Rendez-vous d'agents mobiles

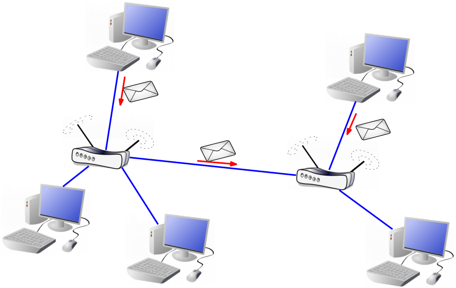
Arnaud Labourel

Équipe d'algorithmique distribuée
LIF, CNRS & Aix-Marseille Université, France

Visite de l'ENS Cachan, le 19 décembre 2013

LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Algorithmique distribuée

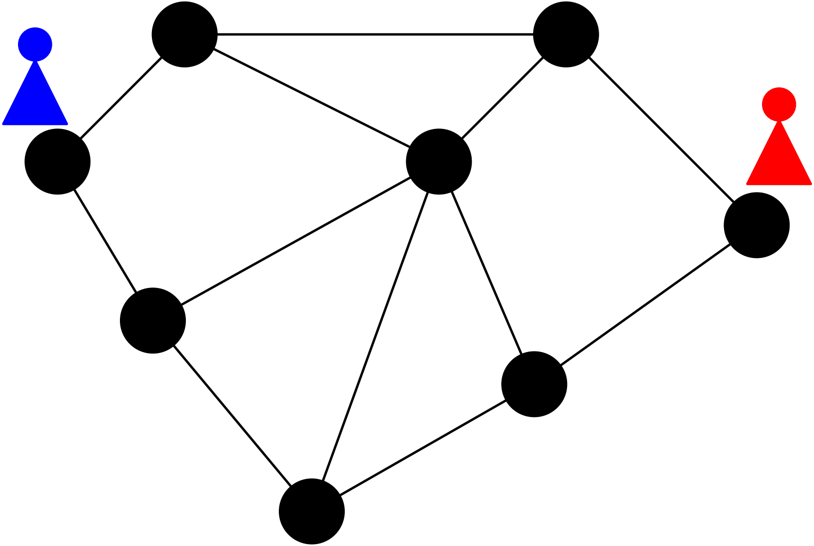


Un système distribué est un ensemble d'entités autonomes de calcul qui peuvent communiquer entre elles.

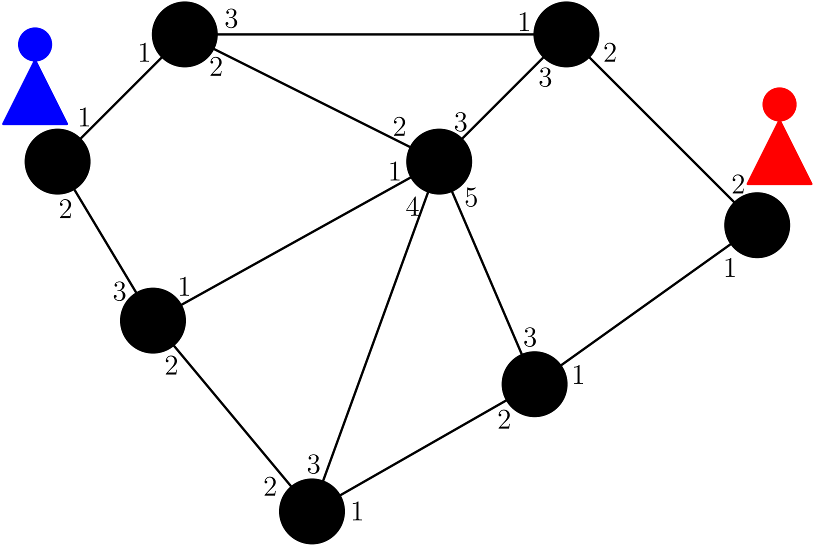
Algorithmique distribuée

Concevoir des algorithmes pour que ces entités collaborent afin d'accomplir une tâche.

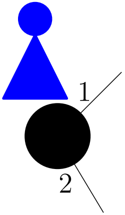
Rendez-vous d'agents mobiles



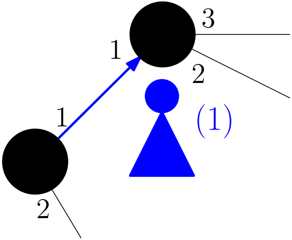
Rendez-vous d'agents mobiles



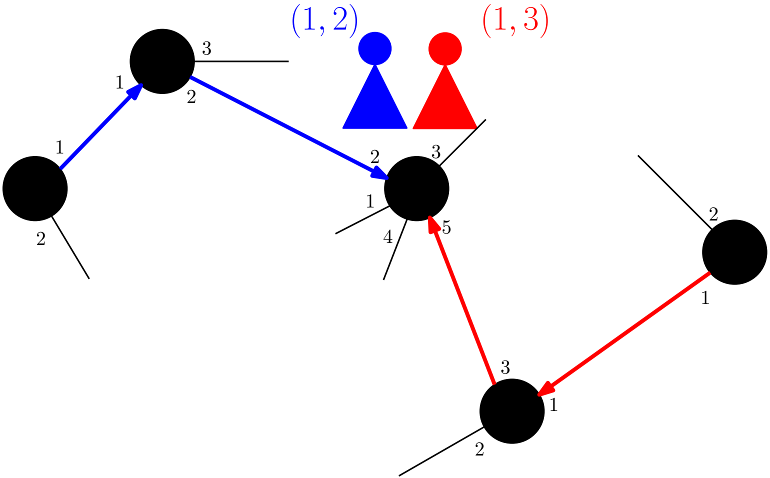
Rendez-vous d'agents mobiles



Rendez-vous d’agents mobiles



Rendez-vous d'agents mobiles



Le problème du rendez-vous

Le problème

Deux agents mobiles doivent se rencontrer dans un réseau (être sur le même nœud au même moment).

Modélisation du problème :

- Réseaux $\rightarrow$ graphes
- Agents mobiles $\rightarrow$ agents autonomes se déplaçant de nœud en nœud
- Complexité $\rightarrow$ temps mis par les agents pour se rencontrer

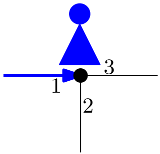
Navigation dans le graphe

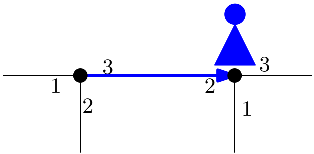
Les arêtes incidentes à un nœud sont numérotées localement (avec des numéros de port)

Déplacement des agents

À chaque étape :

- choix d'un numéro de port sur le nœud courant
- déplacement le long de l'arête correspondante
- récupération du numéro de port de l'arête dans le nouveau nœud





Modèle synchrone

Étape :

- choix du numéro de port
- déplacement sur un nœud voisin

Synchrone

La durée des étapes est la même pour les deux agents.
$\rightarrow$ Ils se déplacent toujours en même temps

De nombreux modèles possibles

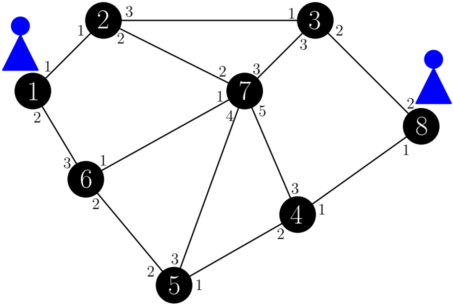
- Réseau anonyme ou pas
  (identifiants sur les nœuds)
- Agents anonymes ou pas
  (les agents ont le même algorithme ou pas)
- Connaissance préalable du réseau par l'agent ou pas
  (par exemple connaissance de la taille du graphe)
- Mémoire disponible pour l'agent
  (quantité de mémoire en bits conservée par l'agent après chaque déplacement)

Rendez-vous dans un réseau non-anonyme

Chaque nœud possède un identifiant unique (entier) visible par un agent sur le nœud.

Exercice

Est-ce qu'il existe un algorithme de rendez-vous pour les réseaux non-anonymes ?



Solution de l'exercice

Algorithme :

1. Chaque agent explore le graphe (avec un parcours en profondeur)
2. Chaque agent se rend sur le nœud ayant le plus petit identifiant

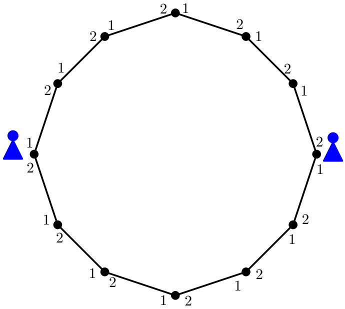
Complexité :

- $O(n)$ mouvements.
- mémoire en $O(n \log(n))$

Réseaux anonymes et agents anonymes

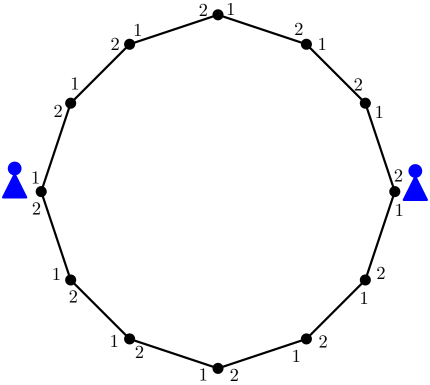
Exercice

Est-ce que le rendez-vous déterministe d'agents anonymes dans un anneau anonyme est possible ? Pas d'identifiants sur les nœuds et les agents exécutent le même algorithme.



Solution de l'exercice

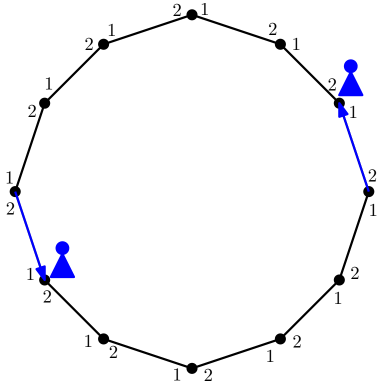
Les agents en positions symétriques voient la même configuration à chaque étape.
$\Longrightarrow$Ils se déplacent par le même numéro de port et restent en position symétrique.



Solution de l’exercice

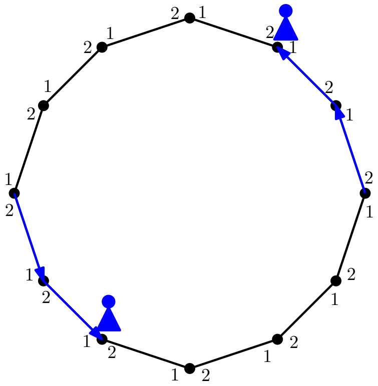
Les agents en positions symétriques voient la même configuration à chaque étape.

$\Longrightarrow$Ils se déplacent par le même numéro de port et restent en position symétrique.



Solution de l'exercice

Les agents en positions symétriques voient la même configuration à chaque étape.
$\Longrightarrow$Ils se déplacent par le même numéro de port et restent en position symétrique.



Agents avec identifiants

On suppose donc que les agents ont des identifiants distincts.
$\Rightarrow$ les agents ont des algorithmes différents suivant leurs identifiants.

Deux modèles pour les identifiants :

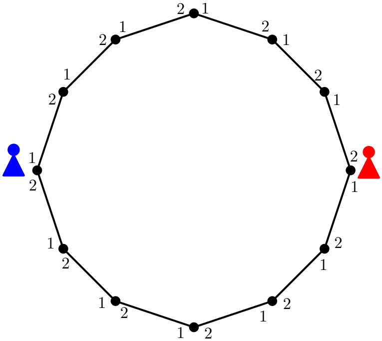
- Il y a que deux identifiants possibles : bleu et rouge
- Il y a une infinité d'identifiants possibles : des entiers ($\mathbb{N}$)

Deux identifiants possibles

Deux types d'agents : agent bleu et agent rouge

Exercice

Est-ce que le rendez-vous d'agents (rouge et bleu) dans un anneau anonyme est possible ?



Solution de l'exercice

Deux types d'agents : agent bleu et agent rouge

Stratégie attendre maman (waiting for mommy) :

- L'agent bleu attend
- L'agent rouge explore le graphe jusqu'à qu'il trouve l'agent bleu.

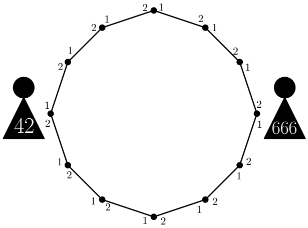
Cela revient à résoudre le problème d'exploration.
Dans l'anneau, l'agent rouge prend toujours le port 1 jusqu'à qu'il rencontre l'agent bleu.

Plus que deux identifiants

Les agents ont chacun un identifiant distinct choisi parmi une infinité de possibilités (entiers).

Exercice

Est-ce que le rendez-vous d'agents ayant des identifiants distincts dans un anneau anonyme est toujours possible si on suppose que les agents connaissent la taille n de l'anneau ?



Solution de l'exercice

Chaque agent a identifiant $id \in \mathbb{N}$

Algorithme :
Faire $id \times n$ déplacements par le port 1.

Preuve de correction :
Les deux agents se rencontrent car un des deux a fait un tour de plus que l'autre.

Complexité : $O(n \times \min(id_1, id_2))$ avec id_1 et id_2 les identifiants des deux agents.

Rendez-vous dans l'anneau

Il est possible d'obtenir une meilleure complexité.

Théorème [Dessmark, Fraigniaud, Pelc 2003]

Il existe un algorithme de rendez-vous déterministe en $O(D\log(\min\{id_1, id_2\}))$ mouvements dans l'anneau.

D : distance de départ entre les deux agents

Remarques :

- C'est optimal
  (borne inférieure en $\Omega(D\log(\min\{id_1, id_2\})))$.
- On n'a pas besoin de connaitre la taille de l'anneau.

Préliminaire

Pour chaque identifiant on construit un mot binaire :

- brique de base : représentation de l'identifiant *id* en binaire.
  Exemple : $15 \rightarrow 1111$, $16 \rightarrow 10000$
- On rajoute des 0 pour que le mot soit de longueur égale à une puissance de 2.
  $\Longrightarrow$ longueur du mot $= 2^{\lfloor \log \log id \rfloor + 1}$
  Exemple : $15 \rightarrow 1111$, $16 \rightarrow 00010000$

On note id^* le mot obtenu

Agents avec des identifiants de tailles similaires (algorithme 1)

Algorithme de rendez-vous dans l'anneau pour des agents id_1 et id_2 tels que $\lfloor \log \log id_1 \rfloor = \lfloor \log \log id_2 \rfloor$

Pour i de 1 à l'infini **faire**
  Pour j de 1 à taille de id^* **faire**
    Si le j-ème bit de id^* est 1 **alors**
      se déplacer de 2^i pas dans une direction
      se déplacer de 2^{i+1} pas dans la direction opposée
      revenir au point de départ
    Sinon attendre 2^{i+2}

Preuve de l'algorithme 1

On a $|id_1^*| = |id_2^*|$ et on est synchrone
$\Longrightarrow$les deux agents ont les mêmes i et j à tout moment.

Puisque les identifiants sont différents il existe un k tel que le k-ème bit de id_1^* est différent de celui de id_2^*.
Pour $i = \lceil \log D \rceil$ et $j = k$, les deux agents se rencontrent car :

- un agent atteint la position de départ de l'autre agent
- l'autre agent ne bouge pas et reste sur sa position de départ

Complexité : $O(D \log(\min\{id_1, id_2\}))$

Agents avec des identifiants de tailles différentes (algorithme 2)

$$id^+ = \begin{cases} 1 & \text{si } id = 1 \\ \lfloor \log \log id \rfloor + 2 & \text{autrement} \end{cases}$$

Algorithme de rendez-vous dans l'anneau pour des agents id_1 et id_2 tels que $\lfloor \log \log id_1 \rfloor \neq \lfloor \log \log id_2 \rfloor$

Pour i de 1 à l'infini **faire**
  Pour j de 1 à i **faire**
    Si $j = i - id^+$ **alors**
      se déplacer de 2^j pas dans une direction
      se déplacer de 2^{j+1} pas dans la direction opposée
      revenir au point de départ
    Sinon attendre 2^{j+2}

Preuve de l'algorithme 2

On suppose $id_1^+ < id_2^+$.

Durant la i-ème itération de la boucle, l'agent 1 parcourt tous les nœuds de l'anneau à distance $\leq 2^{i-id_1^+}$.

Pour le plus petit s tel que $2^{s-id_1^+} \geq D$, l'agent 1 rencontre l'agent 2 puisque celui-ci attend sur sa position de départ car $id_1^+ \neq id_2^+$.

Complexité :
$O(2^s) = 2^{id_1^+ + \log D} = O(D \log(\min\{id_1, id_2\}))$

Algorithme final

Idée

Alterner les exécutions des deux algorithmes

Algorithme final : **Pour** i de 1 à l'infini **faire**

- Exécuter l'algorithme 1 pendant 2^i étapes
- Revenir sur la position de départ en t étapes
- Attendre $2^i - t$ étapes
- Exécuter l'algorithme 2 pendant 2^i étapes
- Revenir sur la position de départ en t étapes
- Attendre $2^i - t$ étapes

Borne inférieure en $\Omega(\log(\min\{id_1, id_2\})$

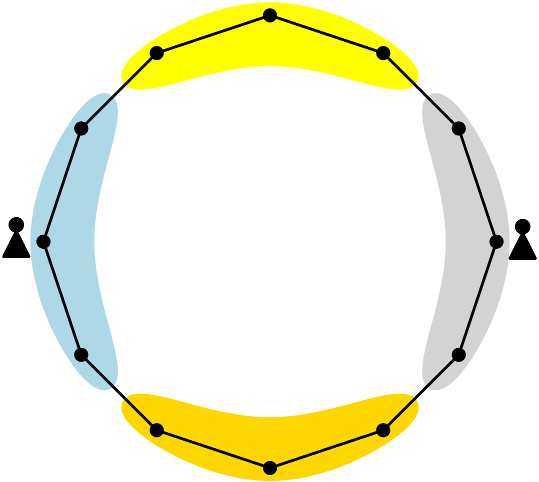
Théorème [Dessmark, Fraigniaud, Pelc 2003]

Tout algorithme de rendez-vous déterministe a une complexité de $\Omega(D \log(\min\{id_1, id_2\}))$ mouvements dans l'anneau.

L'algorithme de rendez-vous vu précédemment est donc optimal à un facteur multiplicatif constant près.

Idée de la preuve

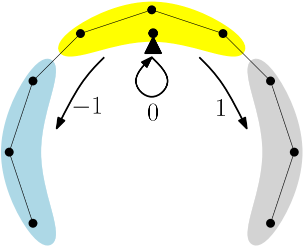
On découpe l'anneau en tranches de $D/2$ nœuds.



Idée de la preuve

On regarde la position de l'agent tout les $D/2$ étapes.

- 0 : il est dans la même tranche
- 1 : il est dans la tranche de droite
- -1 : il est dans la tranche de gauche



Cela nous donne un code de comportement (par exemple $(0, 1, -1, 1, \dots)$)

Idée de la preuve

Par l'absurde, il existe un algorithme résolvant le rendez-vous pour $Dy/4$ mouvements avec $y \in \mathbb{N}$ pour tous identifiants $id_1, id_2 \leq 2^y$.

Il y a $3^{y/2} < 2^y$ codes de comportement de longueur $y/2$.

On peut trouver deux identifiants $id_1, id_2 \leq 2^y$ qui ont le même code de comportement de longueur $y/2$.

Ces deux agents ne se rencontrent pas avant $Dy/4$ mouvements.

Contradiction
$\Longrightarrow$complexité en $\Omega(Dy) = \Omega(D \log(\min\{id_1, id_2\}))$