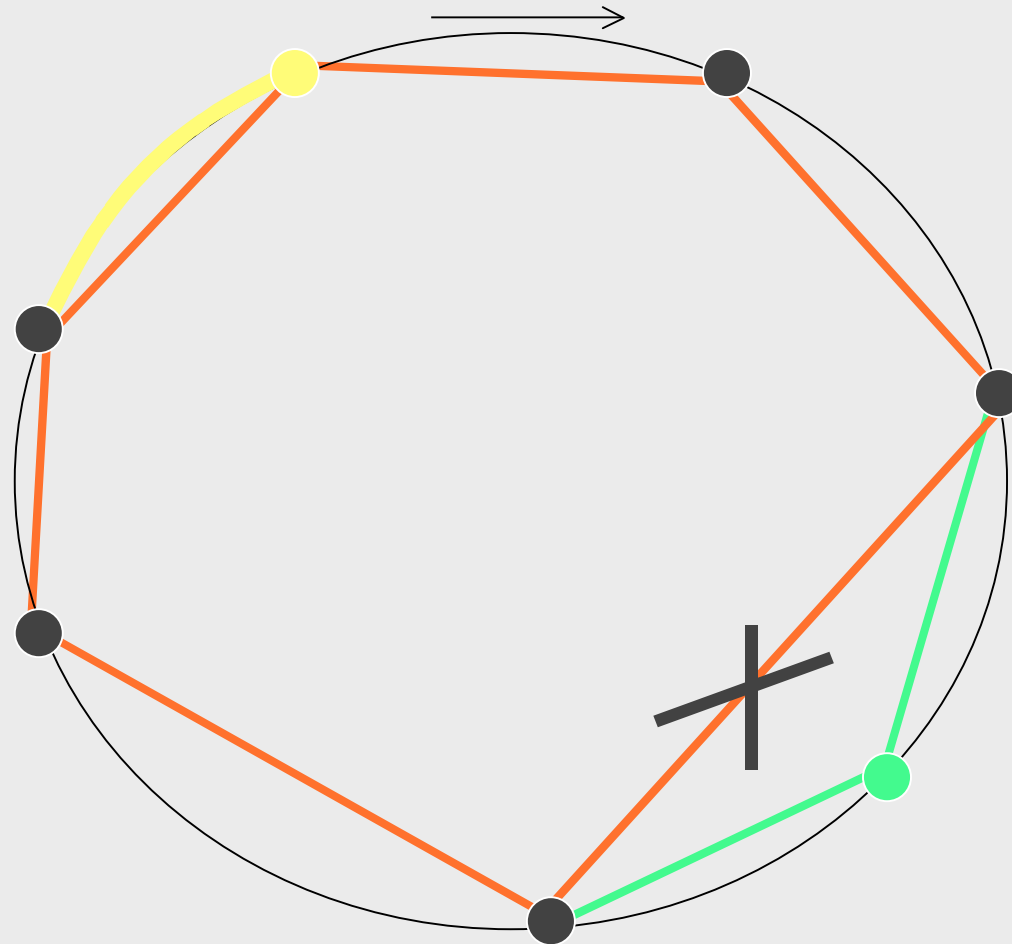
# PROBLEM

- Design a dynamic network (i.e., nodes can join and leave at their convenience) in which routing and updating are efficient.
- Note: Many solutions, based on standard static graph topologies

# CONSTRAINTS

- Fast updates
  - Limited amount of control messages
  - ⇒ small degree
- Fast lookups
  - Short routes ⇒ small diameter
- Balanced routing traffic
  - No hot spot ⇒ symmetric graphs

# EXAMPLE: THE ORIENTED RING

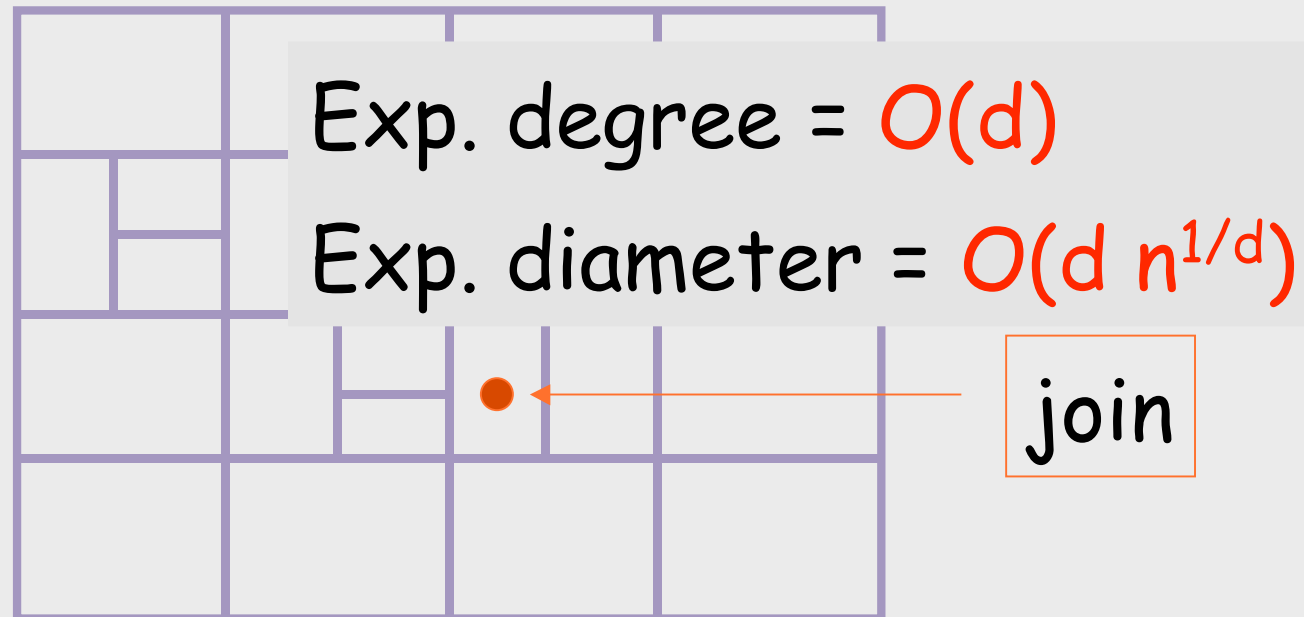


# CAN

“CONTENT-ADDRESSABLE NETWORK”

Ratnasamy, Francis, Handley, Karp, Shenker [SIGCOMM '01]

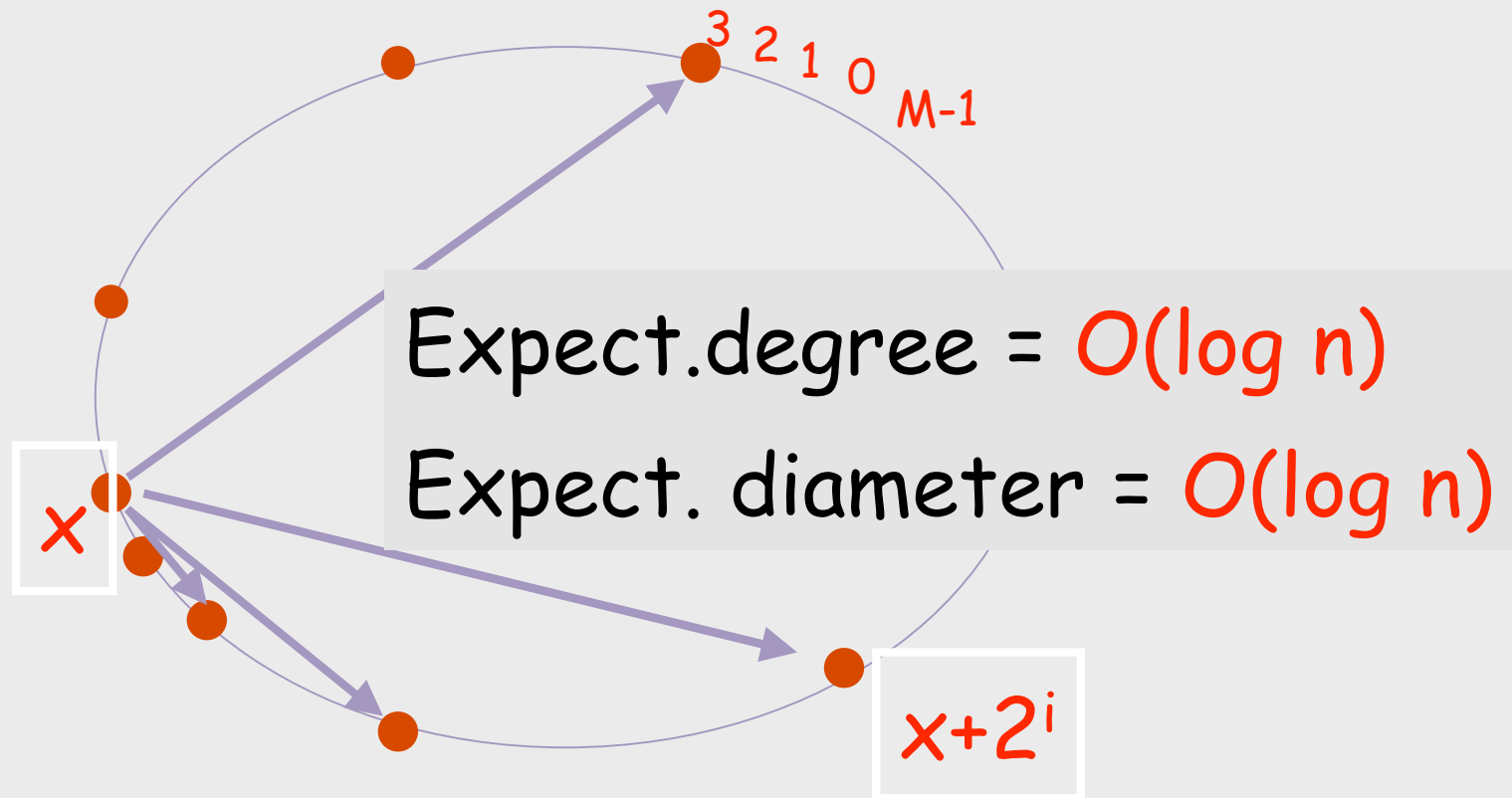
d-dimensionnal torus



# CHORD

Stoica, Morris, Karger, Kaashoek, Balakrishnan [SIGCOMM '01]

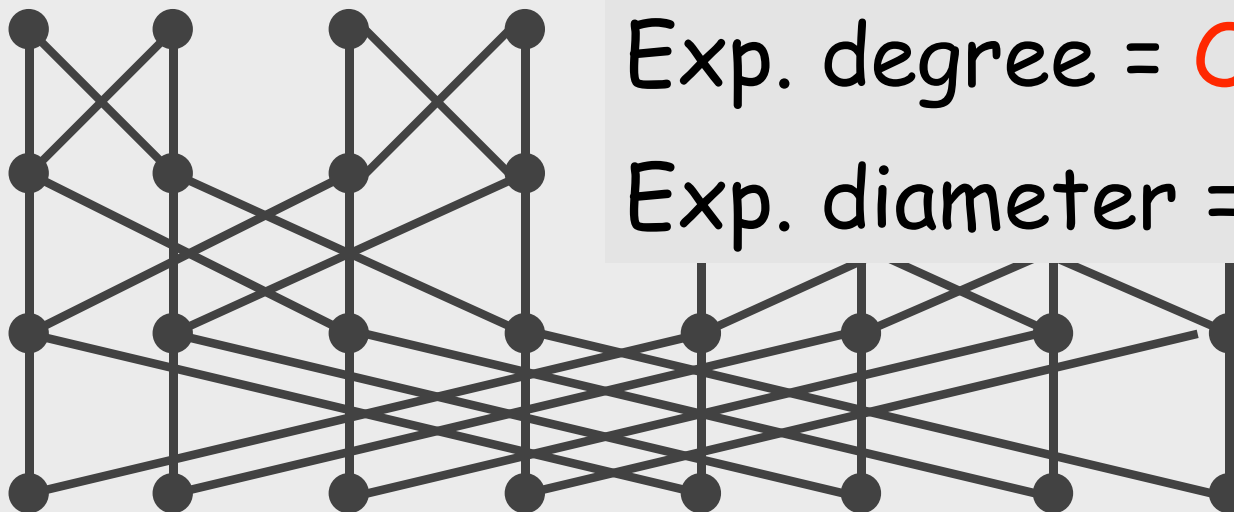
**d**-dimensional hypercube



# VICEROY

Malkhi, Naor, Ratajczak [PODC '02]

## Butterfly Network



Exp. degree =  $O(1)$

Exp. diameter =  $O(\log n)$

# DE BRUIJN-BASED DHTs

- I. Abraham, B. Awerbuch, Y. Azar, Y. Bartal, D. Malkhi, E. Pavlov: *A generic scheme for building overlay networks in adversarial scenarios*
- P. Fraigniaud, Ph. Gauron: *D2B: a de Bruijn Based Content-Addressable Network*
- F. Kaashoek, D. Karger: *Koorde: a simple degree-optimal distributed hash table*
- M. Naor, U. Wieder: *Novel architectures for P2P applications: the continuous-discrete approach*

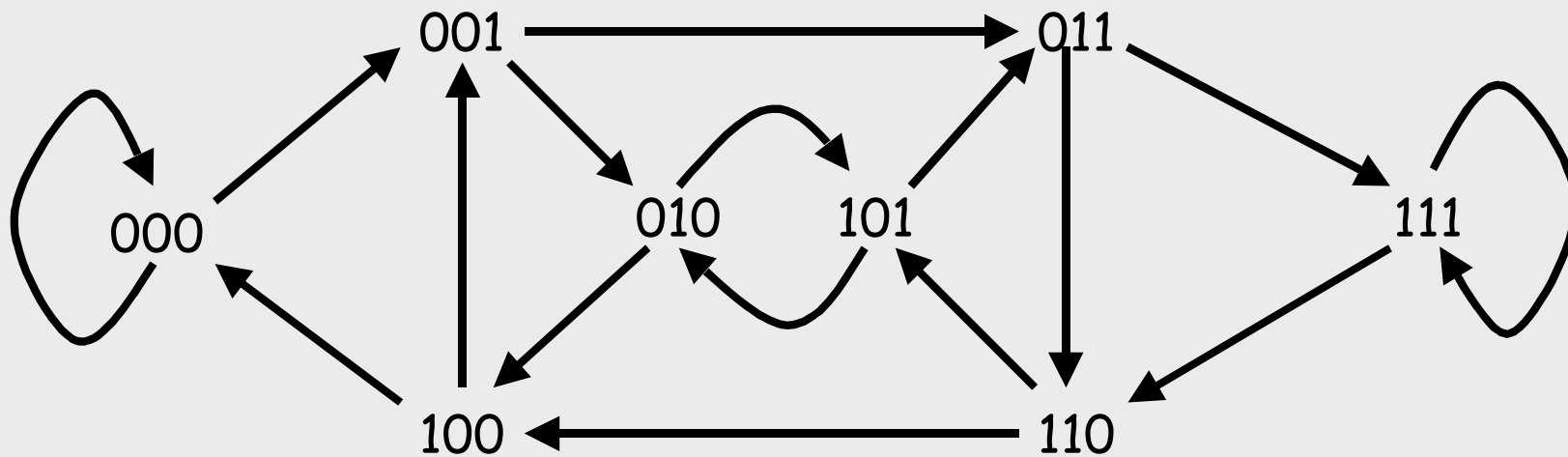
# D2B

- Based on the de Bruijn graph
- Measures:
  - #key per node
  - Degree
  - Length of the routes
  - Congestion
- Performances
  - In expectation
  - With high probability ( $\text{Prob} \geq 1-1/n$ )

# DE BRUIJN GRAPH

$V = \{\text{binary sequences of length } k\}$

$E = \{(x_1x_2\dots x_k) \rightarrow (x_2\dots x_k y), y=0 \text{ or } 1\}$



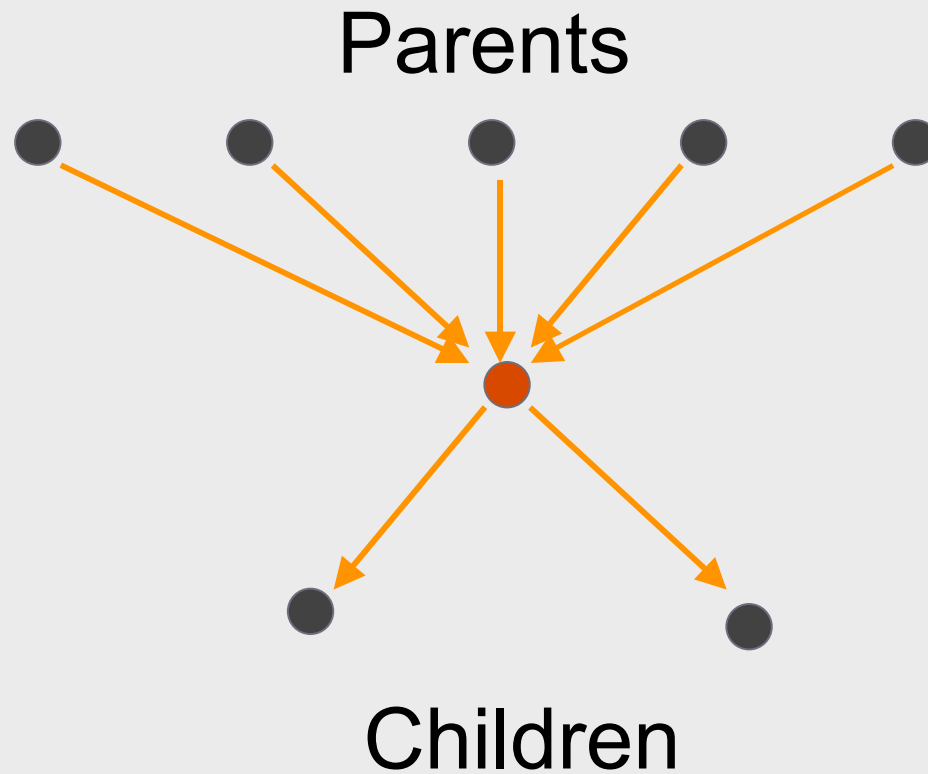
## NODE AND KEY LABELS

- Label = binary sequence of length  $\leq m$ .
- Key = binary sequence of length  $= m$ .  
⇒ up to  $2^m$  labels and keys  
In practice, set  $m=128$  or even  $256$
- The key  $\kappa$  is stored by node  $x$  if and only if  $x$  is a prefix of  $\kappa$ .

# UNIVERSAL PREFIX SET

- Let  $W_i$ ,  $i=1, \dots, q$ , be  $q$  binary sequences.
- The set  $S=\{W_1, W_2, \dots, W_q\}$  is a universal prefix set if and only if, for any infinite binary sequence  $B$ , there is one and only one  $W_i$  which is a prefix of  $B$ .
- Example:  $\{0, 11, 100, 1010, 10110, 10111\}$
- Remark:  $\{\varepsilon\}$  where  $\varepsilon$  is the empty sequence is a universal prefix set.
- By construction, the set of nodes in D2B is a universal prefix set.

# ROUTING CONNECTIONS



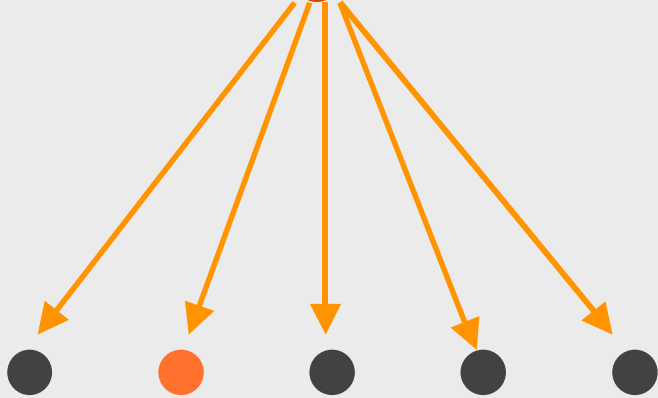
# CHILDREN CONNECTIONS AND ROUTING

$x_1 x_2 \dots x_k$



$x_2 \dots x_j$

$x_1 x_2 \dots x_k$



$x_2 \dots x_k y_1 y_2 \dots y_j$

The set  $\{y_1 y_2 \dots y_j\}$  is a UPS

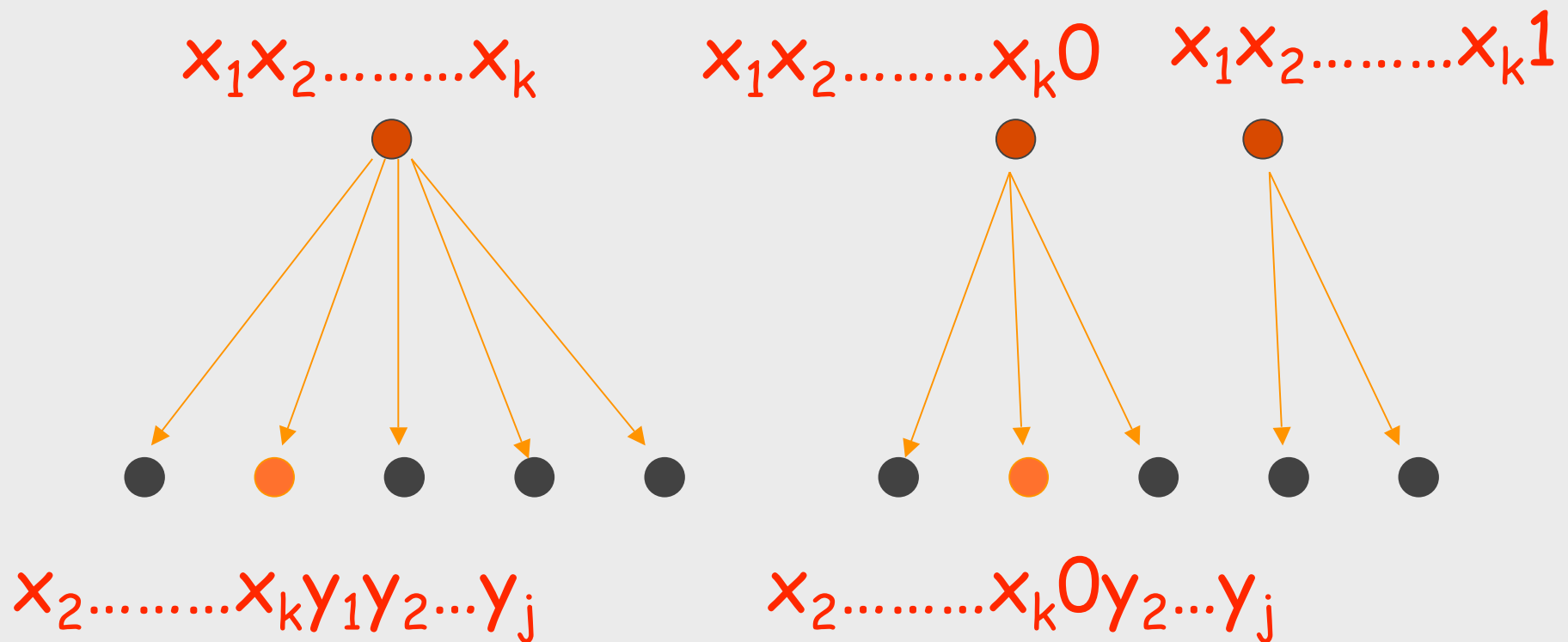
## JOIN PROCEDURE (1/3)

- A joining node  $u$  contacts an entry point  $v$  in the network
- Node  $u$  selects an  $m$ -bit binary sequence  $L$  at random: its preliminary label
- A request for join is routed from  $v$  to the node  $w$  that is in charge of key  $L$

## JOIN PROCEDURE (2/3)

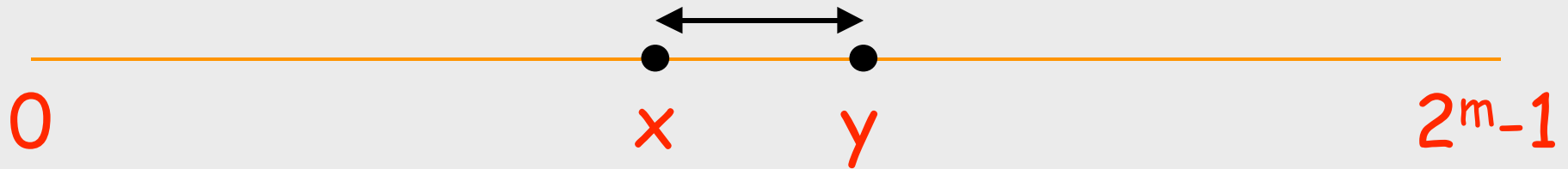
- Node  $w$  labeled  $x_1x_2\dots\dots x_k$  extends its label to  $x_1x_2\dots\dots x_k0$
- Node  $u$  takes label  $x_1x_2\dots\dots x_k1$
- Node  $w$  transfers to  $u$  all keys  $\kappa$  such that  $x_1x_2\dots\dots x_k1$  is prefix of  $\kappa$

# JOIN PROCEDURE (3/3)



# #KEYS PER NODE (1/2)

$$x_1x_2\dots x_k \Rightarrow x_1x_2\dots x_k0\dots\dots 0$$



## #KEYS PER NODE (2/2)

- Divide  $K$  in  $n/(c \log n)$  intervals, each containing  $c \log n |K|/n$  keys.
- Let  $X$  = #nodes in interval  $J$  starting at  $x$
- $n$  Bernoulli trials with probability  $p = c \log n/n$
- Chernoff bound:

$$\text{Prob}( |\sum X_i - np| > k ) < 2 e^{-k^2/3np}$$

$$\Rightarrow \text{Prob}(|X - c \log n| > (3c)^{1/2} \log n) < 2/n$$

$\Rightarrow$  W.h.p., there is at least one node in  $J$

$\Rightarrow$  W.h.p., a given node manages  $O(|K| \log n/n)$  keys

# LOOKUP ROUTING

Node  $x_1x_2\dots x_k$  looks for key  $k_1k_2\dots k_m$

$\Rightarrow x_2\dots x_k k_1\dots k_h$

$\Rightarrow x_3\dots x_k k_1\dots k_h k_{h+1}\dots k_{h+r}$

$\Rightarrow x_4\dots x_k k_1\dots k_h k_{h+1}\dots k_{h+i}$

$\Rightarrow x_5\dots x_k k_1\dots k_h k_{h+1}\dots k_{h+i} k_{h+i+1}\dots k_{h+i+s}$

$\Rightarrow x_6\dots x_t$

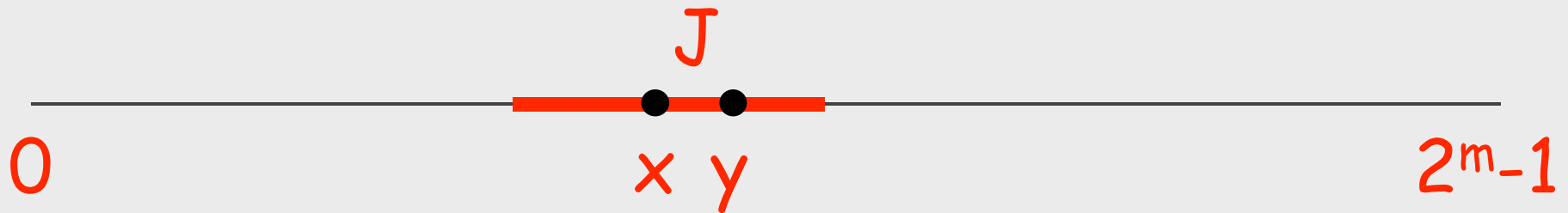
$\Rightarrow x_7\dots x_t k_1\dots k_d$

At most  $k$  hops to reach the node in charge of the key

$k_1k_2\dots k_m$

# LENGTH OF NODE LABEL (1/2)

$$x_1x_2\dots x_k \Rightarrow x_1x_2\dots x_k0\dots\dots 0$$



$$|J| = c |K| \log n/n$$

## LENGTH OF NODE-LABEL (2/2)

$$\text{Prob}(|X - c \log n| > (3c)^{1/2} \log n) < 2/n$$

$\Rightarrow$  W.h.p., at most  $O(\log n)$  nodes in  $J$

$\Rightarrow$   $x$  manages at least  $|J| / 2^{O(\log n)}$  keys

$$\Rightarrow k \leq m - \log |J| + O(\log n)$$

$$\Rightarrow k \leq O(\log n)$$

$\Rightarrow$  W.h.p., a lookup route is of length  $O(\log n)$

## DEGREE AND CONGESTION

- W.h.p., degree =  $O(\log n)$  using similar techniques (expected degree  $O(1)$ )
- Congestion = proba that a node is traversed by a lookup from a random node to a random key =  $O(\log n/n)$

## SUMMARY: EXPECTED PROPERTIES

	Update	Lookup	Congestion
CAN	$O(d)$	$O(dn^{1/d})$	$O(d/n^{1-1/d})$
Chord	$O(\log n)$	$O(\log n)$	$O(\log n/n)$
Viceroy	$O(1)$	$O(\log n)$	$O(\log n/n)$
D2B	$O(1)$	$O(\log n)$	$O(\log n/n)$

Using Small World and Scale Free  
Properties for the Design of  
Overlay Networks in P2P Systems

# COMMUNITIES

- Context: unstructured overlay networks
- Objective: create communities
- Rule: keep connected to nodes with whom you exchanged resources
- Impact: the search time is significantly reduced (observed in, e.g., Gnutella)
- Reason: acquaintances have high clustering coef., thus resources you are interested in are close to you in a network that maps the acquaintance graph.

# HIGH-DEGREE-FIRST SEARCH

- Context: unstructured overlay networks with power law degree distribution
- Rule: High-degree-first search strategy
  - Every node keeps track of the list of all the resources stored by its neighbors
  - DFS search visiting high degree neighbors first
- Impact: sub-linear search time
- Reason: well informed nodes are reached rapidly

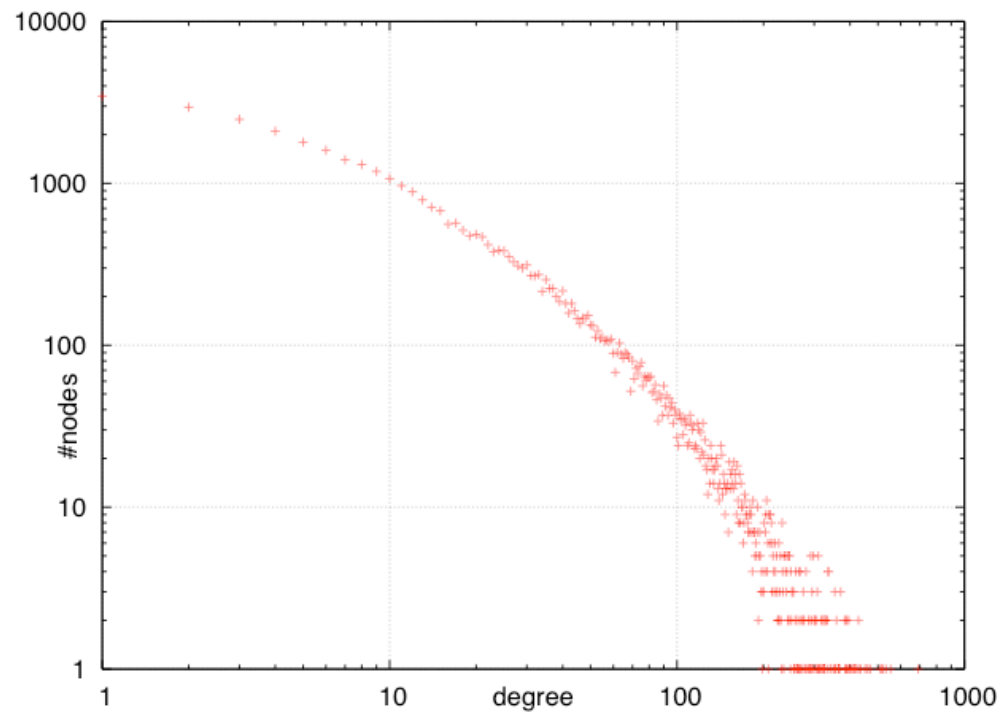
# MIXING THE TWO

- Nodes
  - join one by one, and initially connect to  $k$  arbitrary nodes
  - keep connected to nodes with whom they exchanged resources
  - store the lists of their neighbors' resources
  - perform DFS search with high-degree node priority

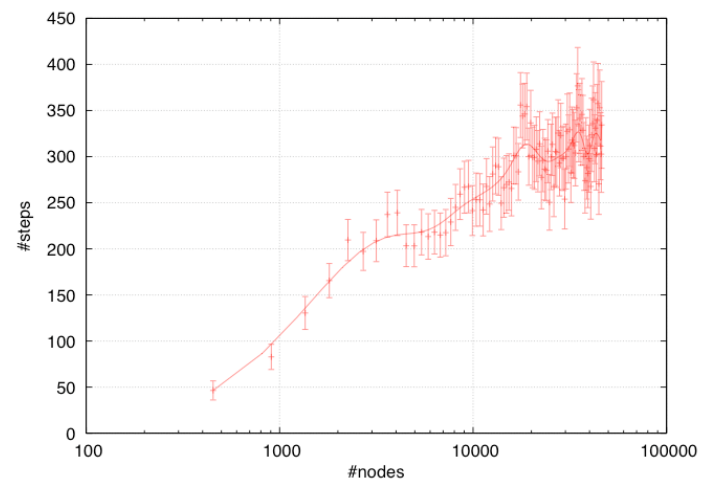
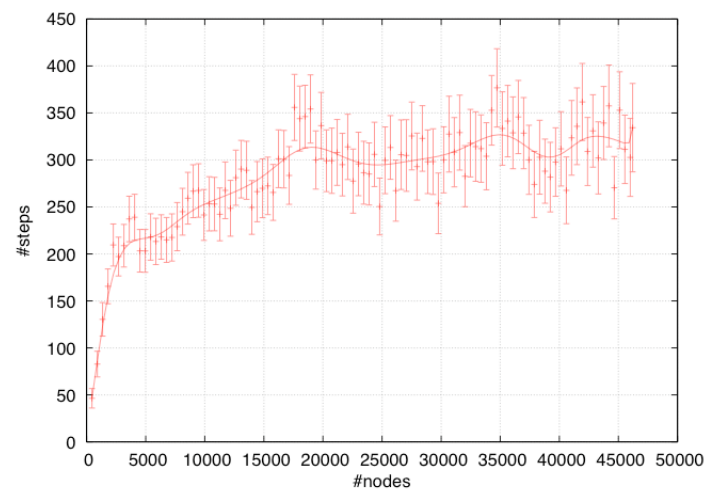
# EXPERIMENTS

- P2P trace from eDonkey
- The trace is 2h53 long
- Involves 46.202 peers and 342.204 requests
- Simulation of each (search) request:
  - Routing from source to targets (there can be many)
  - Update connections

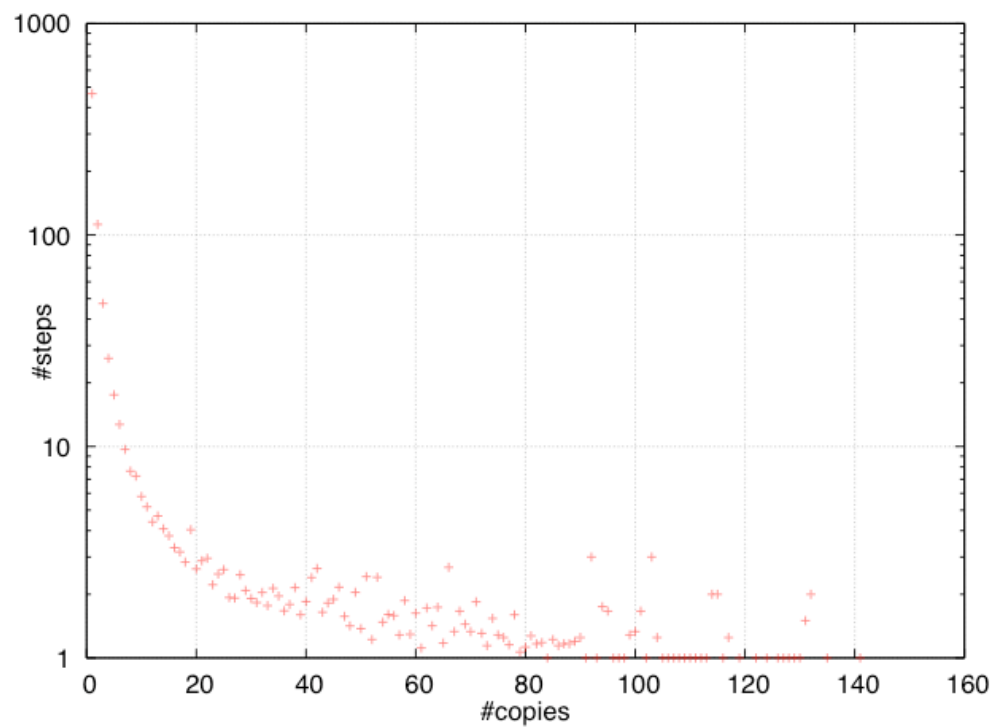
# DEGREE DISTRIBUTION



# SEARCH TIME



# SEARCH TIME VS. #COPIES



# CONCLUSION

- P2P paradigm is fruitful, and may apply to various fields (data bases, ad hoc networks, etc.)
- DHT technique is powerful, but needs further improvements to insure more flexibilities