SDR

Diffusion

Algorithme de Chang et Roberts

Olivier Lemer

Suppositions

Pour l'instant, on supposera

- **Pas de perte** de message
- **Pas de duplication** de message
- **Pas de changement d'ordre** de messages
- **Pas de panne** serveur

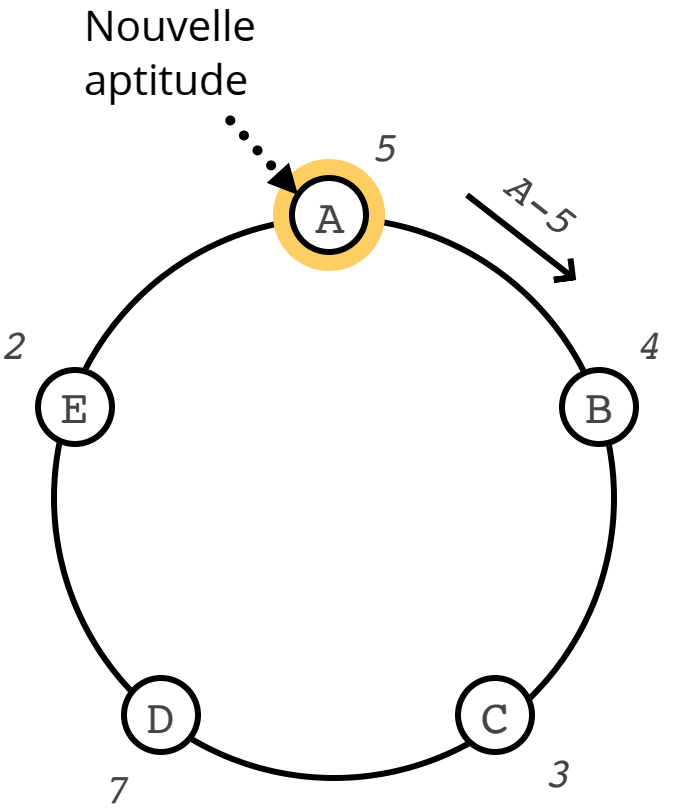
- **Système asynchrone** : délai d'envoi non borné

Chang et Roberts

Pour envoyer moins de messages

Chang et Roberts, 1979

L'initiateur envoie son aptitude et son id.



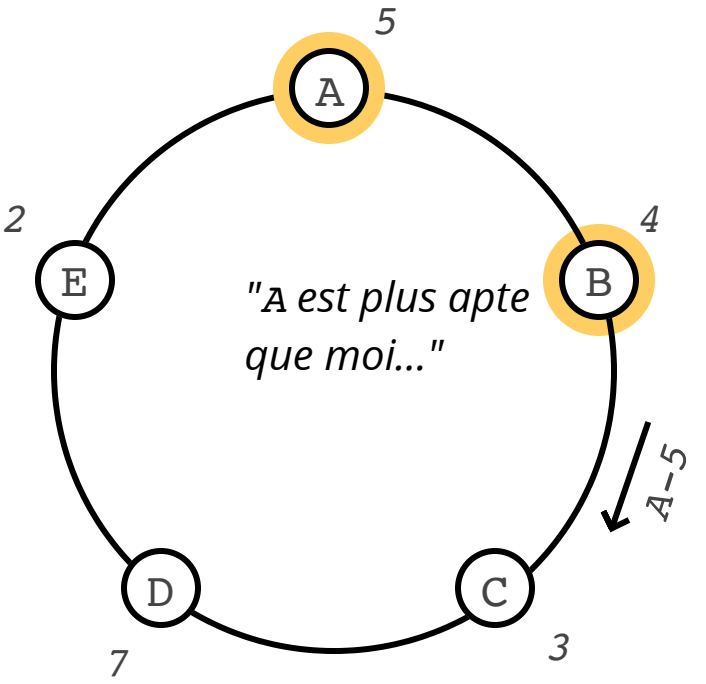
En élection

Connait l'élu

Chang et Roberts, 1979

L'initiateur envoie son aptitude et son id.

Si je reçois une aptitude d'un autre
- Je la compare à la mienne
- Je propage la plus grande des deux



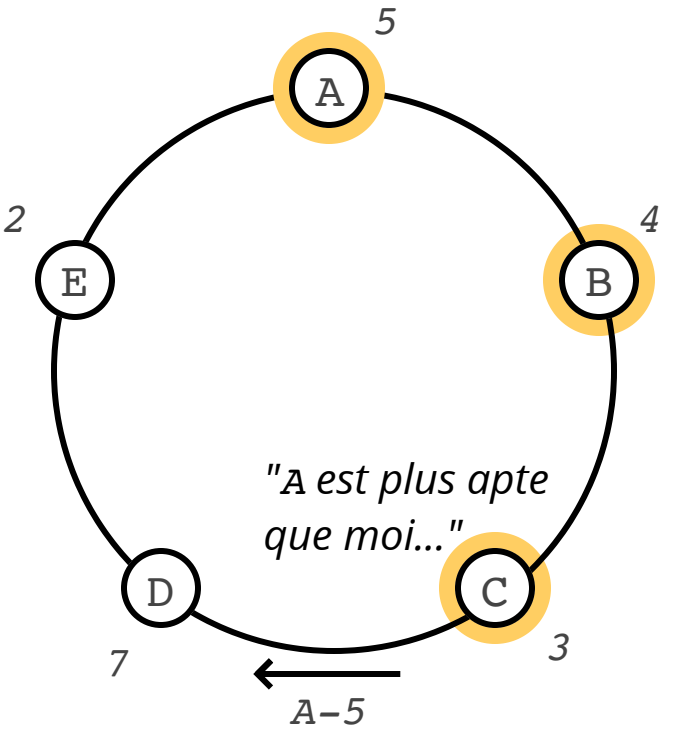
En élection

Connait l'élu

Chang et Roberts, 1979

L'initiateur envoie son aptitude et son id.

Si je reçois une aptitude d'un autre
- Je la compare à la mienne
- Je propage la plus grande des deux



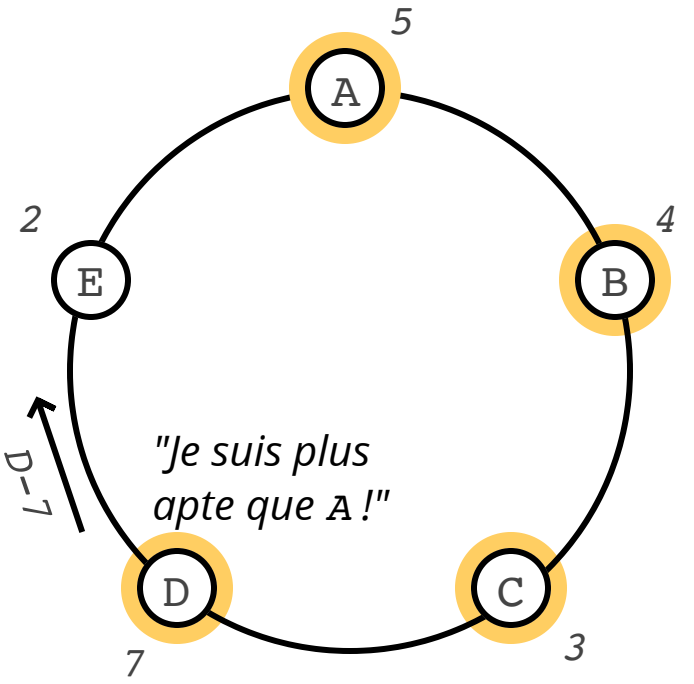
En élection

Connait l'élu

Chang et Roberts, 1979

L'initiateur envoie son aptitude et son id.

Si je reçois une aptitude d'un autre
    Je la compare à la mienne
    Je propage la plus grande des deux



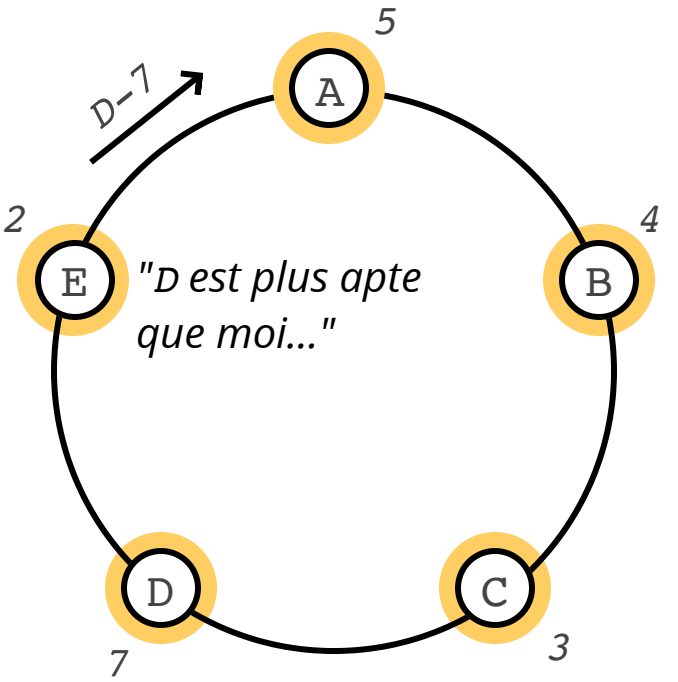
En élection

Connait l'élu

Chang et Roberts, 1979

L'initiateur envoie son aptitude et son id.

Si je reçois une aptitude d'un autre
- Je la compare à la mienne
- Je propage la plus grande des deux



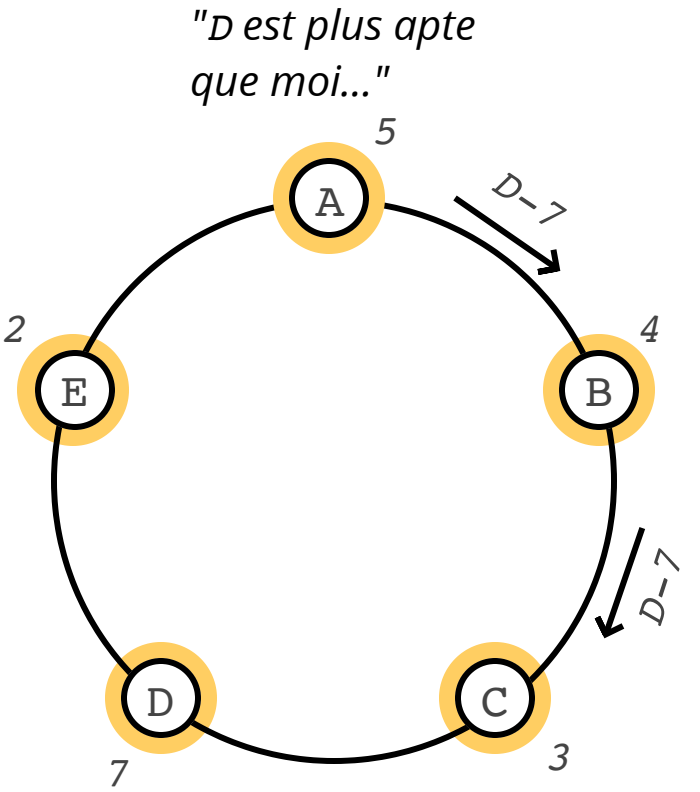
En élection

Connait l'élu

Chang et Roberts, 1979

L'initiateur envoie son aptitude et son id.

Si je reçois une aptitude d'un autre
- Je la compare à la mienne
- Je propage la plus grande des deux



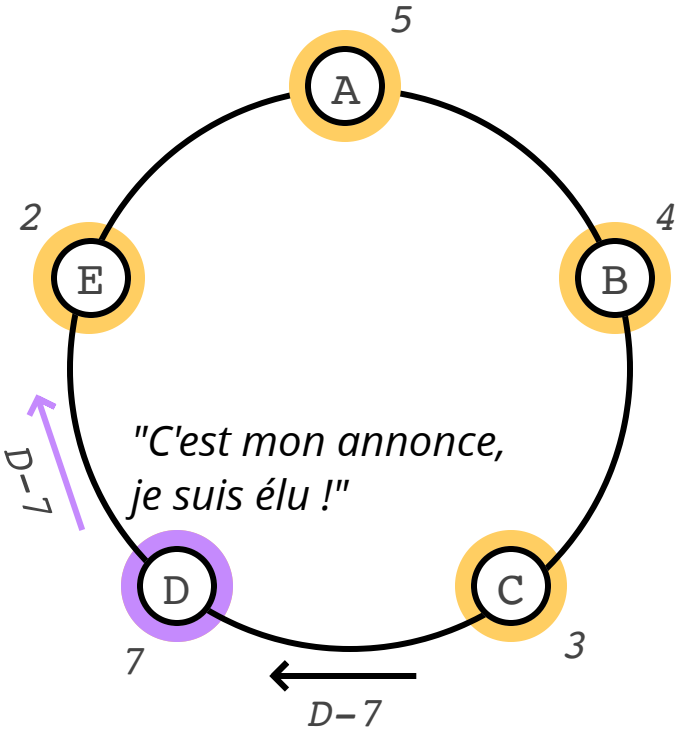
Chang et Roberts, 1979

L'initiateur envoie son aptitude et son id.

Si je reçois une aptitude d'un autre
- Je la compare à la mienne
- Je propage la plus grande des deux

Si je reçois ma propre aptitude
- L'annonce a fait le tour, je suis élu !
- Je propage le résultat

Tour d'annonce



En élection

Connait l'élu

Chang et Roberts, 1979

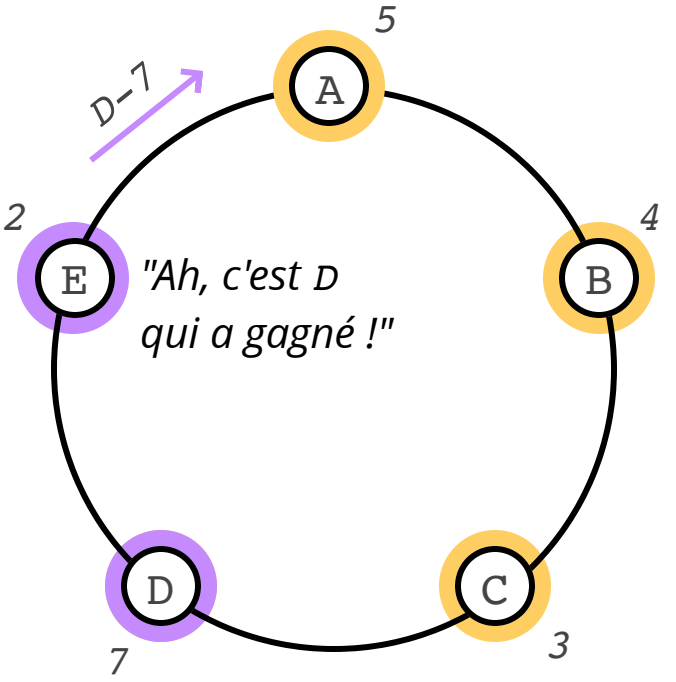
L'initiateur envoie son aptitude et son id.

Si je reçois une aptitude d'un autre
  Je la compare à la mienne
  Je propage la plus grande des deux

Si je reçois ma propre aptitude
  L'annonce a fait le tour, je suis élu !
  Je propage le résultat

Si je reçois un résultat
  Je le propage

Tour d'annonce



Chang et Roberts, 1979

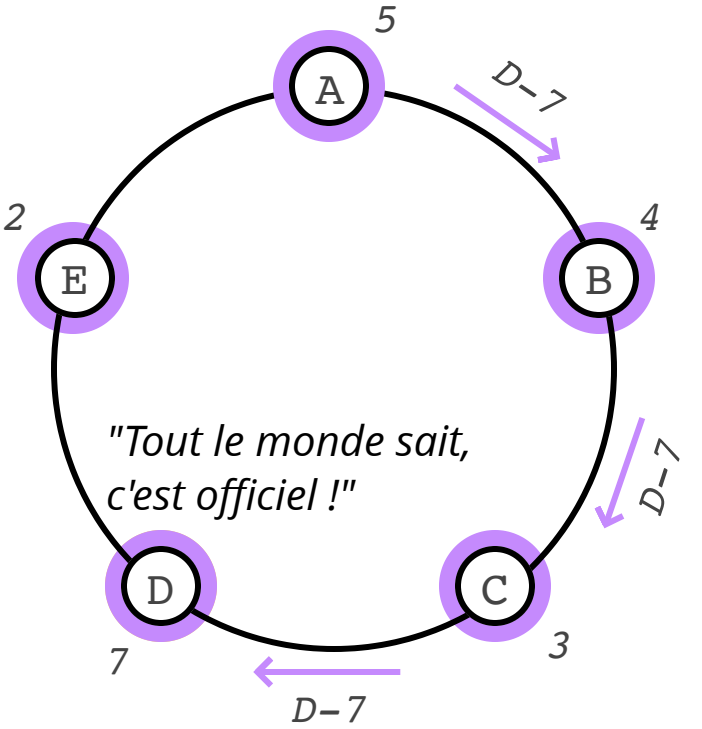
L'initiateur envoie son aptitude et son id.

Si je reçois une aptitude d'un autre
- Je la compare à la mienne
- Je propage la plus grande des deux

Si je reçois ma propre aptitude
- L'annonce a fait le tour, je suis élu !
- Je propage le résultat

Si je reçois un résultat
- Je le propage si ce n'est pas le mien
- Sinon, l'élection est finie, je suis l'élu

Tour d'annonce



Chang et Roberts, 1979

L'initiateur envoie son aptitude et son id.

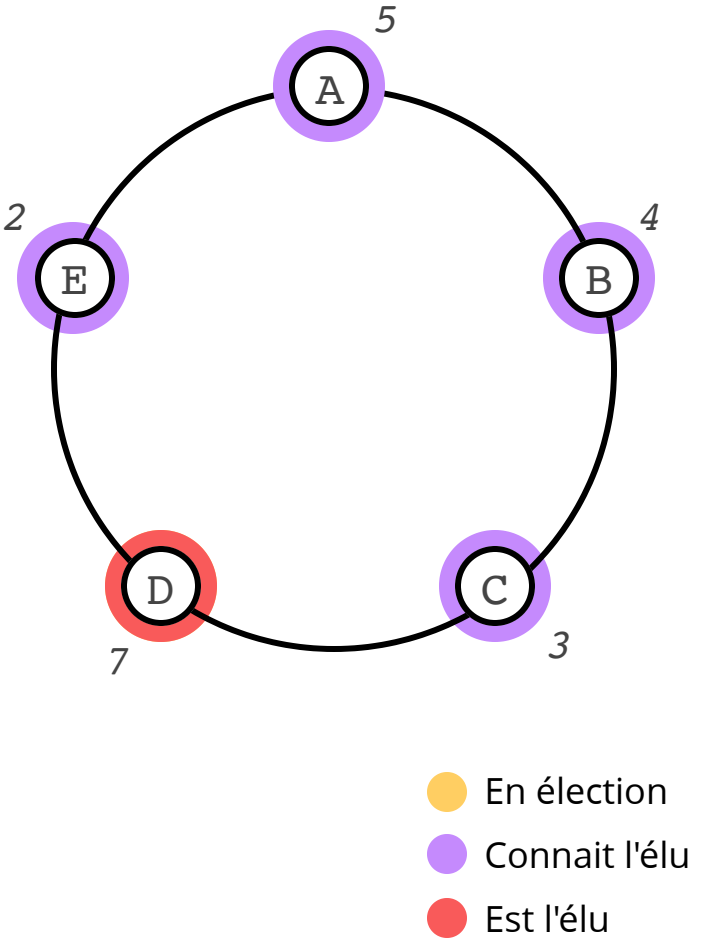
Si je reçois une aptitude d'un autre
  Je la compare à la mienne
  Je propage la plus grande des deux

Si je reçois ma propre aptitude
  L'annonce a fait le tour, je suis élu !
  Je propage le résultat

Si je reçois un résultat
  Je le propage si ce n'est pas le mien
  Sinon, l'élection est finie, je suis l'élu

Tour d'annonce

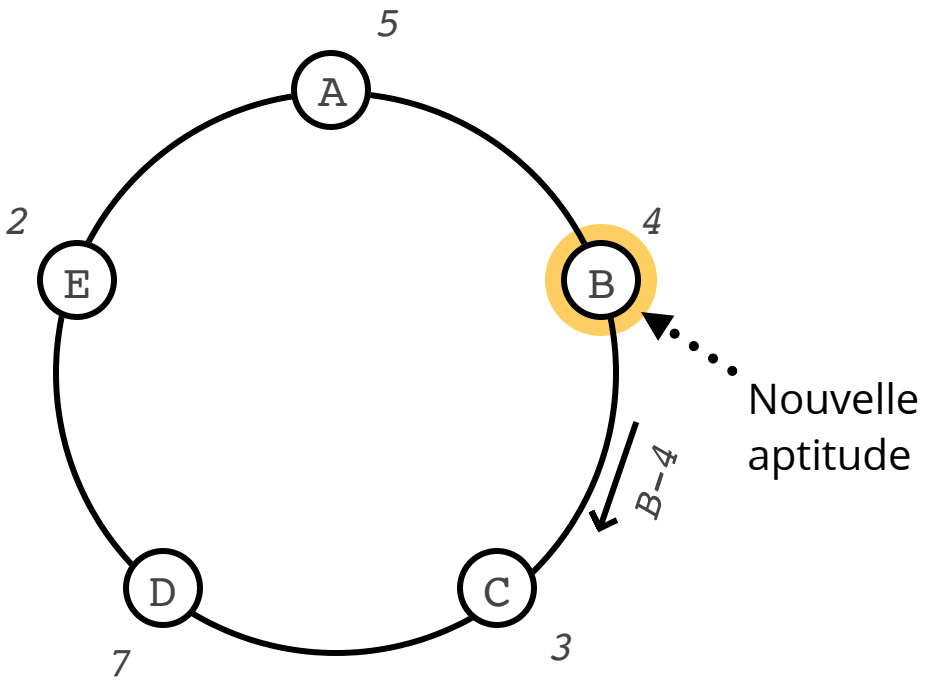
Tour de résultat



Chang et Roberts

Gestion d'élections concurrentes

Élections concurrentes

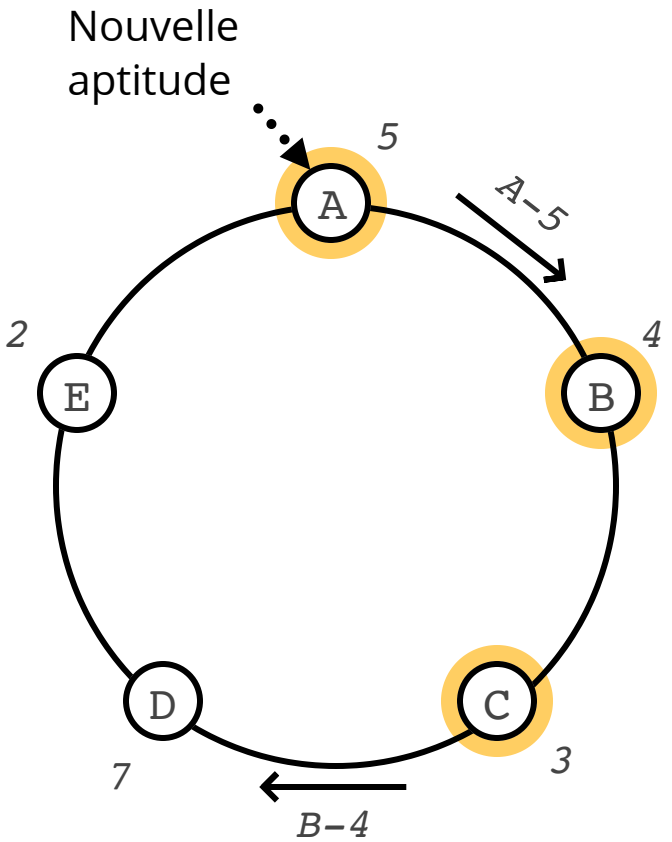


En élection

Connait l'élu

Est l'élu

Élections concurrentes

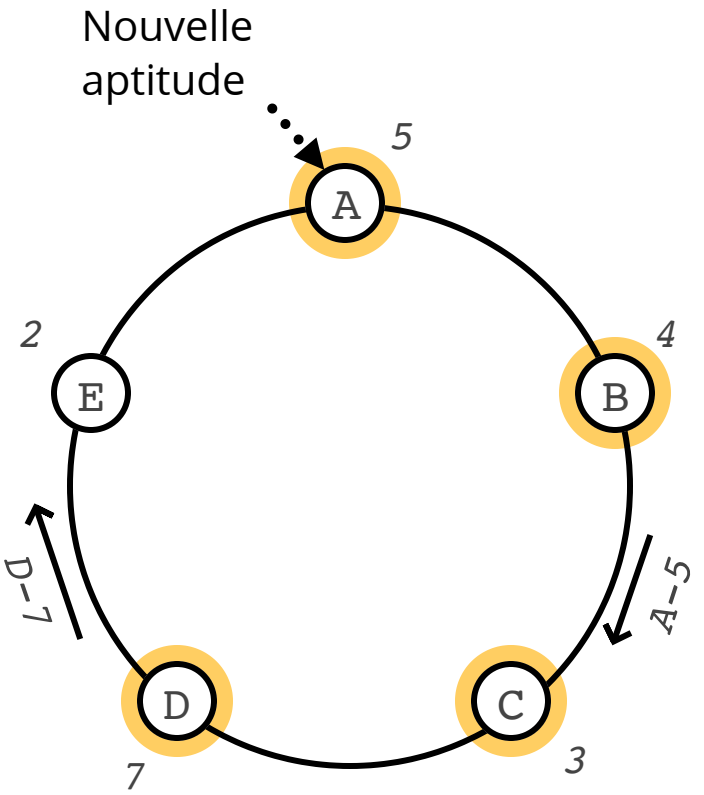


En élection

Connait l'élu

Est l'élu

Élections concurrentes

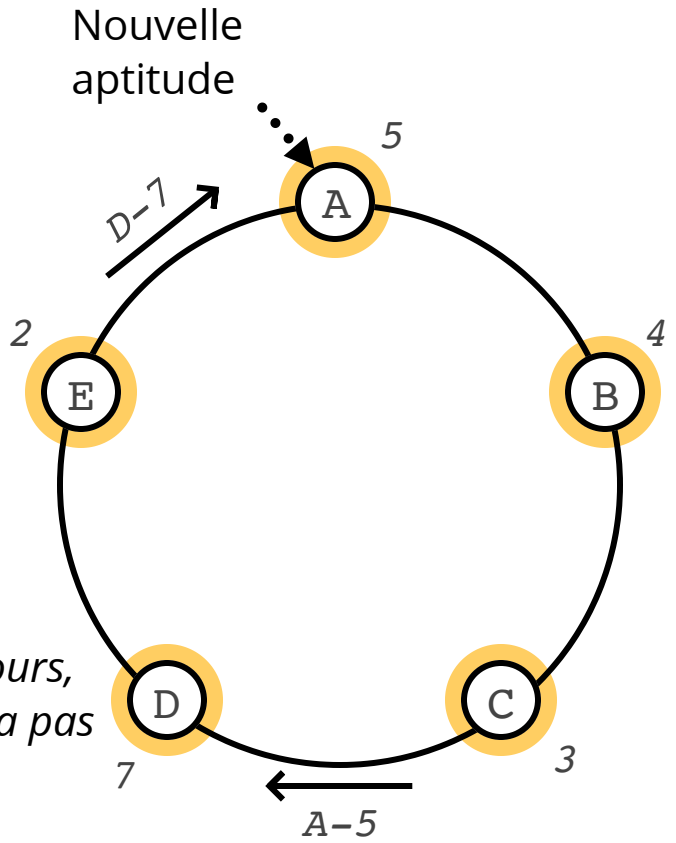


En élection

Connait l'élu

Est l'élu

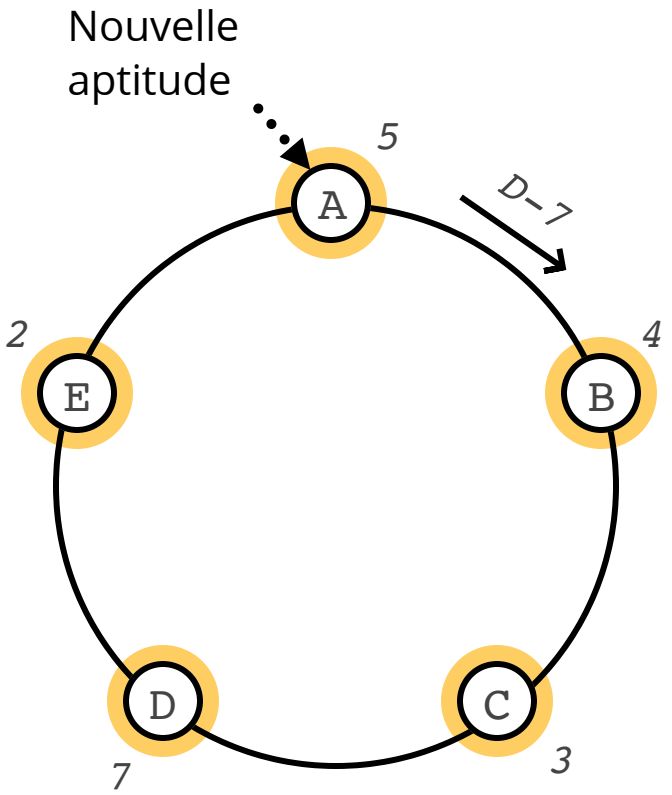
Élections concurrentes



"Une élection est déjà en cours, et cette nouvelle élection n'a pas trouvé mieux, je l'ignore."

En élection
Connait l'élu
Est l'élu

Élections concurrentes



En élection

Connait l'élu

Est l'élu

Élections concurrentes

Quand je suis déjà en cours d'élection,

Si je reçois une aptitude *plus basse*, je l'ignore

(Sinon, l'algorithme normal s'applique)

Chang et Roberts

Résumé et Performances

Résumé

Changement d'aptitude

Si je ne suis pas en cours d'élection

- J'adopte la nouvelle aptitude
- J'entre en élection
- J'envoie mon aptitude et mon ID

Si je suis en cours d'élection

- Je la mets de côté
  (je l'adopterai en sortant d'élection)

Réception d'un résultat

Si ce n'est pas moi

- Je note et je propage
- Je sors d'élection

Si c'est moi

- Je suis l'élu
- Je sors d'élection

Réception d'une aptitude

Si je ne suis pas en cours d'élection

- J'entre en élection
- Je compare l'aptitude reçue avec la mienne
- Je propage la plus grande, avec l'ID associé

Si je suis en cours d'élection

- Si c'est la mienne
  - J'envoie le résultat
- Si ce n'est pas la mienne
  - Je compare l'aptitude reçue avec la mienne
  - Je propage celle reçue ssi elle est plus grande.

Pourquoi pas un problème de ne rien envoyer sinon ?

Performances

Communication

~`3N` messages par élection

Durée d'une élection

~`3NT` pour `T` la durée de transit max

Chang et Roberts

Pseudocode

Pseudocode

Variables

`self`	*entier, constant*	mon numéro de processus
`apt_self`	*entier*	mon aptitude
`suivant`	*entier, constant*	numéro du suivant dans l'anneau
`inElection`	*booléen, init false*	ssi une élection est en cours
`élu`	*entier*	id du processus élu
`apt_pending`	*entier, init* $\perp$	aptitude mise de côté

Remarques

Plus besoin de stocker les aptitudes de tous les autres.

Plus besoin de dépendre de `T.`

Tous les envois se feront à `suivant`.

Pseudocode

Initialisation

Démarrer une élection

Écouter infiniment les événements suivants :

- Demande de changement d'aptitude à `new_apt`
- Demande d'obtention de l'élu par la couche applicative
- Réception d'une annonce `{i, apt_i}`
- Réception d'un résultat `{i}`

Note : les traitements de ces événements sont faits de manière séquentielle.

Pseudocode

Demande de changement d'aptitude à `new_apt`

```
Si inElection
    apt_pending ← new_apt
Sinon
    apt_self ← new_apt
    Démarrer une élection
```

Réception d'un résultat `{i}`

```
élu ← i
Si élu != self
    Envoi de résultat {i}
Sortir d'élection
```

Pseudocode

Démarrer une élection

```
Envoi d'une annonce {self, apt_self}
inElection ← true
```

Sortir d'élection

```
inElection ← false
Si apt_pending ≠ ⊥ et apt_pending ≠ apt_self
    apt_self ← apt_pending
    Démarrer une élection
apt_pending ← ⊥
```

Pseudocode

Demande d'obtention de l'élu

```
Retourner élu.
```

Réception d'une annonce `{i, apt_i}`

```
Si (apt_self, self) > (apt_i, i)
    Si pas inElection
        Démarrer une élection
Sinon, si i == self
    Envoi de résultat {i}
Sinon
    Envoi d'annonce {i, apt_i}
    inElection ← true
```