

# Infrastructures (overlays)

## Auto\*

# Construction et Applications

Maria.Gradinariu@lip6.fr

# Infrastructures - exemples

- Internet
- Réseaux P2P - infrastructures logiques
  - non-structurées
  - structurées (Distributed Hash Tables)
- Réseaux Mobiles & Réseaux de capteurs
  - Clustering – cellules, pico-cellules
  - Couvertures connexes
  - backbones

# Infrastructures - applications

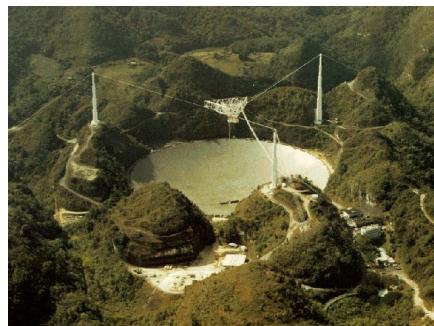
- Router et rechercher des données et services
  - Gnutella, Kazaa, BitTorrent
- Communiquer et collaborer
  - Chat/Irc, NewsGroups
  - Instant Messaging (Aol, Icq, Yahoo, Msn)

# Infrastructures - applications

- Partager/répartir des données, bande passante, puissance de calcul ou stockage
  - Napster, Publius, Freenet, MojoNation, FreeHaven, Groove, e-donkey, Chord, Can, Pastry, Tapestry
  - Voice/IP (Skype)
  - Seti@home (astronomie)
  - genome@home (ADN)
  - folding@home (repliement des protéines)

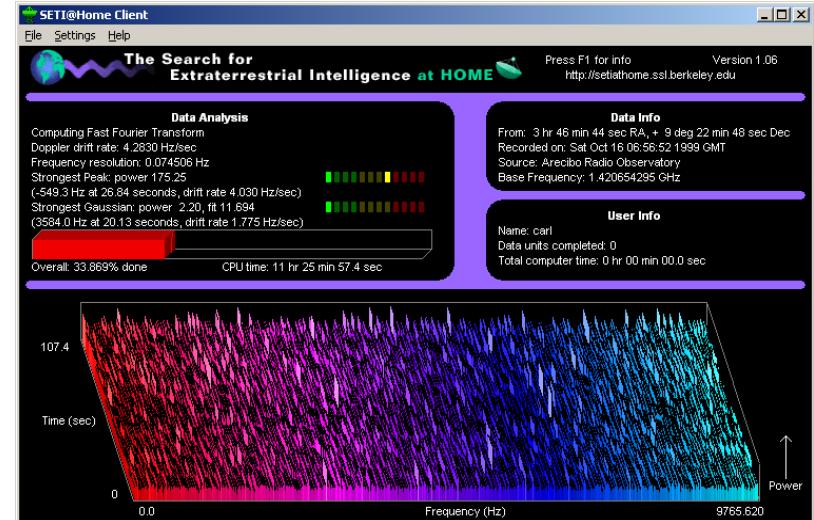


- Expérience en radioastronomie exploitant la puissance inutilisée de millions d'ordinateurs connectés via Internet dans un projet de Recherche d'une Intelligence Extra-terrestre (Search for Extra-Terrestrial Intelligence, alias SETI).
- Les ordinateurs chargent et analysent les données collectées du plus grand radiotélescope au monde à Arecibo.
- 2,4 millions de participants de 226 pays et territoires



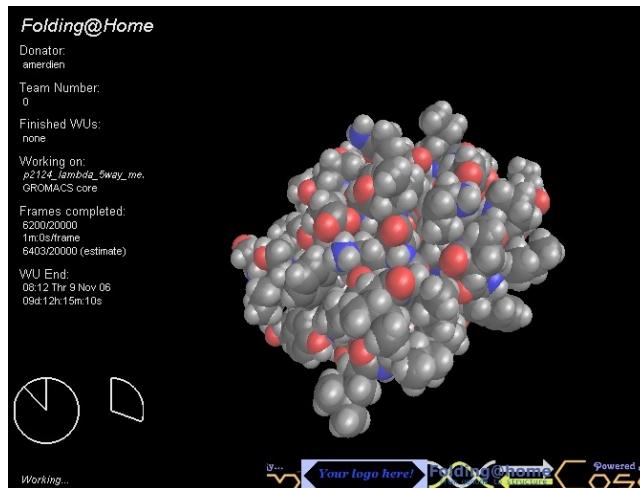


- 3.8M utilisateurs dans 226 pays
- 1200 années CPU / jour
- 38 TeraFlops soutenu (Le Earth Simulator Japonais obtient 40 TF)
- 1.7 Zettaflop ( $10^{21}$ ) pour les 3 dernières années
- Très hétérogène : >77 types de processeurs différents





- Comprendre le repliement et l'agrégation des protéines et les maladies qui sont liées
- Etude de maladies comme celle d'Alzheimer, la fibrose cystique, l'EBS (la Vache Folle), une forme héréditaire de l'emphysème et de nombreux cancers résultent d'un repliement anormal des protéines.
- Depuis le 1er octobre 2000, plus de 500,000 CPUs ont participé à ce programme



# Réseaux P2P

# Infrastructures P2P

- non –structurées (Gnutella, Kazaa )
  - la topologie du système est déterminée par les utilisateurs
  - le placement des données dans le système ne tient pas compte de la topologie du système
- structurées (Chord, CAN, Pastry,Tapestry)
  - le placement des données dans le système est fait à des locations précises (utilisation des fonctions de « hash »)
  - la topologie du réseaux a des formes particulières (ex. anneau, arbre, grille)

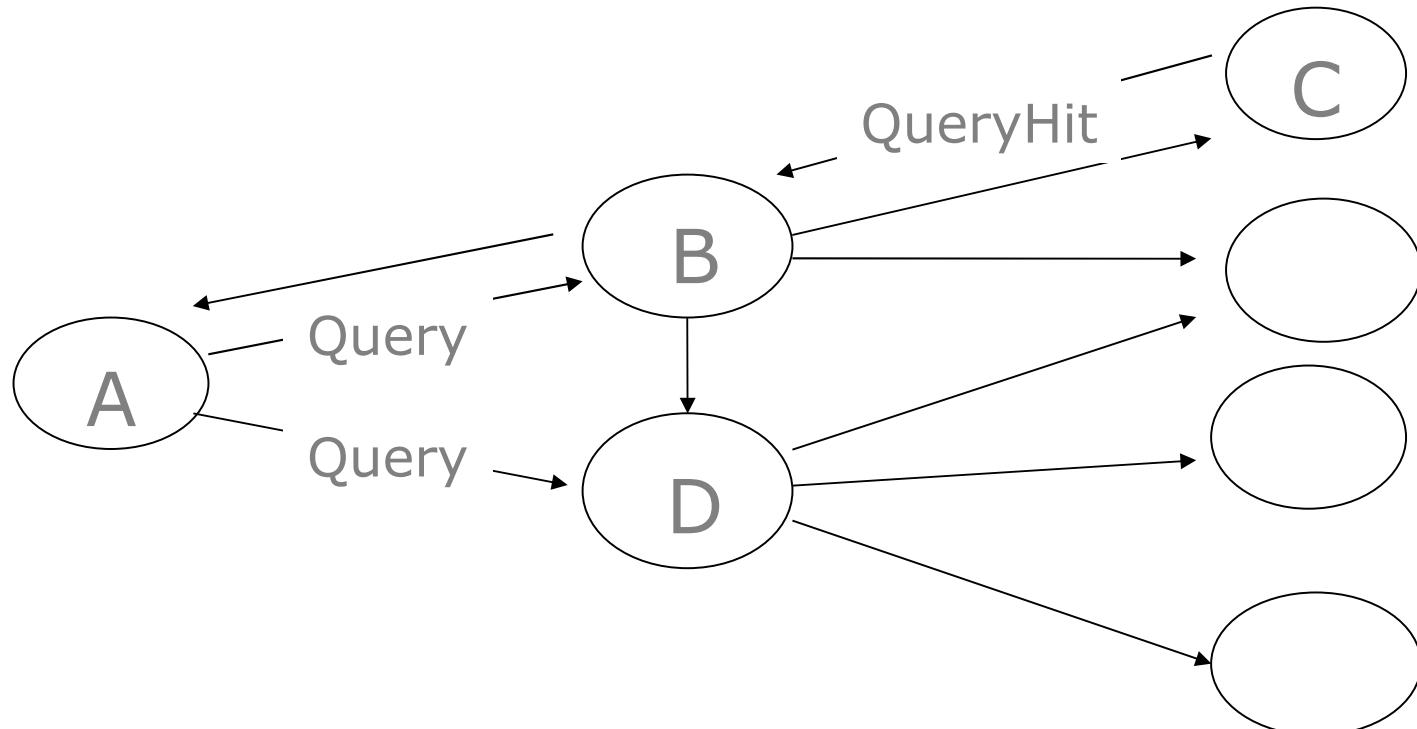
# Réseaux P2P non-structurées

## Gnutella

- Protocole de recherche de données et services
  - chaque nœud est à la fois client et serveur
- Messages Gnutella (TimeToLive)
  - découverte de nœuds PING/PONG
  - découverte de données (fichiers) et services
    - Query
    - QueryHit

# Réseaux P2P non-structurées

## Gnutella



# Réseaux P2P semi-structurées

## FreeNet

- Stockage persistant de données et services
  - Données identifiées par une clé binaire (fonction hash)
  - les données qui traversent un nœud sont copiées dans le cache du nœud
    - L'utilisation de la politique LRU pour la gestion du cache
    - L'information stockée par donnée (fichier)
      - code hash
      - Le dernier temps d'accès/modification

# Réseaux P2P semi-structurées

## FreeNet

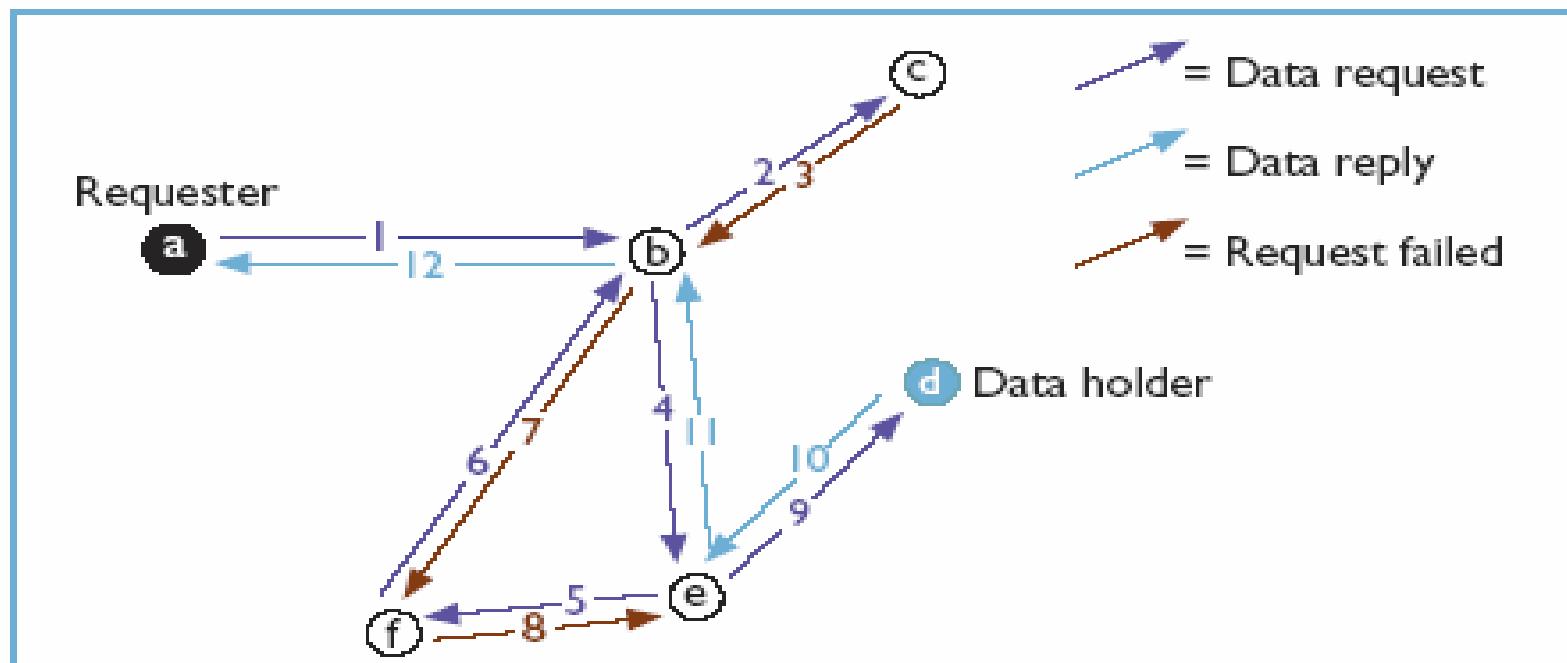


Figure 1. Typical request sequence. The request moves through the network from node to node, backing out of a dead-end (step 3) and a loop (step 7) before locating the desired file.

# Gnutella vs. FreeNet

- Routage basé sur la diffusion (flooding) vs. Routage dynamique basé sur la similarité des clés
- Aucune mémoire du trafique passé vs. Tables de routage
- Read-only vs. Read/Write
- Système non sécurisé vs. Système sécurisé

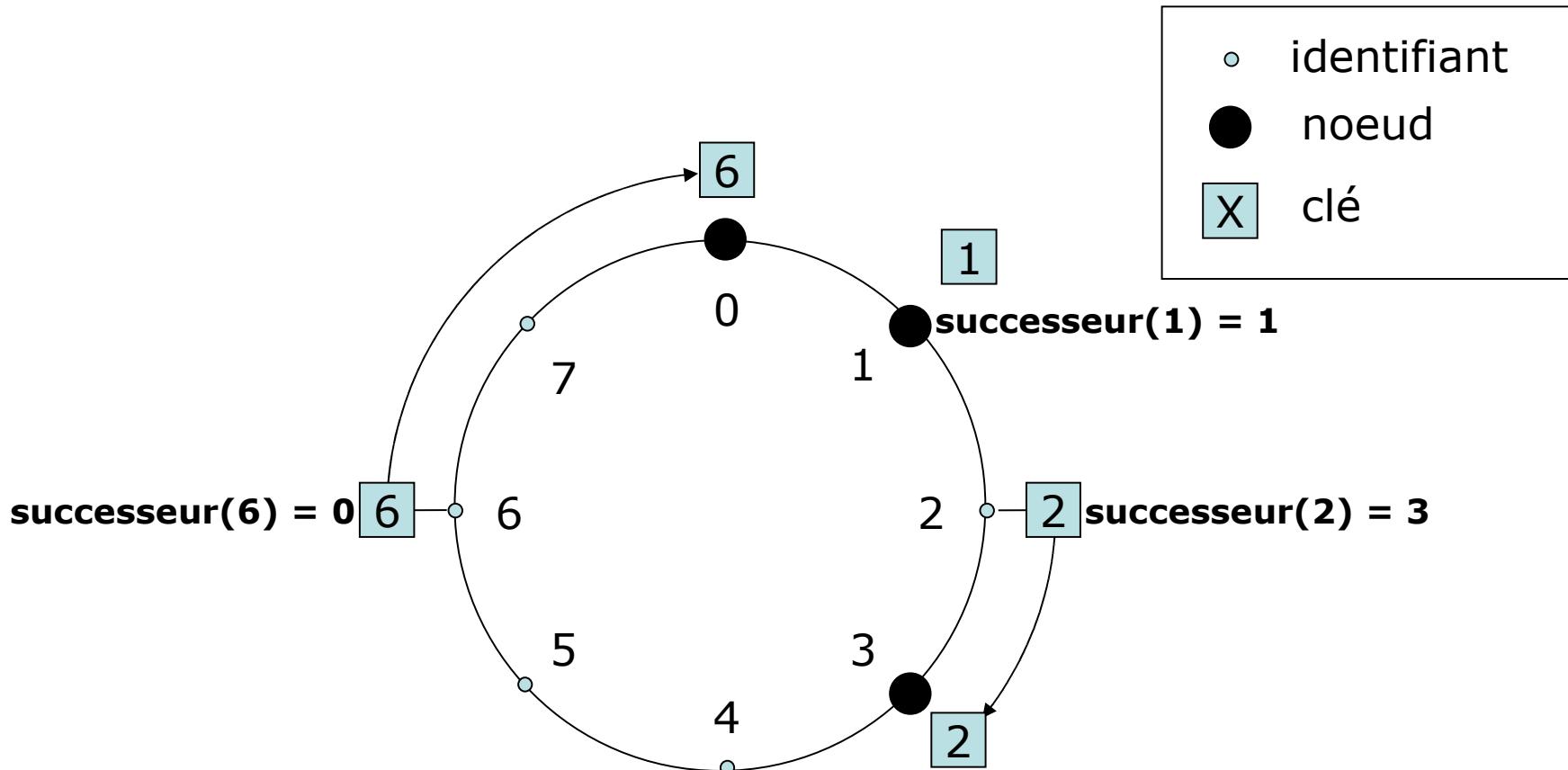
# Réseaux P2P structurées

## Chord

- une infrastructure de stockage et routage
- à chaque utilisateur et à chaque fichier dans le système on associe un identifiant sur  $m$  bits ( $2^m$  identifiants)
  - Id utilisateur = hash(IP)
  - Id fichier ou clé fichier = hash(contenu)
  - L'espace des IDs organisé en anneau
  - Un fichier de clé  $k$  est stocké sur le premier noeud dans le système ayant l'identifiant  $y$  ( $y > k \text{ mod } 2^m$ )

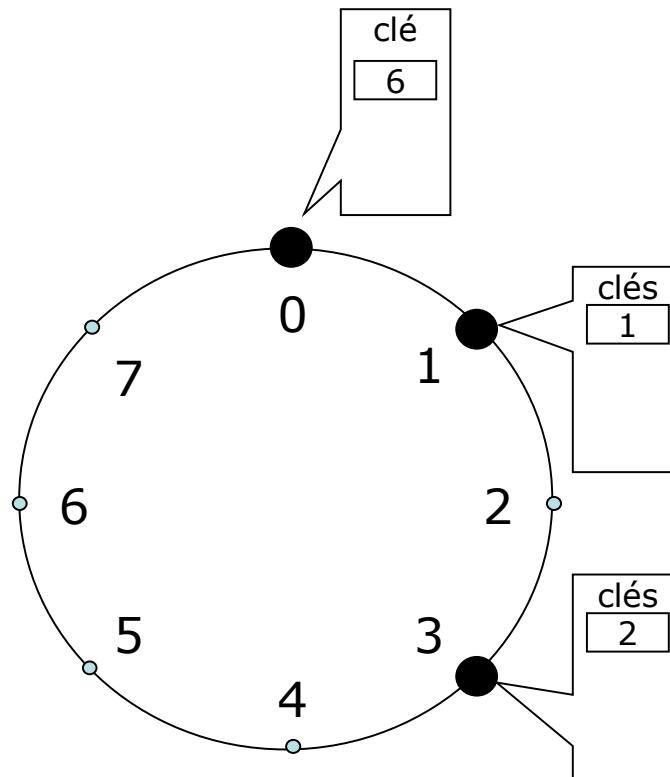
# Réseaux P2P structurées

## Chord (l'association clés - noeuds)



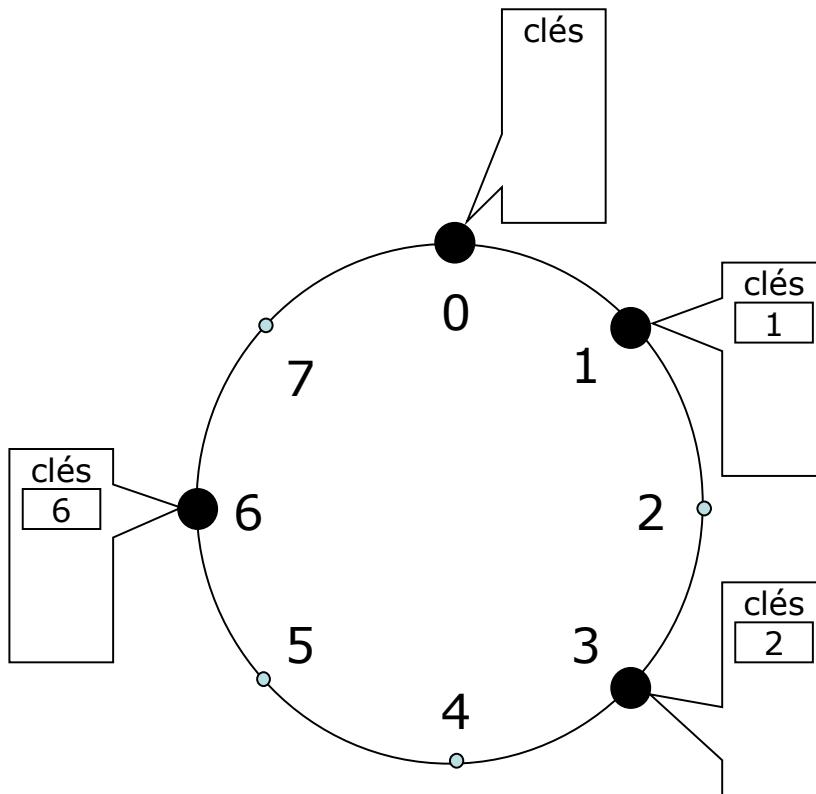
# Réseaux P2P structurées

## Chord (l'association clé - noeuds)



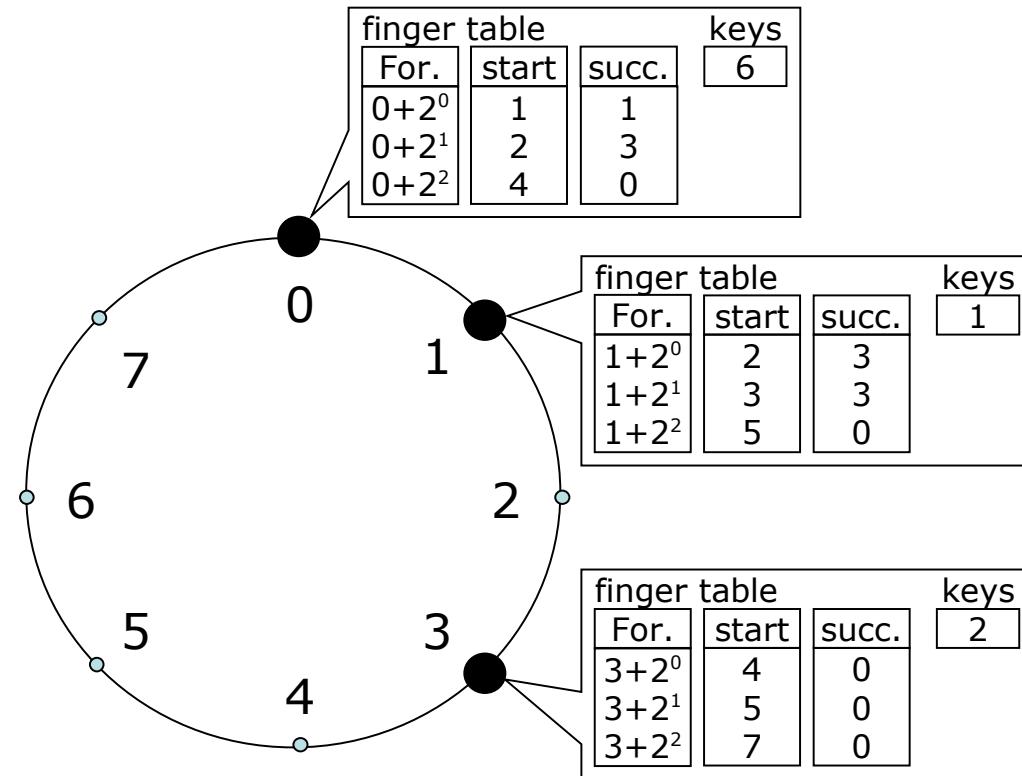
# Réseaux P2P structurées

## Chord (l'entrée du nœud 6)



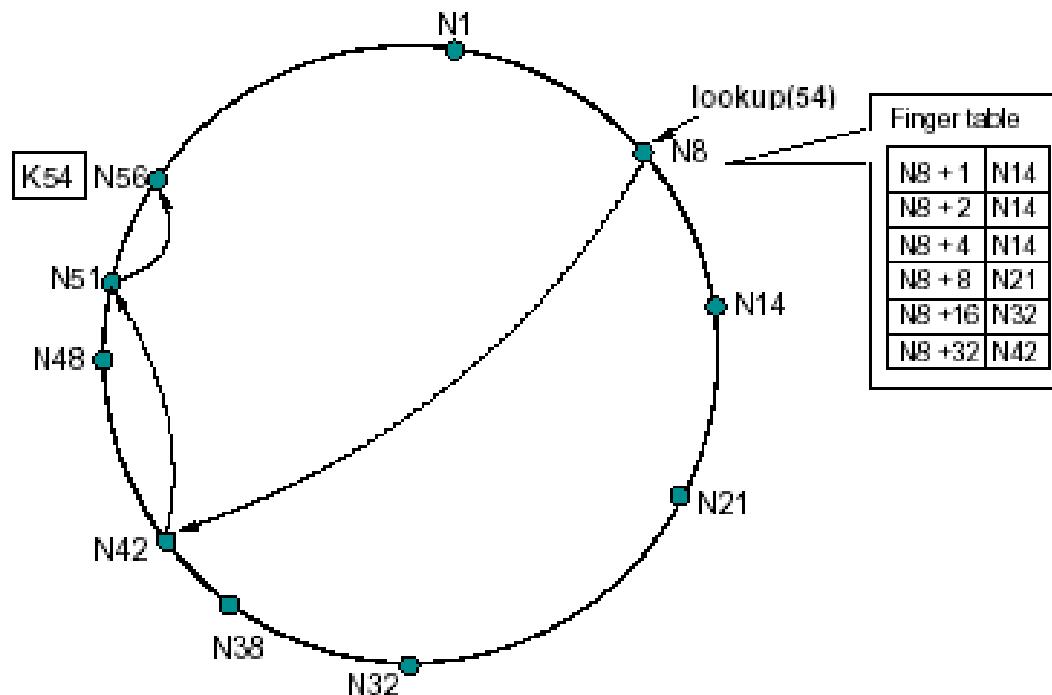
# Réseaux P2P structurées

## Chord (les raccourcis)



# Réseaux P2P structurées

## Chord (la recherche de la clé 54)



# Réseaux P2P structurées

## Chord

- la mémoire utilisée par noeud  $O(\log(N))$
- le temps de recherche d'une clé  
 $O(\log(N))$
- le système est auto-reconfigurable et tolérant aux défaillances

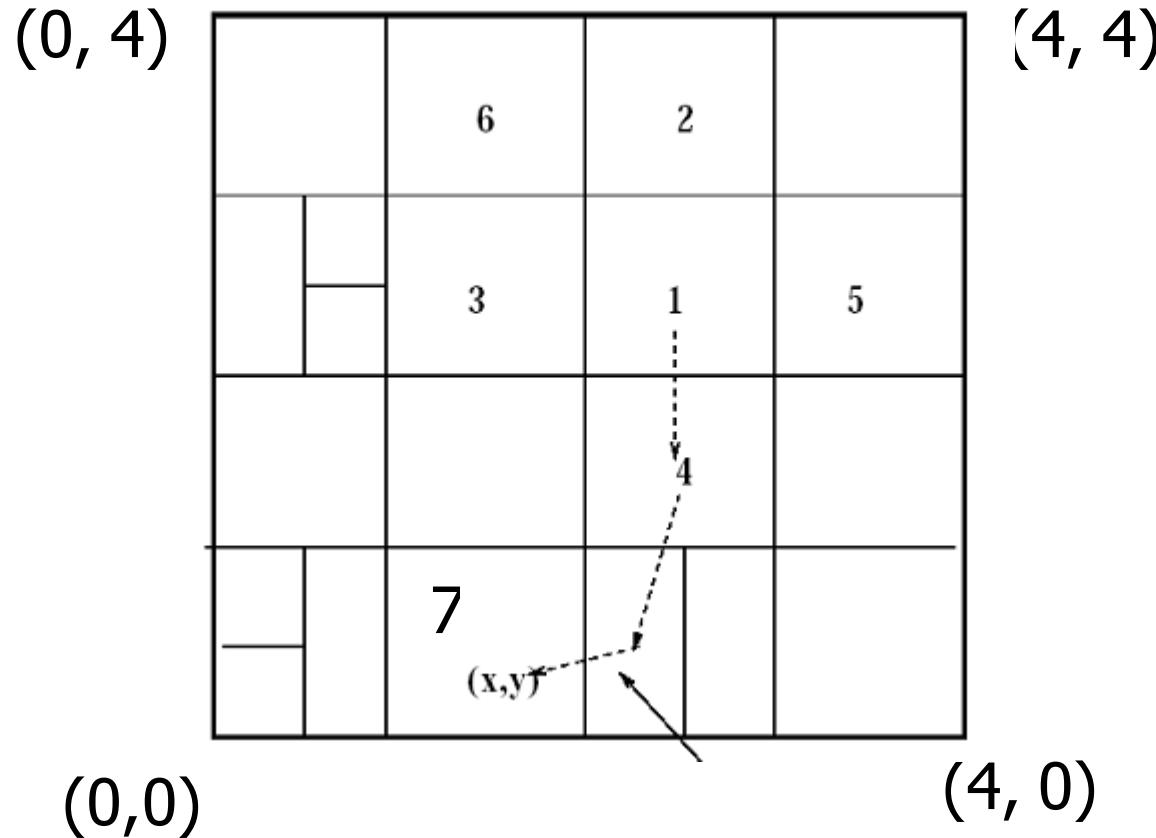
# Réseaux P2P structurées

## CAN

- Idée de conception :
  - espace cartésien virtuel 2-dimesionnel
  - chaque nœud du système est le propriétaire d'une zone dans l'espace virtuel
  - les données sont stockées sous la forme (clé, val)
    - $\text{hash}(\text{clé}) \rightarrow$  un point  $(x,y)$  dans l'espace virtuel
    - $(\text{clé}, \text{val})$  est stocké par le nœud propriétaire de la zone dont fait partie  $(x,y)$

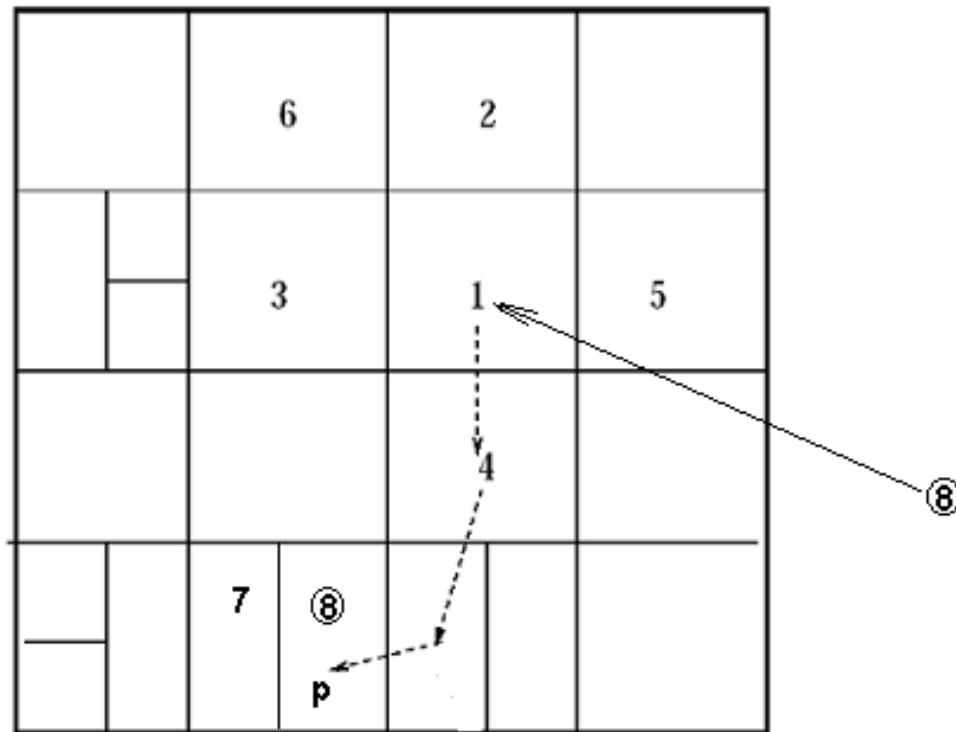
# Réseaux P2P structurées

## CAN (le routage)



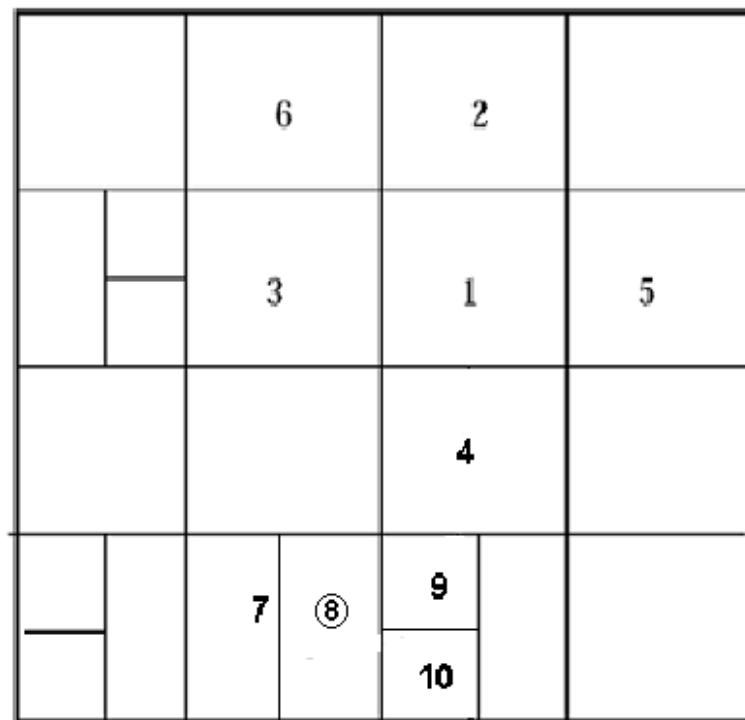
# Réseaux P2P structurées

## CAN (l'insertion du nœud 8)



# Réseaux P2P structurées

## CAN (le départ du nœud 9)



# Infrastructures P2P

## RéPLICATION de données

- FreeNet – les données sont copiées par les nœuds qui participent à leur routage
- MojoNation – les copies des données très demandées sont disséminées dans le réseau par un serveur
- CAN (multi-dimensionnel) – une donnée peut avoir une clé par dimension

# Infrastructures P2P

## Sécurité



*"On the Internet, nobody knows you're a dog."*

# Infrastructures P2P

## Sécurité

- éviter les connexions directes entre le demandeur d'information et le propriétaire (FreeNet)
- utiliser des TTL choisis aléatoirement
- dissocier le propriétaire d'un document de l'emplacement où le document est stocké (CAN, Chord)

# Infrastructures P2P

## Sécurité

- la vérification des données
  - clés cryptographiques (CFS, Past)
- la dissémination des données
  - les fichiers à stocker sont décomposés en n blocs de telle manière à ce que m blocs ( $m < n$ ) sont suffisants pour reconstituer le fichier (Publius, Mnemosyne, FreeHaven)

# Infrastructures P2P

## Sécurité

- « free-riding » et collusion
  - un ou plusieurs utilisateurs profitent du système sans partager leur ressources
  - Problème : écroulement du système
  - Solutions :
    - Utilisation des technique d'incitation à la participation
      - Découverte des ressources du système proportionnelle à la participation
      - Payement virtuel ou micro-payement (MojoNation)
    - Surveiller les pairs

# Infrastructures P2P

## Sécurité

- « sybil attack » [Douceur 2002]
  - Un utilisateur peut entrer dans le réseau en utilisant plusieurs identités
  - Problèmes dans les systèmes qui utilisent la réPLICATION ou la fragmentation de données
  - Solution (utopique) : l'identification unique des ressources d'un nœud

# Mobile Robots and distributed computing

Maria Gradinariu Potop-Butucaru  
[maria.gradinariu@lip6.fr](mailto:maria.gradinariu@lip6.fr)  
LIP6, Université Paris 6

# Real Robots

- UAVs(Unamed aerial vehiculs) & AUV  
(Autonomous underwater vehiculs)
- Autonomous Ground Robots
- Metamorphic Robots

# UAVs



RQ-2 Pioneer used in the golf war

- Military operations
  - Reconnaissance
  - Attack
- Civilian applications
  - Firefighting
  - Police observation
- Characteristics
  - Remote controlled
  - Pre-programmed flight schedule

# UAVs & AUVs practical challenges

- Distributed Cooperation
  - UAVs Adaptive Flight schedule (in the air and for take off/landing)
  - Pattern formation
  - Connectivity (visual and/or communication)
- Fault-tolerance
  - Byzantine faults
  - Crash faults
  - Transient faults

# Autonomous Ground Robots



- Military operations
- Sensing capabilities
- Artificial intelligence embedded
  - Adaptive route planning
  - Adaptive motion (walk, trot, run)

DARPA – BIG DOG

# Autonomous Ground Robots practical challenges



- Performant Sensors and cameras (LASAR – laser detection and ranging is currently used)
- Distributed Cooperation and coordination
- Fault-tolerance

DARPA – BIG DOG

# Metamorphic Robots



S-bot project

- Cooperative modules
  - They can connect to each other and form complex structures
- Broad area of application
  - Exploration
  - Execute complex tasks

# Metamorphic Robots practical challenges



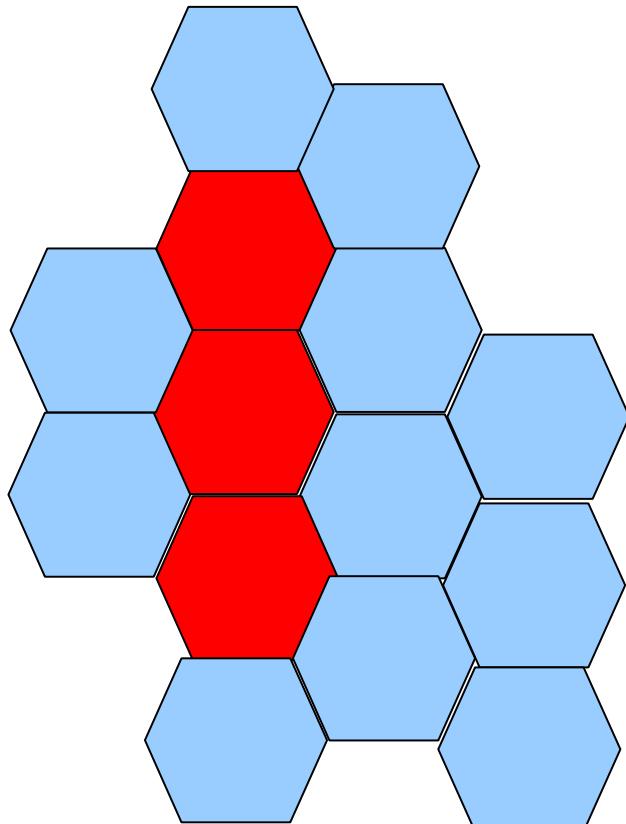
S-bot project

- Fully decentralization
- Fault tolerance
  - Crash faults
  - Byzantine faults
  - Transient faults
- Use of minimal communication

# Theoretical Models

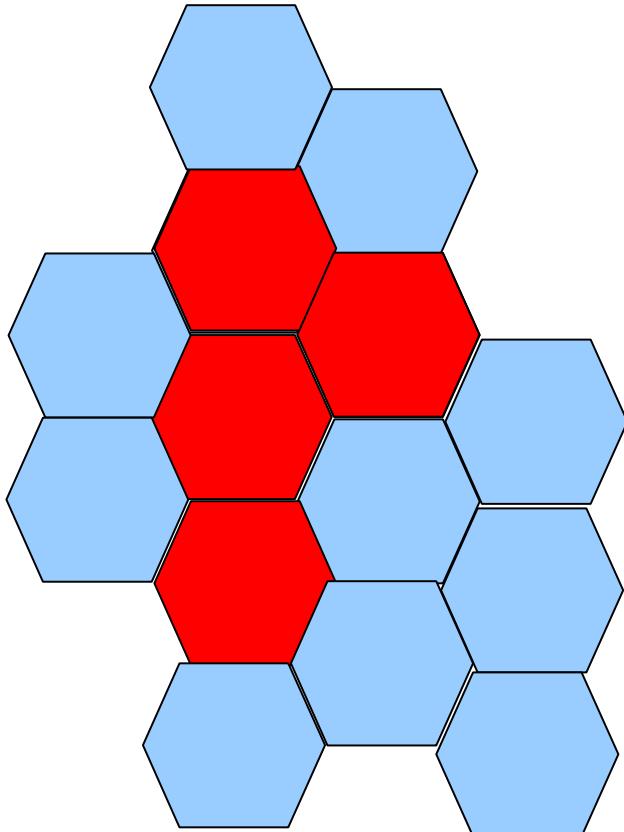
- Distributed metamorphic robots
- Topology restricted robots
- Oblivious planary robots

# Distributed Metamorphic robots (Welch et al. 2004)



- All modules share the same coordinate system
- Each module knows:
  - Its current location
  - Which cell is occupied by another module
- Round-based execution:  
in each round an unique module moves

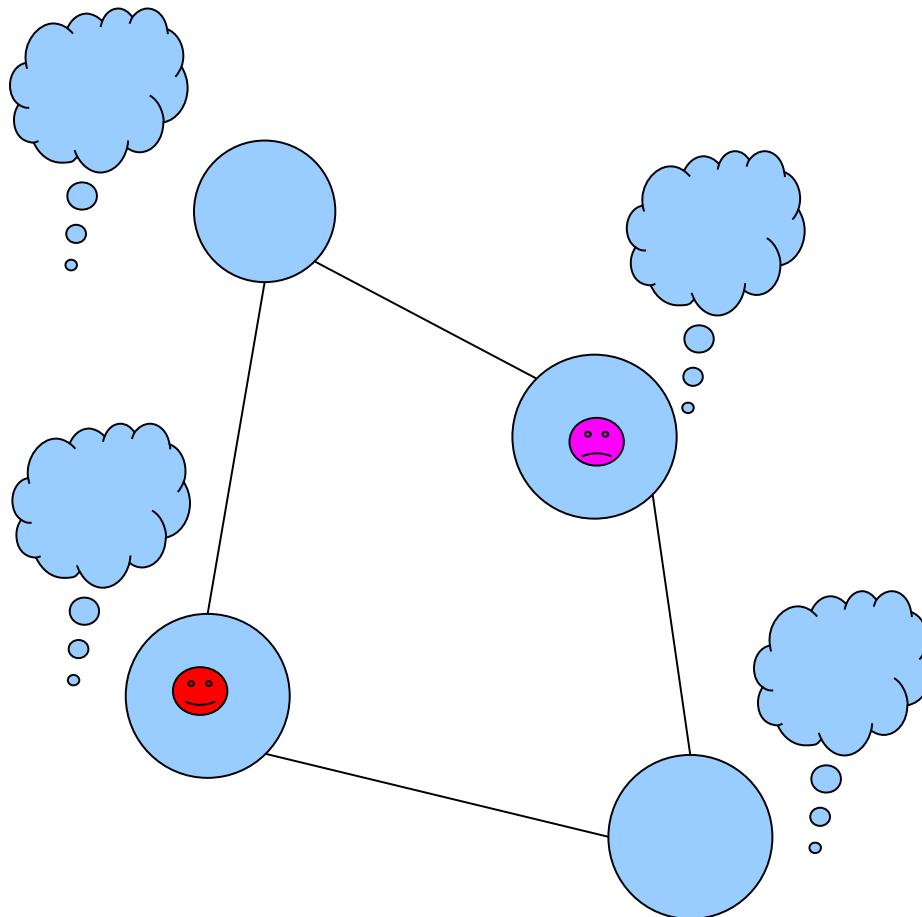
# Distributed Metamorphic robots (Welch et al. 2004)



- All modules share the same coordinate system
- Each module knows:
  - Its current location
  - Which cell is occupied by another module
- Round-based execution:  
in each round an unique module moves

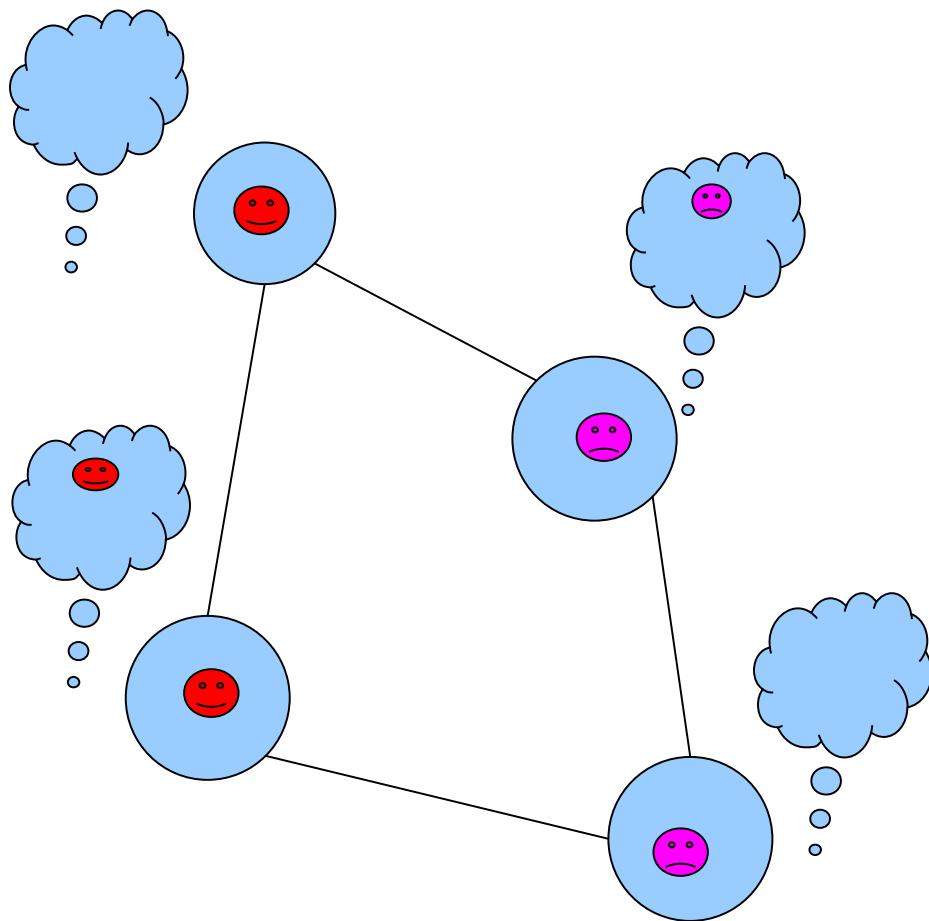
# Topology restricted robots (mobile agents on graphs)

Communication:

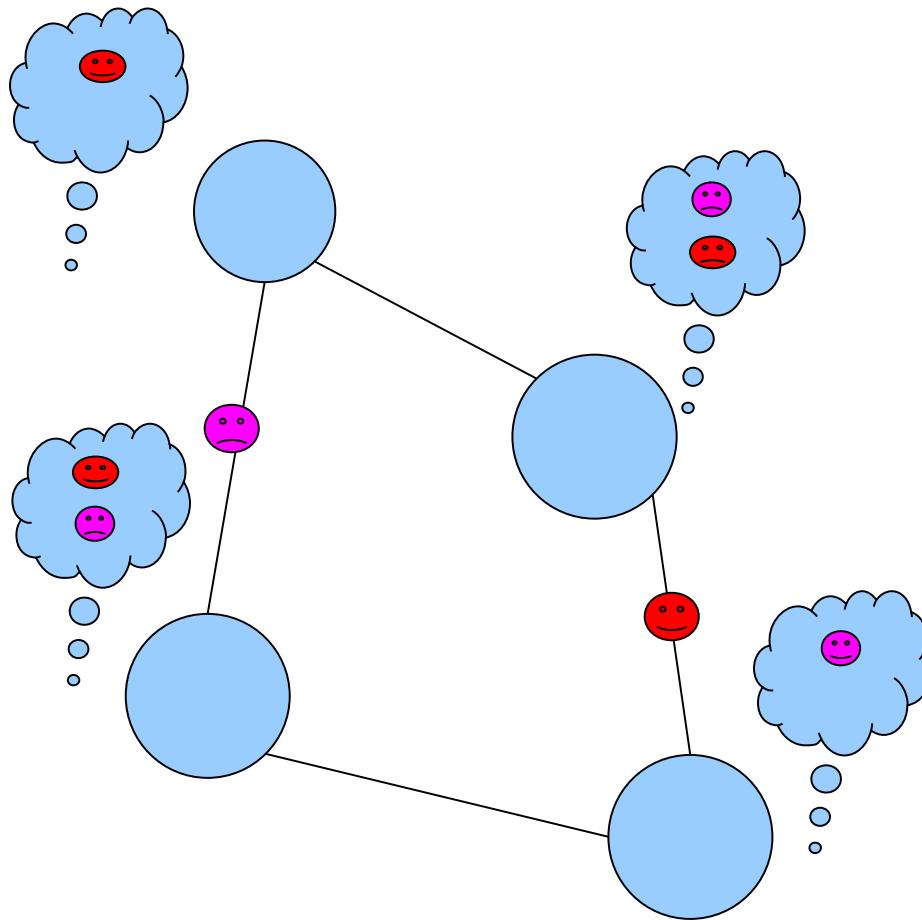


- white boards
- hand-shake (when robots meet on a particular node)
- System:
  - Asynchronous
  - Synchronous
- Memory assumptions:
  - Oblivious robots/white boards on the nodes of the graph
  - Robots with memory/oblivious nodes in the graph

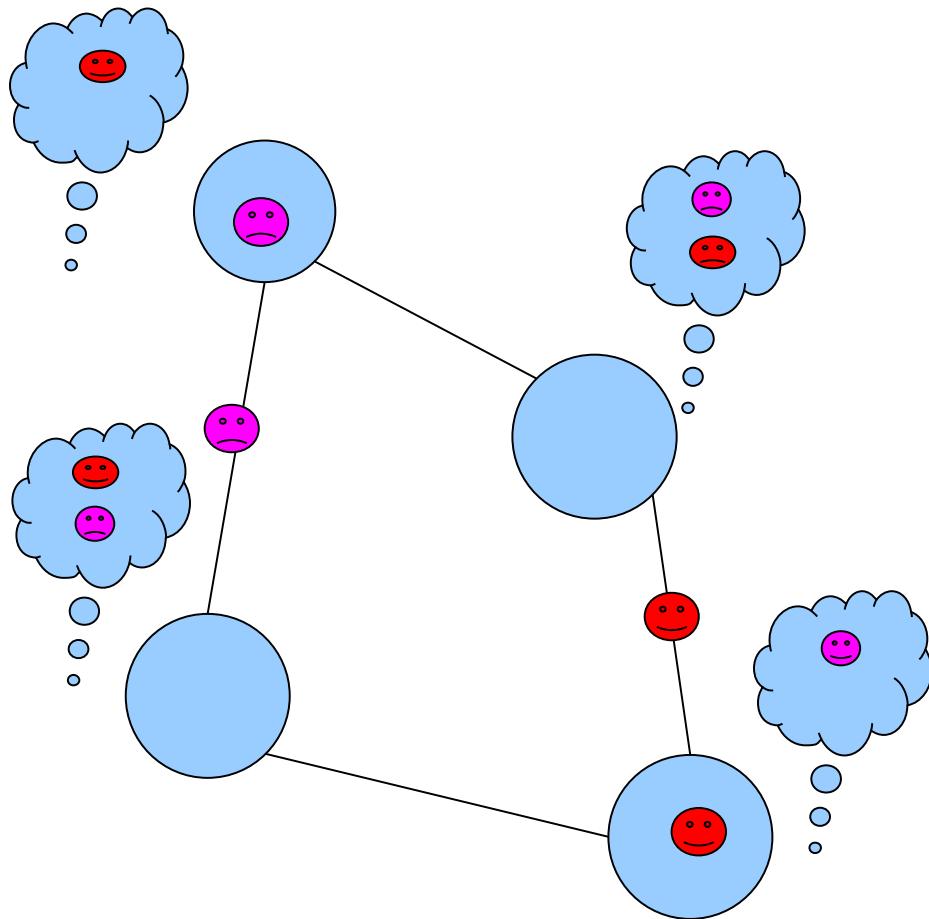
# Topology restricted robots (mobile agents on graphs)



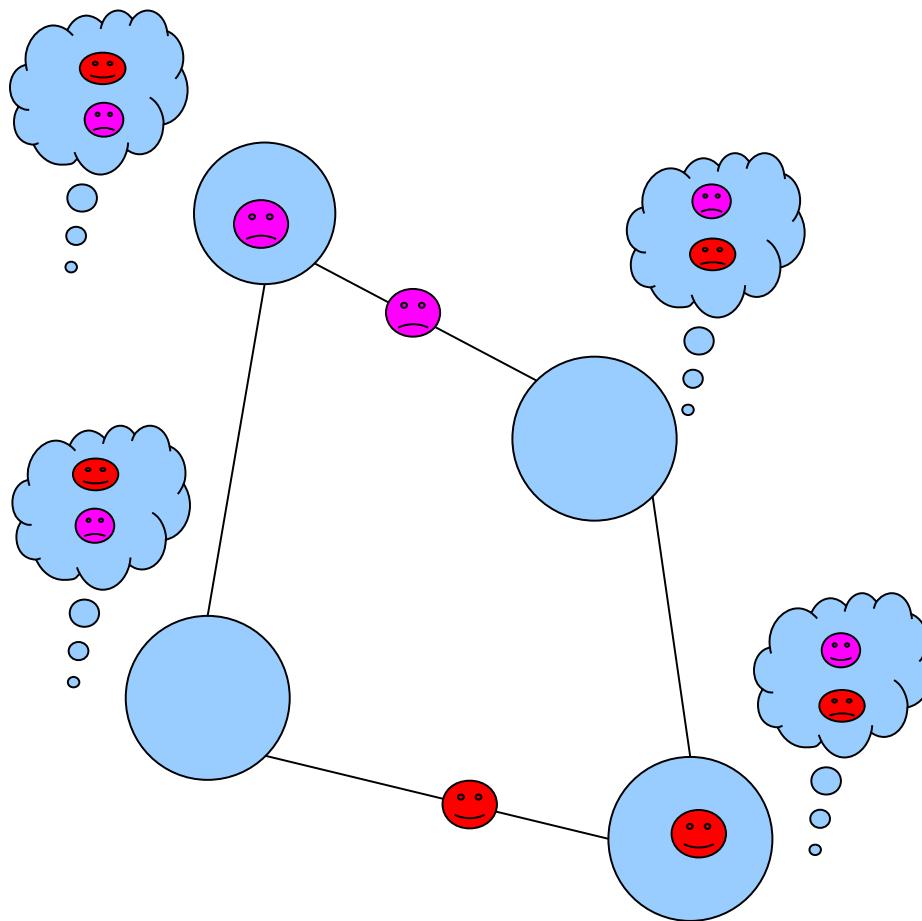
# Topology restricted robots (mobile agents on graphs)



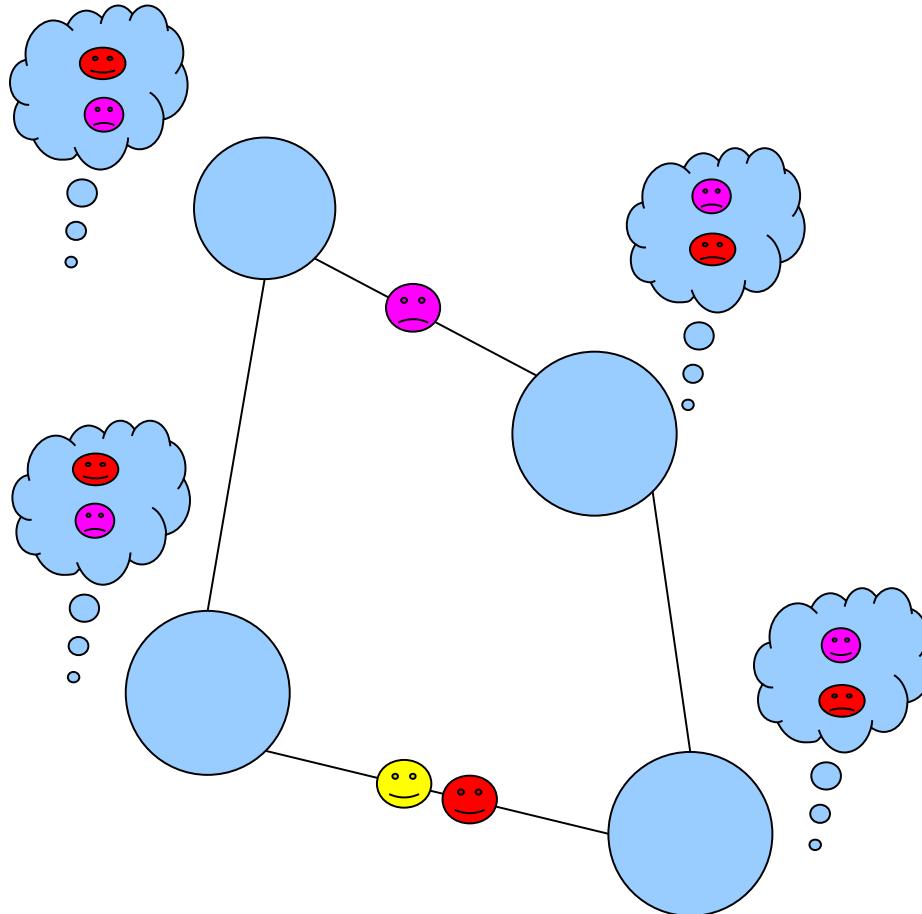
# Topology restricted robots (mobile agents on graphs)



# Topology restricted robots (mobile agents on graphs)



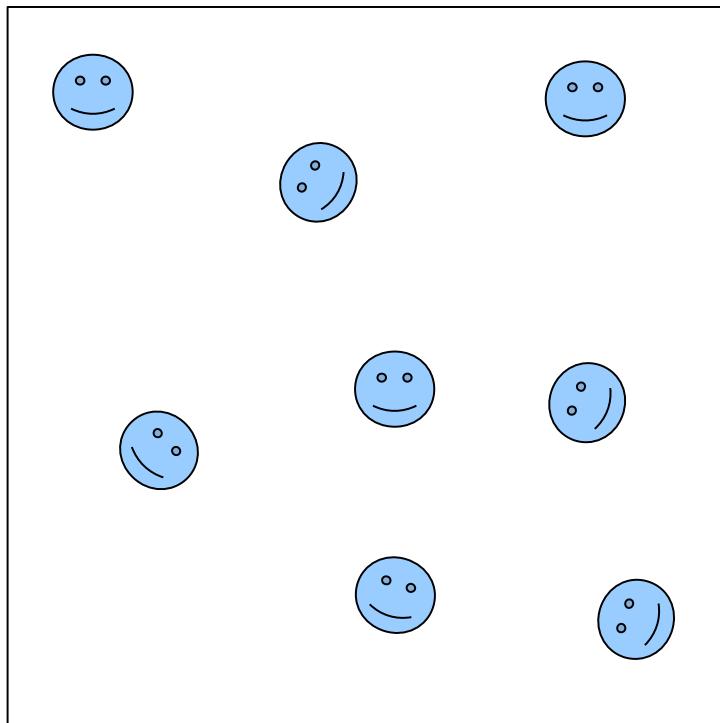
# Topology restricted robots challenges



- The class of problems that can be solved in these models
- Fault-tolerance
  - Transient
  - Byzantine
  - Crash / Duplications
  - White washing

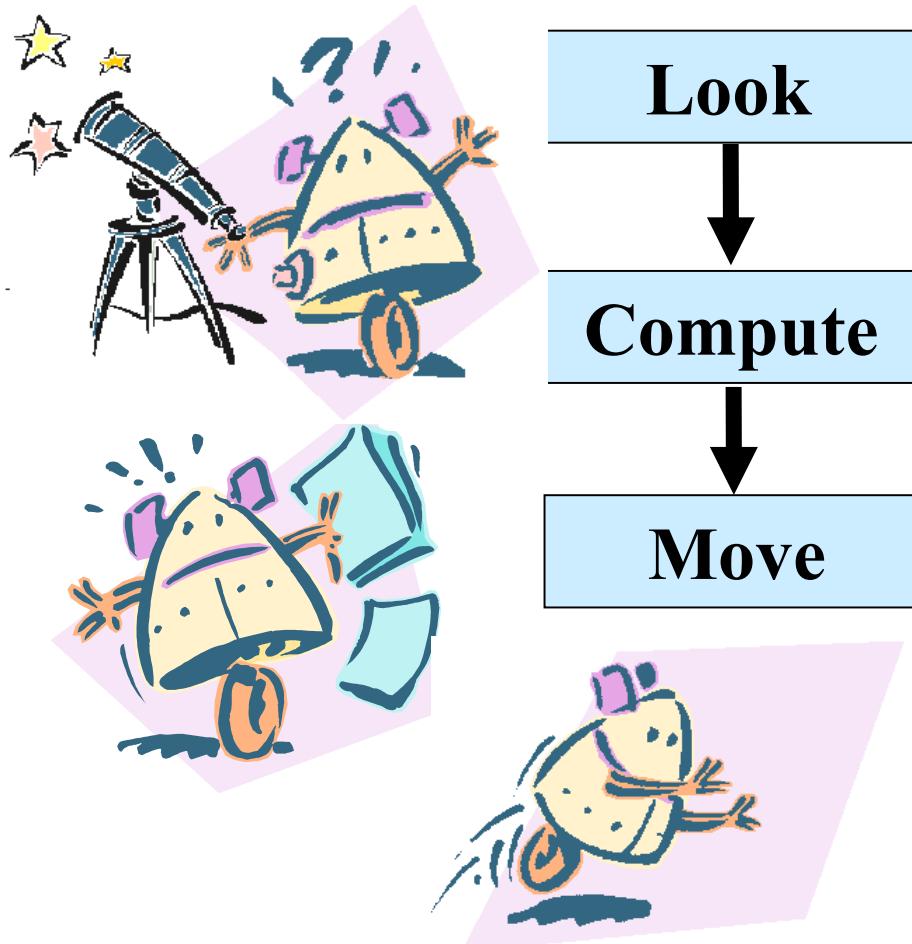
# Distributed Oblivious planary Robots

## Suzuki and Yamashita '96



- Anonymous.
- No direct communication.
- No common coordinate system.
- No common sens of direction (no common compas)
- Memoryless.
- No volume.

# Computational Models for Oblivious Robots

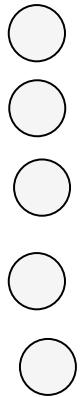


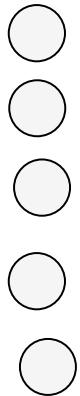
- **SYM (semi-synchronous)**

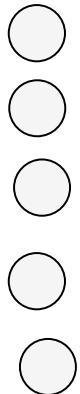
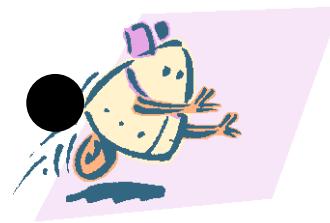
Atomic cycles (LOOK-CALC-MOVE)

Some robots may not be activated.

- **CORDA (asynchronous)**



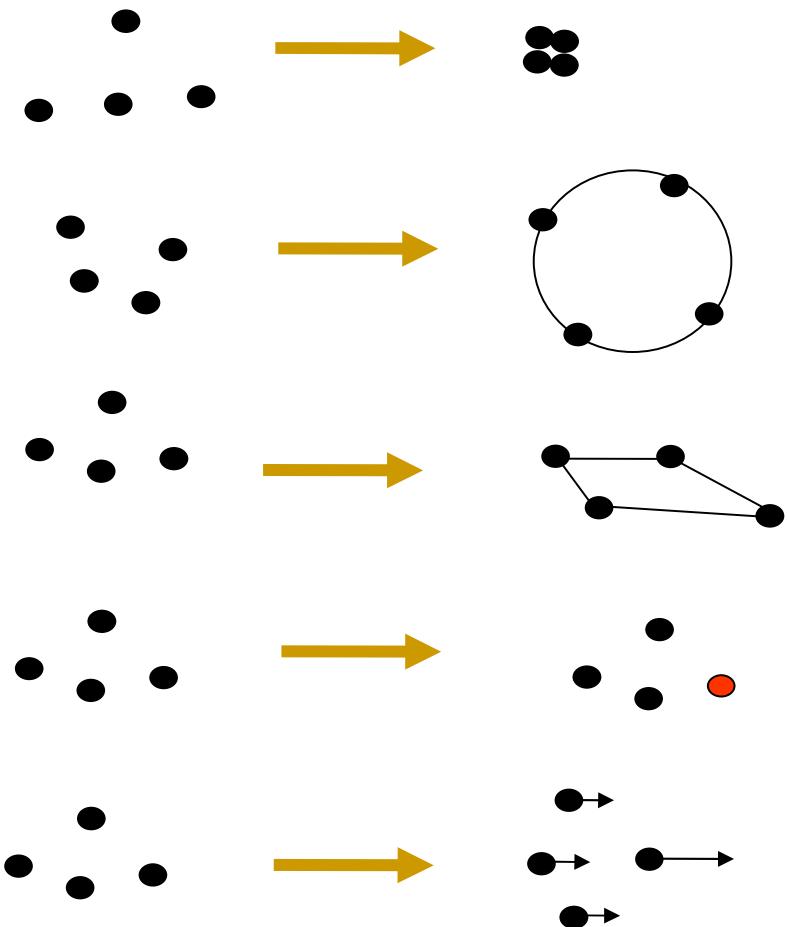






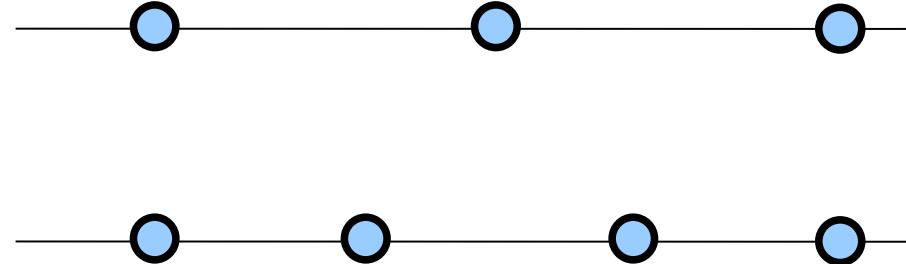
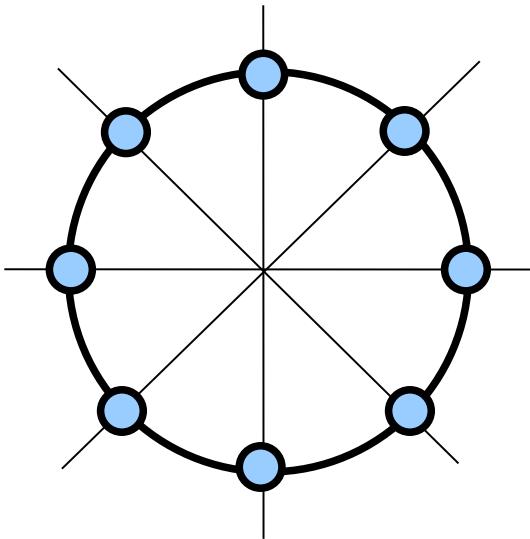
# Tasks of interest

- Gathering
- Pattern formation
- Election
- Flocking



# What Makes the Difference ? (1) Robots are ANONYMOUS

- Breaking the symmetry is **hard** and sometimes **unfeasible**.
- Cannot rely on the presence of unique identifiers.



# What Makes the Difference ? (2)

## COORDINATE System

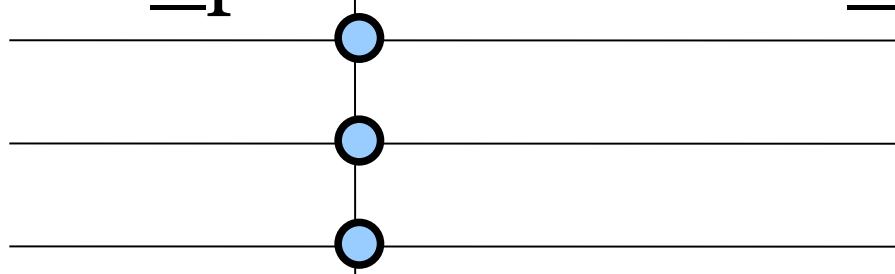
Functions with different semantic:

**Maximum** (among a set of positions)



**Incrementation**

( $x=x+1$  /  
`current_position=current_position+1`)



# What Makes the Difference ? (3) Robots are OBLIVIOUS

- (+) Dynamic environments.
- (+) Inherently self-stabilizing.
  
- (-) Robots cannot remember **previous states**.
- (-) Decisions based only on the **current configuration**.
- (-) Divide a complex problem into several **subproblems**:  
Very hard to achieve in mobile robots (difficulty of **cancatenation** of subroutines).

# What Makes the Difference ? (4)

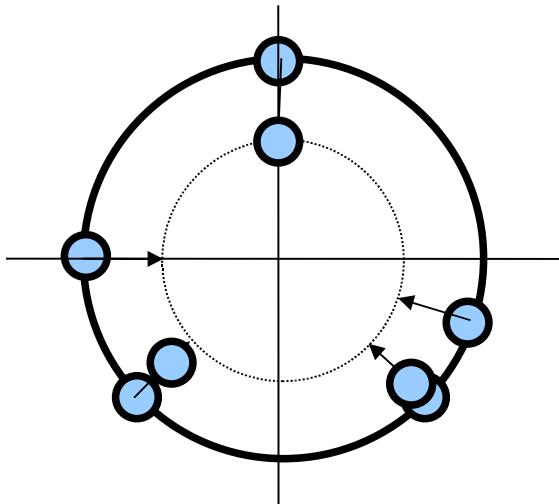
## NON ATOMIC Movements

A robot can be stopped by the scheduler before it reaches its calculated destination.

$x_i = a$  (atomic)

$\neq$

**Position<sub>i</sub>=point\_a** (non atomic)



- Instruct the robots to move to the internal concentric circle.
- **Seems simple.**
- But impossible to achieve even if the system is fully synchronous.

## Oblivious robots convergence

For every  $\epsilon > 0$ , there is a time  $t_\epsilon$  after which all correct robots are within distance of at most  $\epsilon$  of each other.

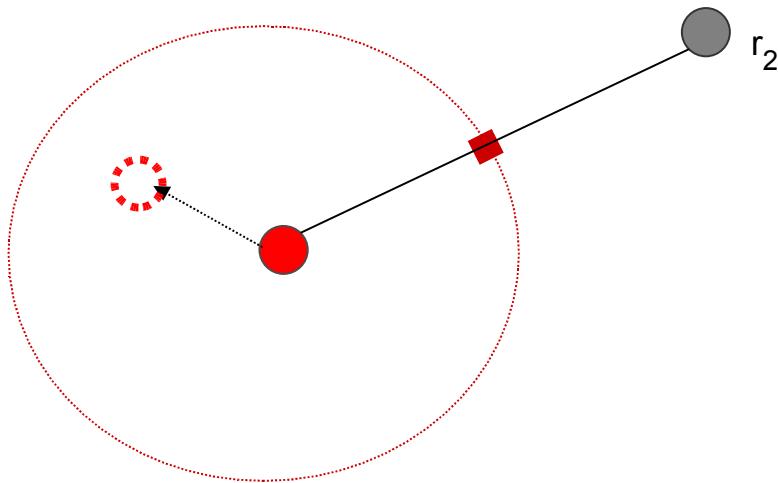
- **Distributed approximate agreement:**  
 $N > 3f$  for asynchronous/synchronous systems.
- **Mobile robots:**  
Fully Sync:       **$N > 2f$**   
Semi Sync:       **$N > 3f$**     +    **k-bounded**  
                      scheduler  
Async    :       **$N > 4f$**     +    **k-bounded** scheduler

# Oblivious robots scattering

Two properties:

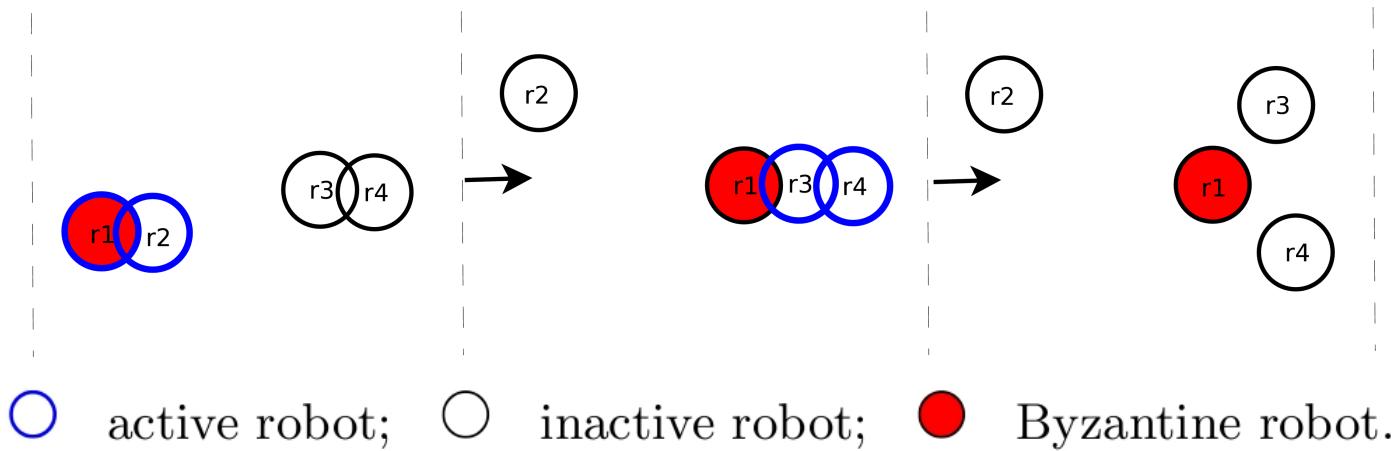
- **Convergence** – regardless of the initial position of the robots on the plane, no two robots are eventually located at the same position.
- **Closure** – starting from a configuration where no two robots are located at the same position, no two robots are located at the same position thereafter.

# Probabilistic scattering



# Byzantine Tolerant Scattering

- System under a *fair* scheduler:
- $(n, 1)$ ;



Time of convergence of weak scattering:  $(n - 1) + \frac{1}{kd - 1}$

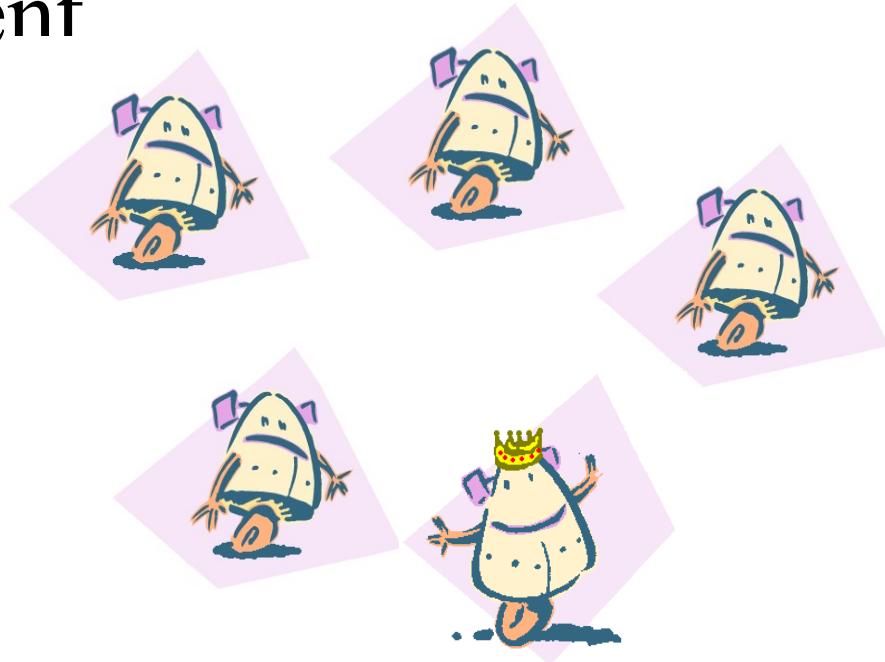
# Election d'un chef et déplacement de groupe

- Entrée : Un groupe de robots répartis aléatoirement dans un plan.



# Election d'un chef et déplacement de groupe

- **Objectif :** Il existe un unique chef au bout d'un temps fini et tous les robots le reconnaissent



# Flocking à travers l'élection d'un chef

L'algorithme a trois phases :

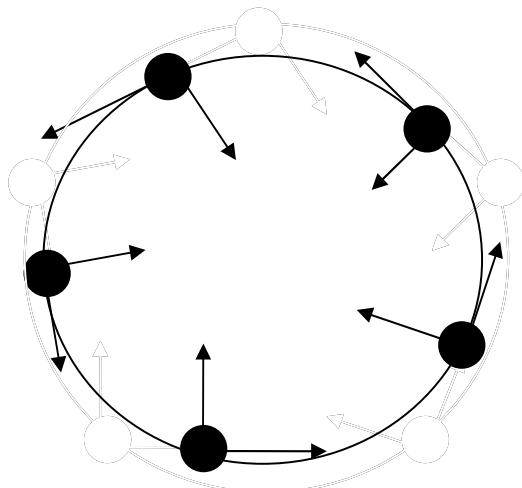
- **Election de chef**
- **Positionnement afin de préserver le chef**
- **Flocking**

# Résultat d'impossibilité

Soit un ensemble de robot dans un polygon.

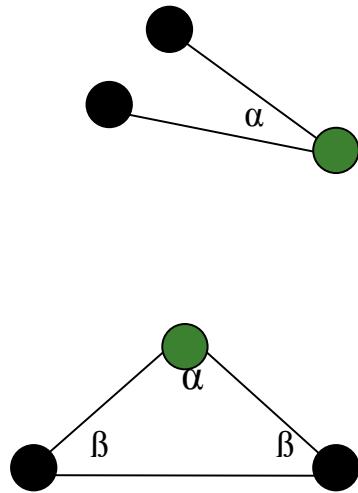
Tous les robots ont la même vue.

Si tous les robots bougent en même temps, alors on obtient une configuration symétrique.



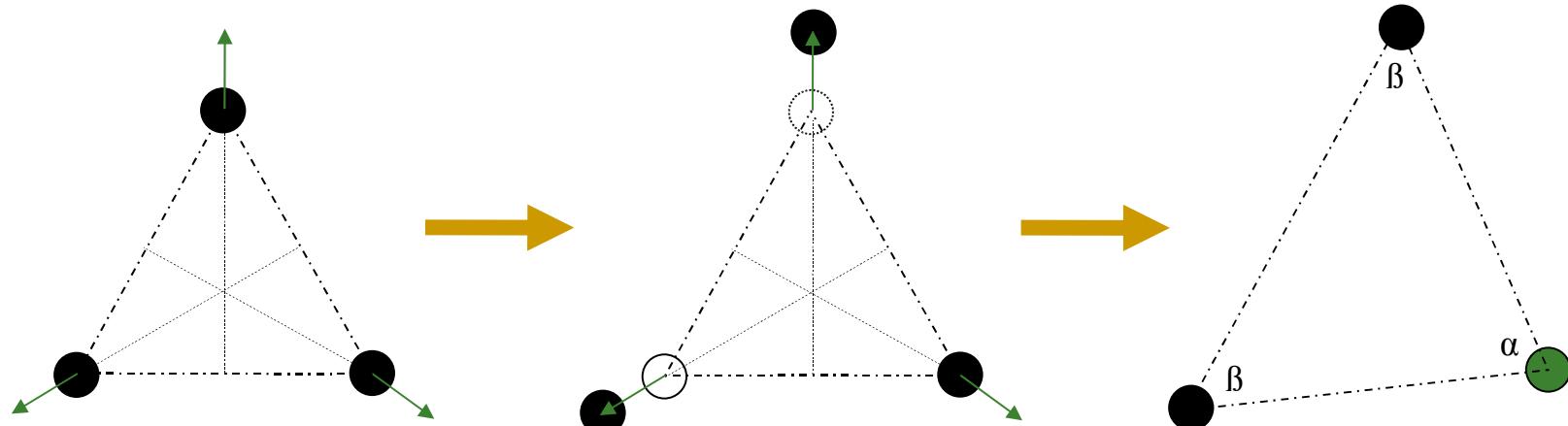
# Election de chef pour 3 robots

Si mon angle est le plus petit ou s'il est différent des autres je suis leader.

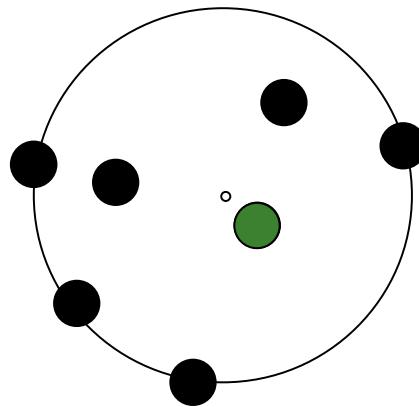
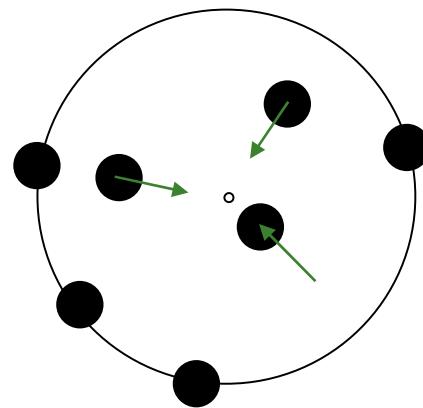
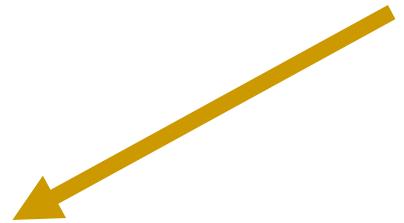
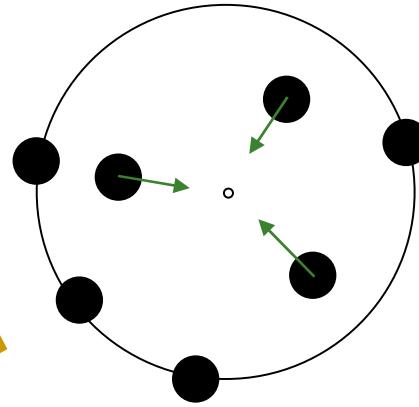
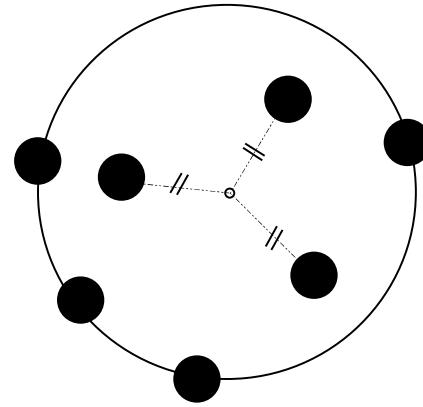


# Election de chef pour 3 robots

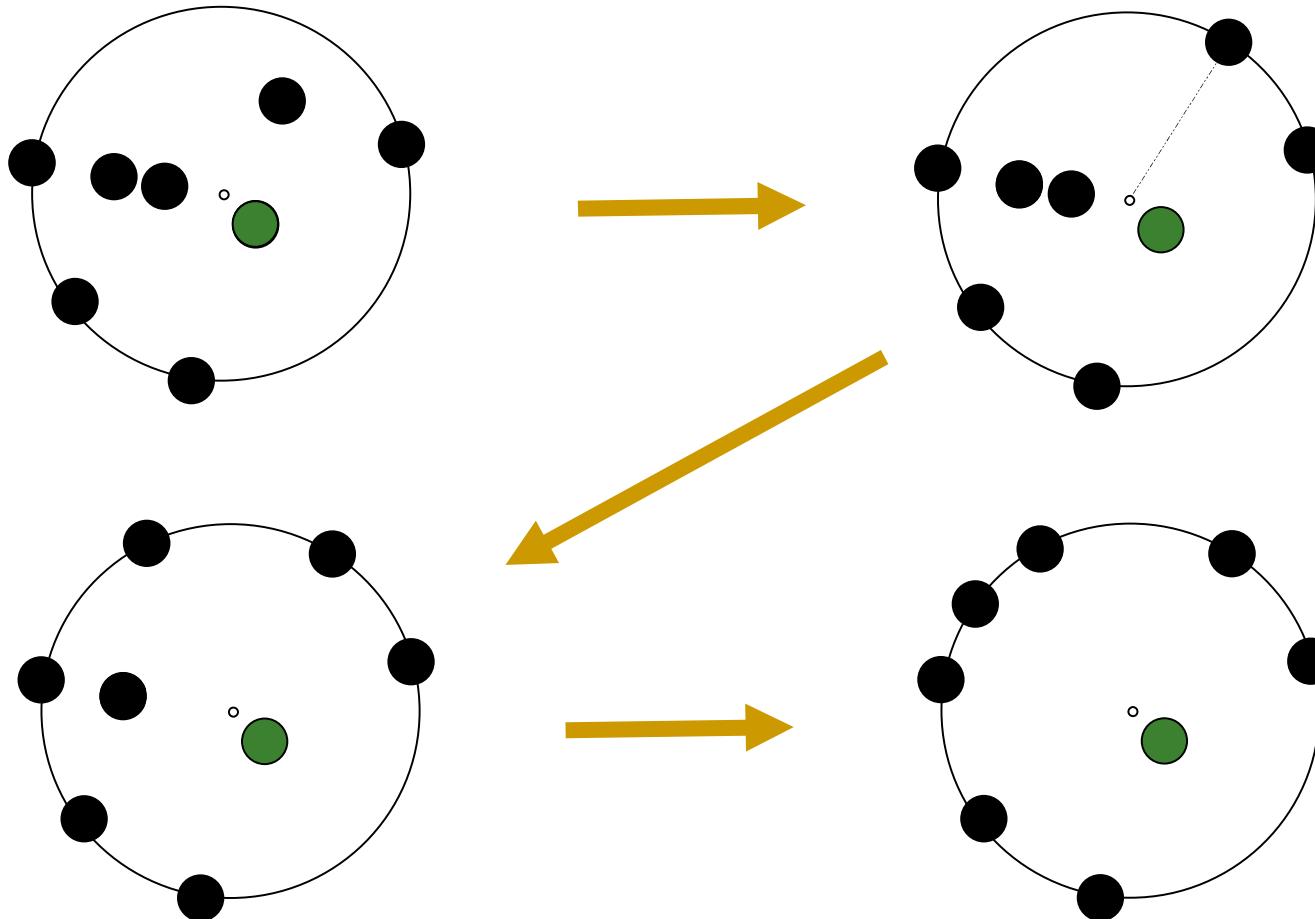
- Au cas contraire, les robots reculent perpendiculairement au segment liant les deux autres robots avec probabilité  $1/3$



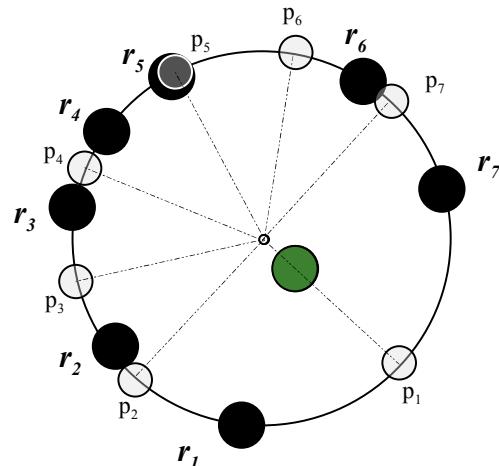
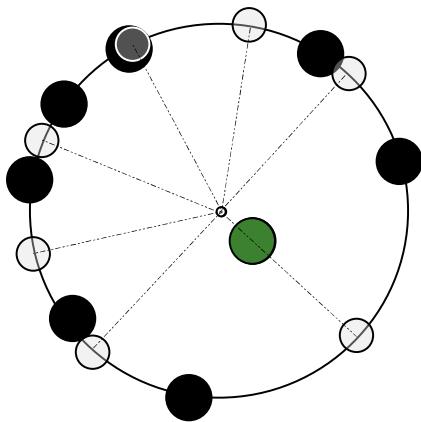
# Election de chef pour n robots



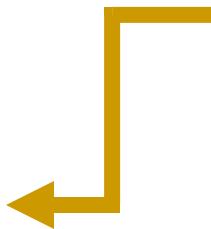
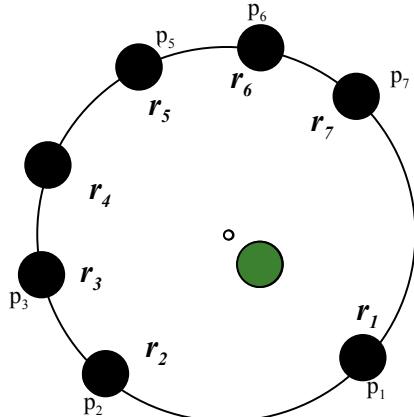
# Positionnement des robots sur le SEC



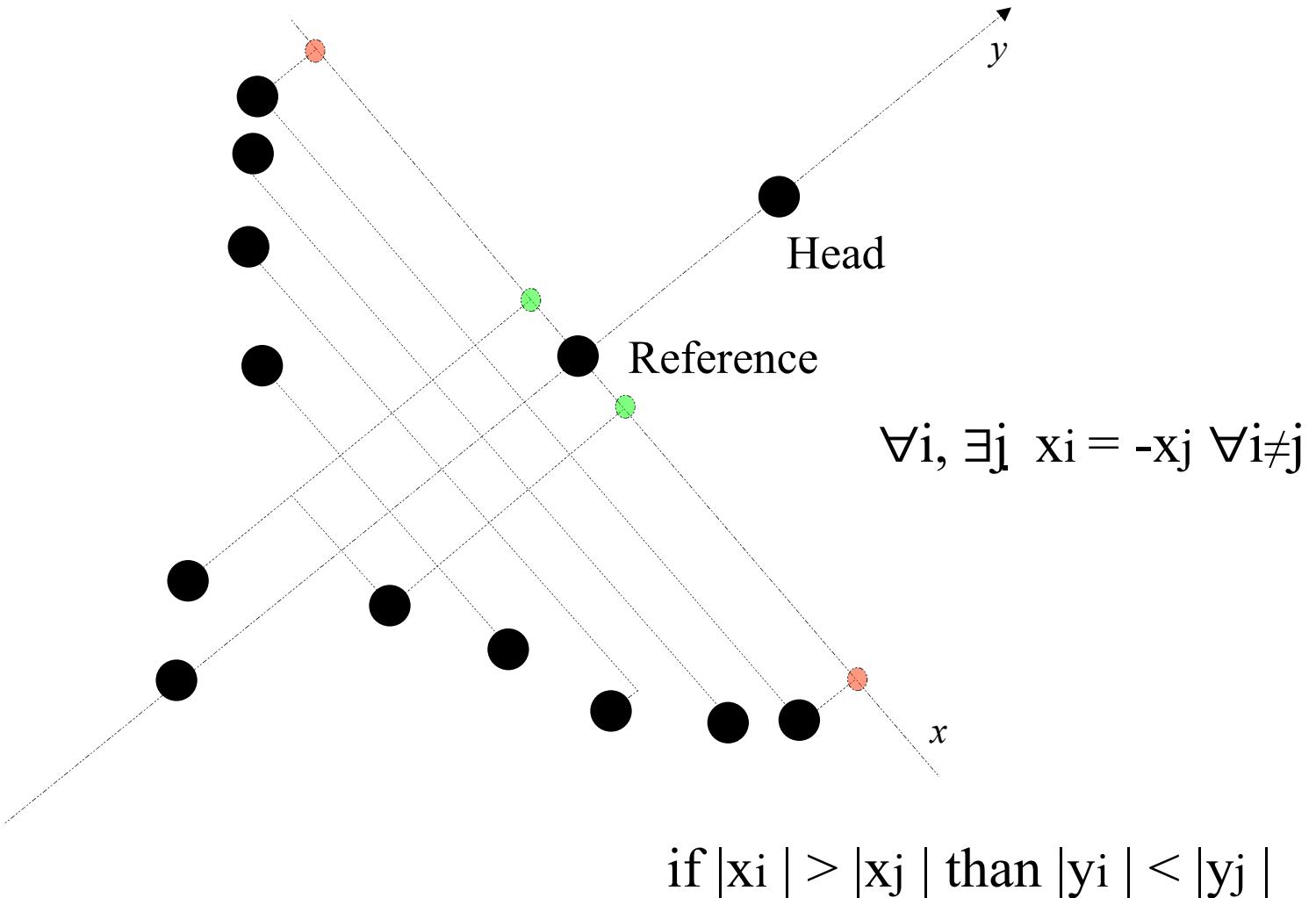
# Formation circulaire



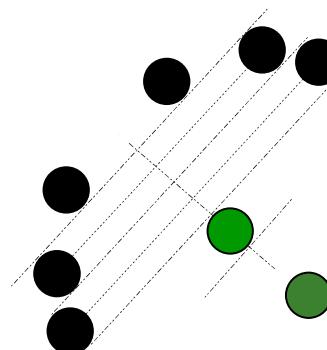
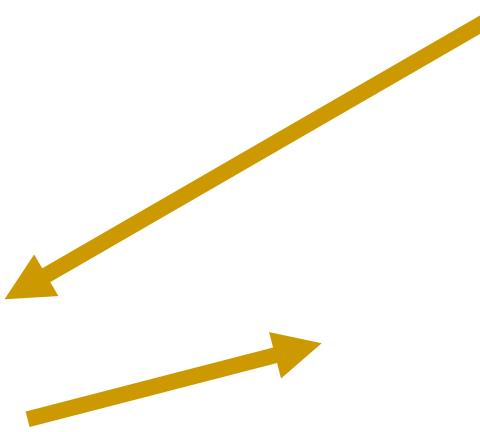
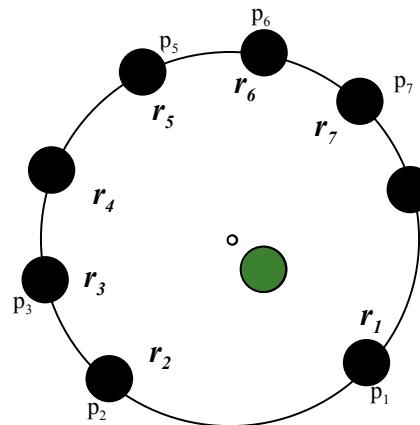
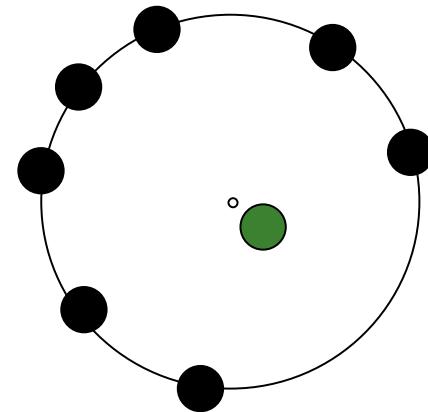
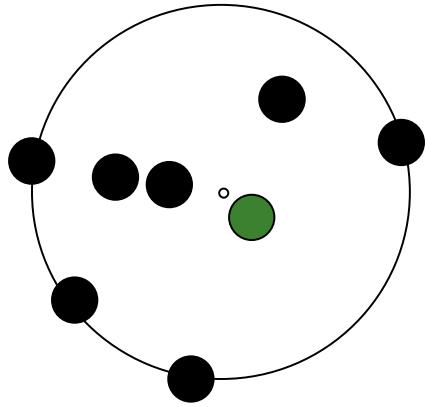
Formation de flocking



# Formation de déplacement



# Algorithme global



# Problèmes ouverts

- Étudier le modèle des robots avec volume
- Introduire la communication afin de contourner les résultats d'impossibilité
- Étudier les problèmes
  - d'allocation de ressources
  - élection de chef en étudiant plusieurs stratégies
  - scattering
  - déplacement en groupe quand le leader change de direction
  - déplacement en groupe tout en préservant la formation
  - recherche dans un plan