

Accord et coordination

Élection d'un maître

Master 1 RID

Accord et coordination

- ❑ Une famille de problèmes en algorithmique distribuée.
- ❑ Catégories de problèmes de cette famille :
 - ❑ Accord sur l'accès à une ressource partagée
 - ❑ **Exclusion mutuelle distribuée**
 - ❑ Accord sur l'ordre d'envoi de messages à tous
 - ❑ **Diffusion atomique**
 - ❑ Accord sur un processus jouant un rôle particulier
 - ❑ **Élection d'un maître**
 - ❑ Accord sur une valeur commune
 - ❑ **Consensus**
 - ❑ Accord sur une action à effectuer par tous ou personne
 - ❑ **Transaction**

Accord sur un processus jouant un rôle particulier
« Élection d'un maître »

Plan

- ❑ Élection : motivations
- ❑ Élection : principes
- ❑ Phases d'Élection
 - ❑ Phase de préparation
 - ❑ Phase de décision
 - ❑ Phase de proclamation du résultat
- ❑ Algorithmes d'Élections
 - ❑ **Algorithme de Chang et Roberts (1979)**

Élection : motivations

- ❑ De nombreux algorithmes de contrôle, bien que dits distribués, sont en fait centralisés dans le sens qu'un processus particulier joue le rôle de **coordonateur** et effectue certains services à la demande des autres processus.
 - ❑ C'est le schéma du modèle « **client-serveur** ».
- ❑ Ce type d'algorithme présente un inconvénient majeur : **la panne du processus coordonnateur** qui entraîne l'arrêt du fonctionnement du système.
- ❑ Il faut alors que les processus en état de fonctionnement puissent s'entendre pour décider lequel d'entre eux prend désormais le rôle de coordonnateur.
 - ❑ Cet accord s'obtient grâce aux **algorithmes d'élection**.

Élection : principes

- ❑ Les protocoles d'élection peuvent aussi être utilisés chaque fois qu'un processus, sans nécessairement être coordinateur, joue un rôle particulier.
- ❑ Les processus sont en général identifiés par des numéros et on décide que le processus qui joue un rôle à part est celui qui possède **le plus grand** numéro.
- ❑ Un algorithme d'élection a donc pour objectif de déterminer le processus de plus grand numéro.

Phases d'Élection

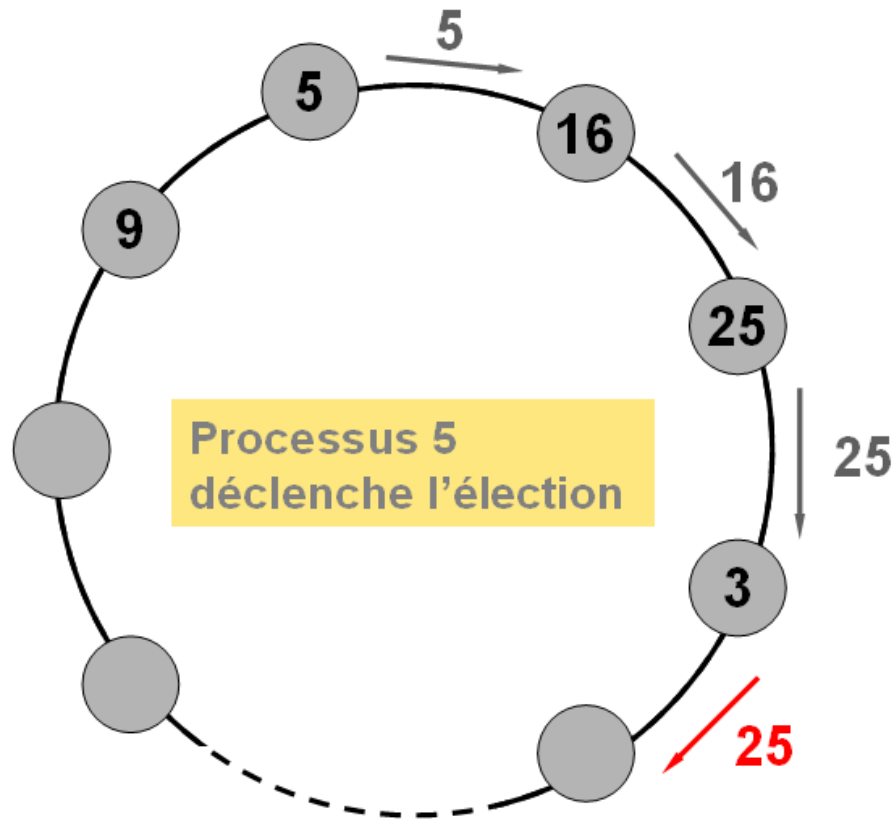
- ❑ **Phase de préparation**, dans laquelle sont déterminés les processus participant à l'élection ainsi que les candidats.
- ❑ **Phase de décision**, dans laquelle un processus candidat est désigné par un certain mécanisme.
 - ❑ Possédant le plus grand numéro, par exemple.
- ❑ **Phase de proclamation du résultat**, où chaque processus participant est informé du processus élu.

Algorithmes d'Élections

- ❑ **But** : élire un seul processus P_i parmi un groupe de processus $P_1 \dots P_n$.
- ❑ **Utilité** : élire un processus maître, un coordinateur ou un serveur central.
- ❑ Chaque processus P_i maintient une variable $\acute{E}lu_i$ (null s'il n'est pas l'élue).
- ❑ **Propriétés à satisfaire** : $\forall p_i$,
 - ❑ **Sûreté** : $\acute{E}lu_i = \text{null}$ ou $\acute{E}lu_i = \text{Vrai}$ (**processus élu**).
 - ❑ **Vivacité** : P_i participe et aura $\acute{E}lu_i = \text{null}$ ou $\acute{E}lu_i = \text{Vrai}$.

Algorithme de Chang et Roberts (1979)

- ❑ Topologie en anneau unidirectionnel entre les processus, où chaque processus sait adresser un message à son voisin de gauche.
- ❑ **Un ou plusieurs processus** peuvent décider de lancer un protocole d'élection.



Algorithme de Chang et Roberts (1979)

❑ Les messages sont de deux types :

- ❑ Les messages de type **élection** : un processus envoie un tel message soit quand il décide de lancer une élection, soit quand il devient participant.
- ❑ Les messages de type **élu** qui font circuler, après qu'il a été déterminé, le plus grand numéro de processus ; un processus qui reçoit un message de type élection portant son numéro sait qu'il est celui de plus grand numéro et envoie le premier message de type élu ; ce message est alors retransmis en faisant le tour de l'anneau.

Algorithme de Chang et Roberts (1979)

❑ Initialisation pour tous les processus :

- ❑ $\text{Participanti} := \text{FAUX}$

- ❑ $\text{Élui} := \text{Null}$

❑ Déclenchement d'une élection par P_i :

- ❑ $\text{Participanti} := \text{VRAI}$

- ❑ Envoie le message $\langle \text{élection}, P_i \rangle$ à son voisin

Algorithme de Chang et Roberts (1979)

Réception d'un message d'élection $\langle \text{élection}, P_i \rangle$ à P_j :

Si $P_i > P_j$ Alors

- Envoie le message $\langle \text{élection}, P_i \rangle$ à son voisin
- Participantj := VRAI

Sinon (//deux cas)

Si $(P_i < P_j \text{ ET Participantj} = \text{FAUX})$ Alors

- Alors Envoie le message $\langle \text{élection}, P_j \rangle$ à son voisin
- Participantj := VRAI

Sinon $(P_i = P_j)$

- Éluj := VRAI
- Participantj := FAUX
- Envoie le message $\langle \text{élu}, P_j \rangle$ à son voisin (initié la proclamation des résultats)

FinSi

FinSi

Algorithme de Chang et Roberts (1979)

❑ Réception d'un message $\langle \text{élu}, P_j \rangle$ à P_i :

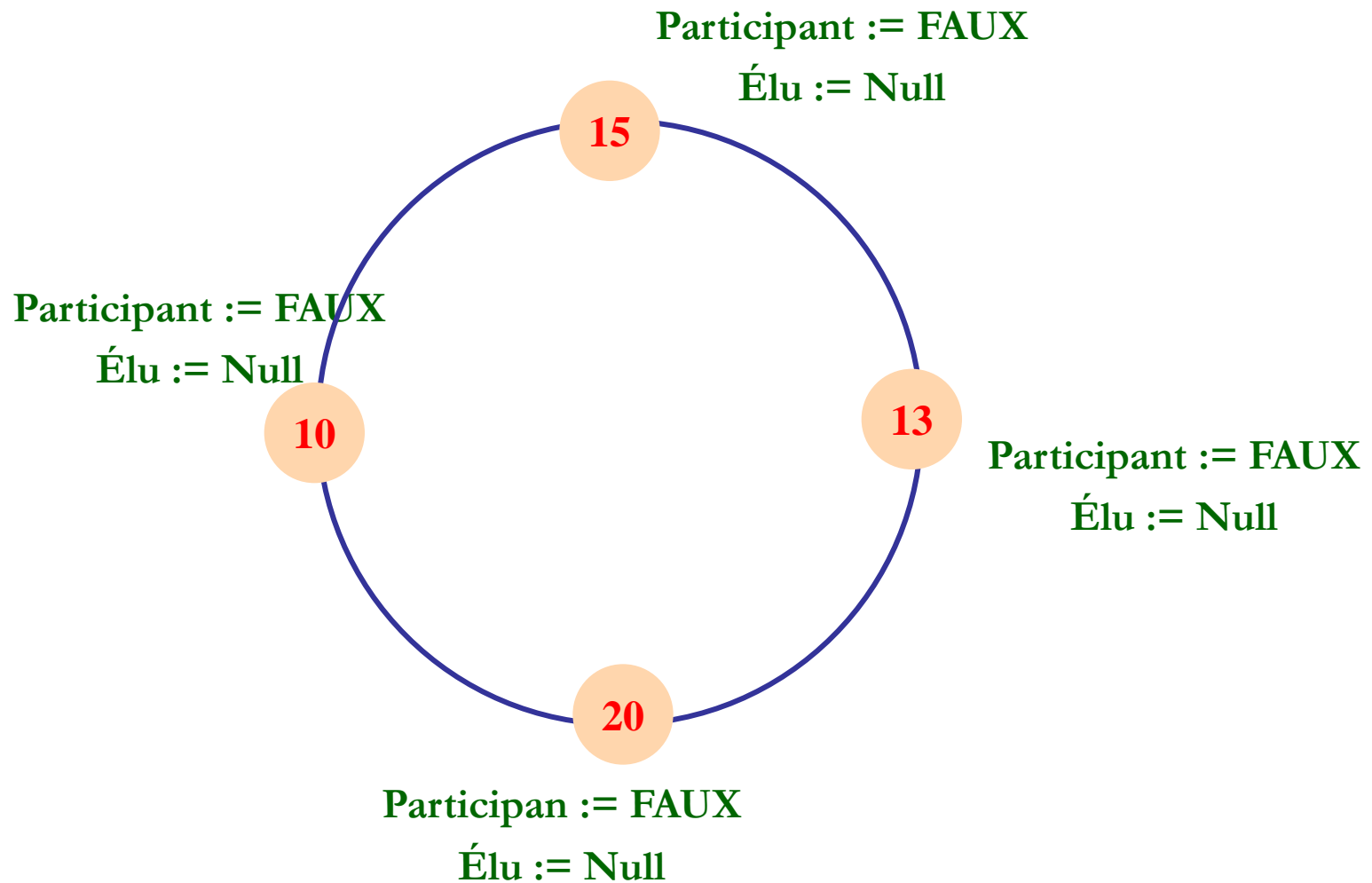
❑ $\text{Participanti} := \text{FAUX}$

❑ Si ($P_i \neq P_j$) alors Envoie le message $\langle \text{élu}, P_j \rangle$ à son voisin (proclamer le résultat)

❑ FinSI

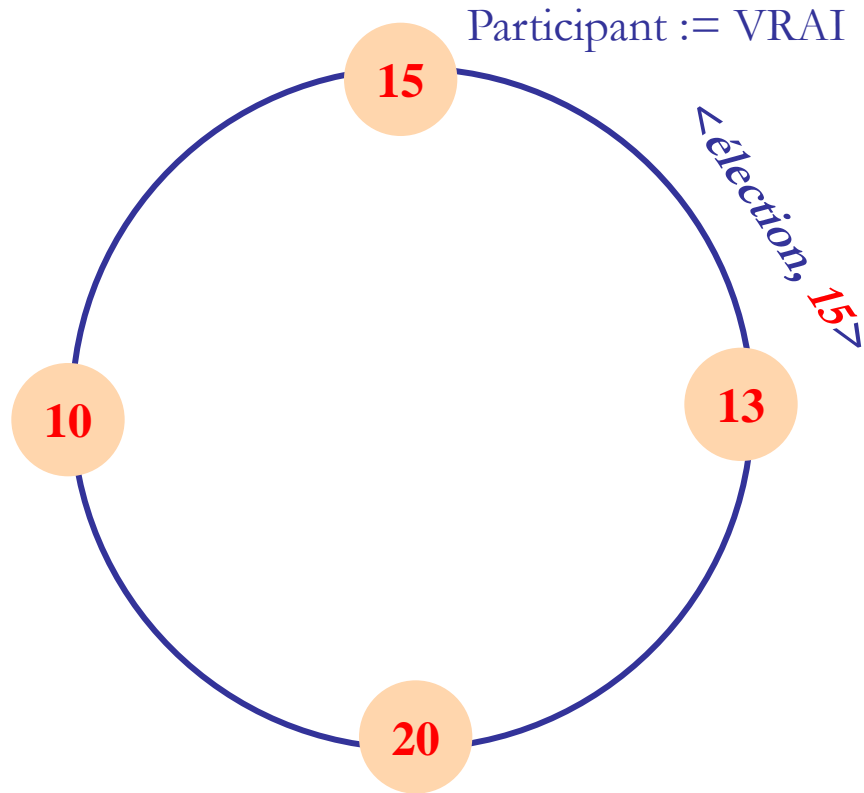
Exemple

□ Initialisation :



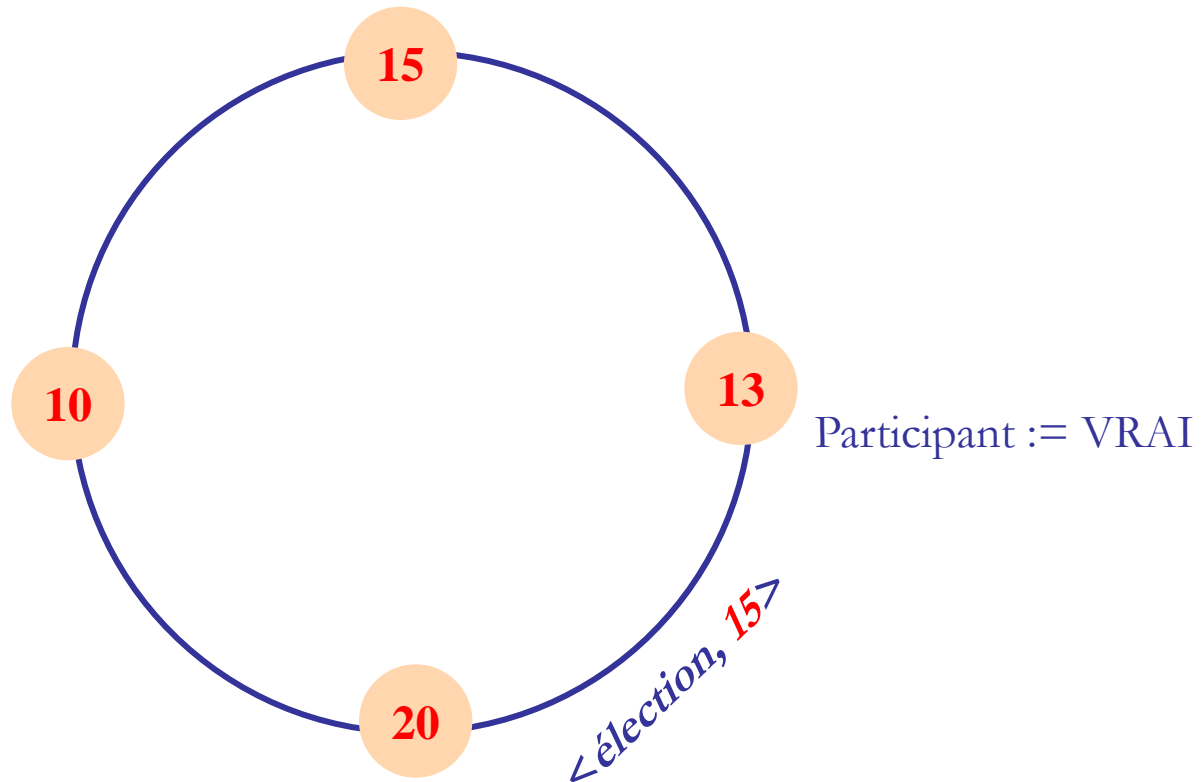
Exemple

- ❑ Processus 15 déclenche l'élection.



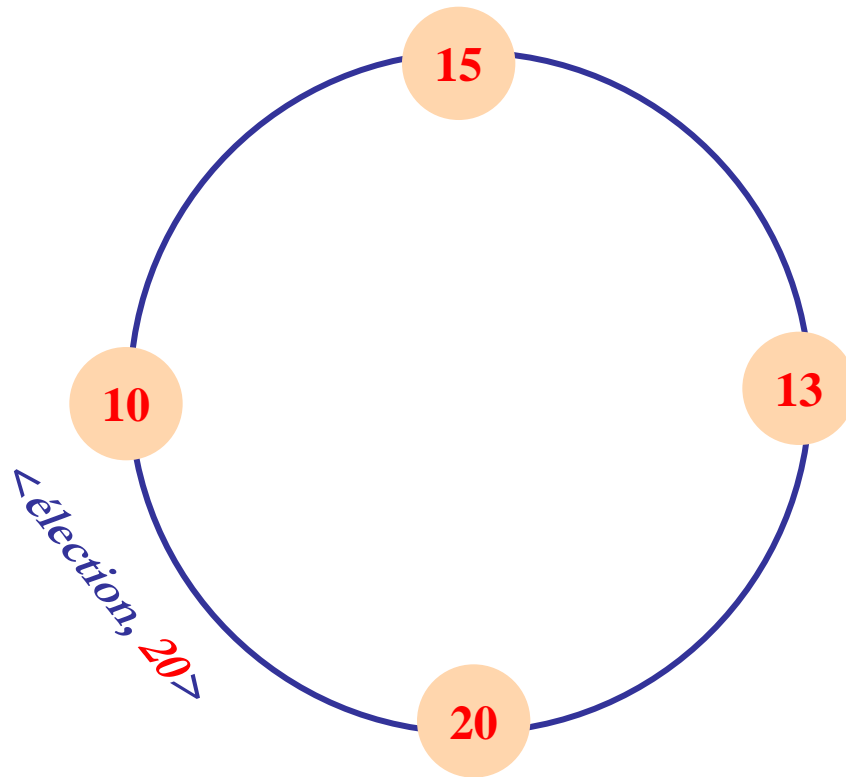
Exemple

- ❑ Processus 15 déclenche l'élection.



Exemple

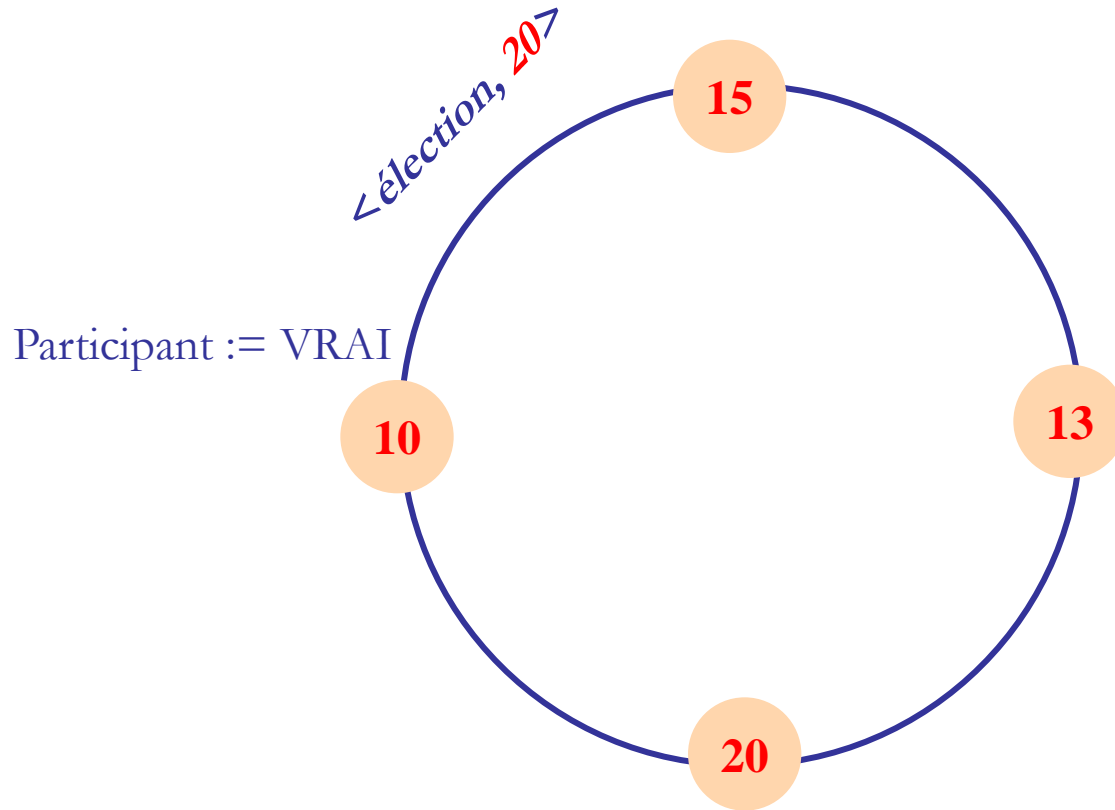
- ❑ Processus 15 déclenche l'élection.



Participant := VRAI

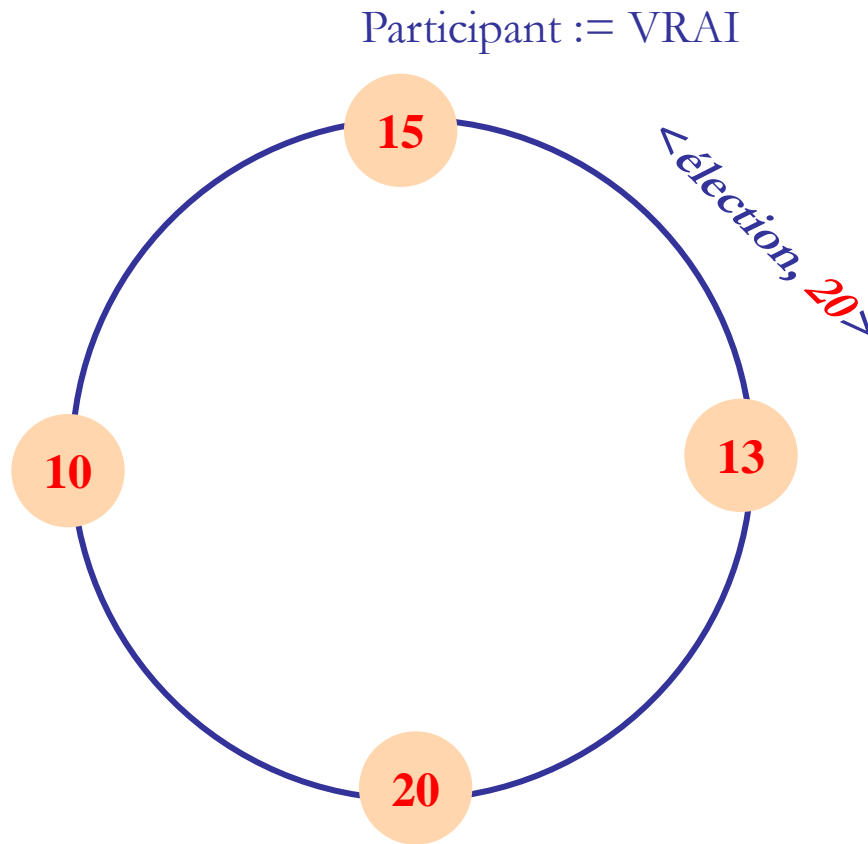
Exemple

- ❑ Processus 15 déclenche l'élection.



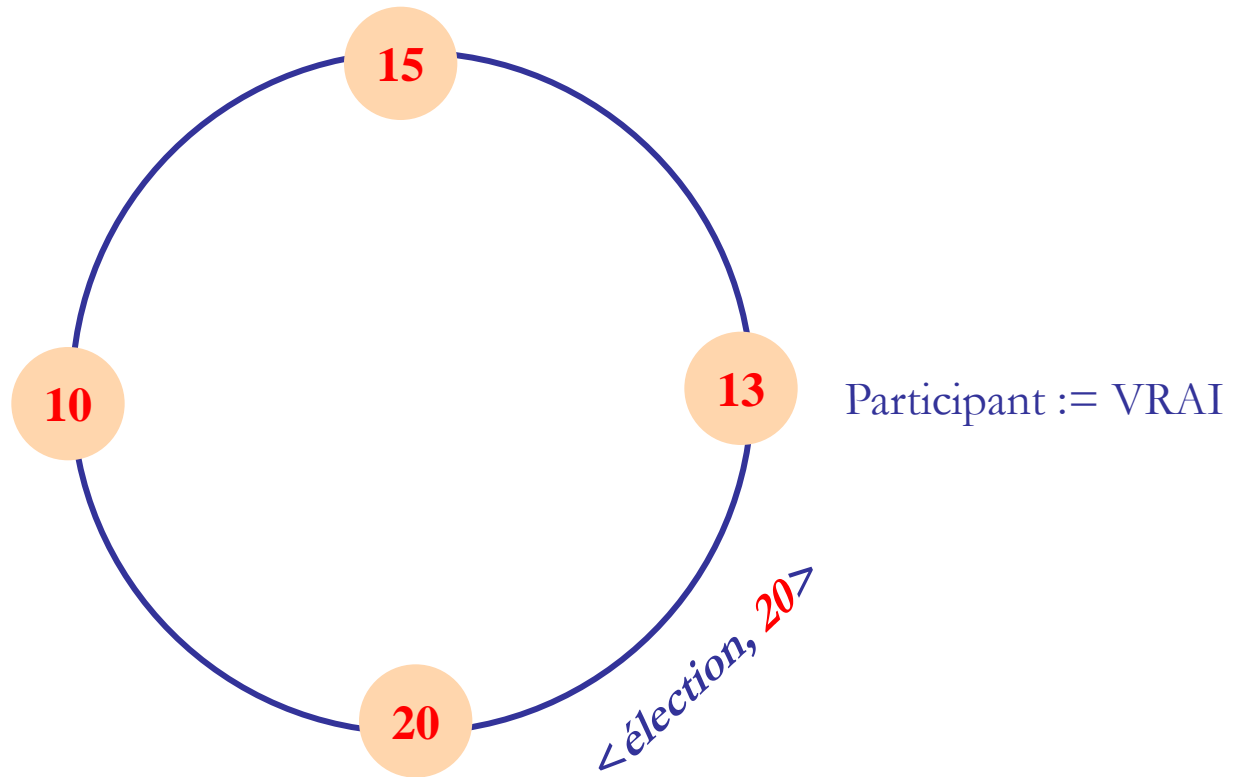
Exemple

- ❑ Processus 15 déclenche l'élection.



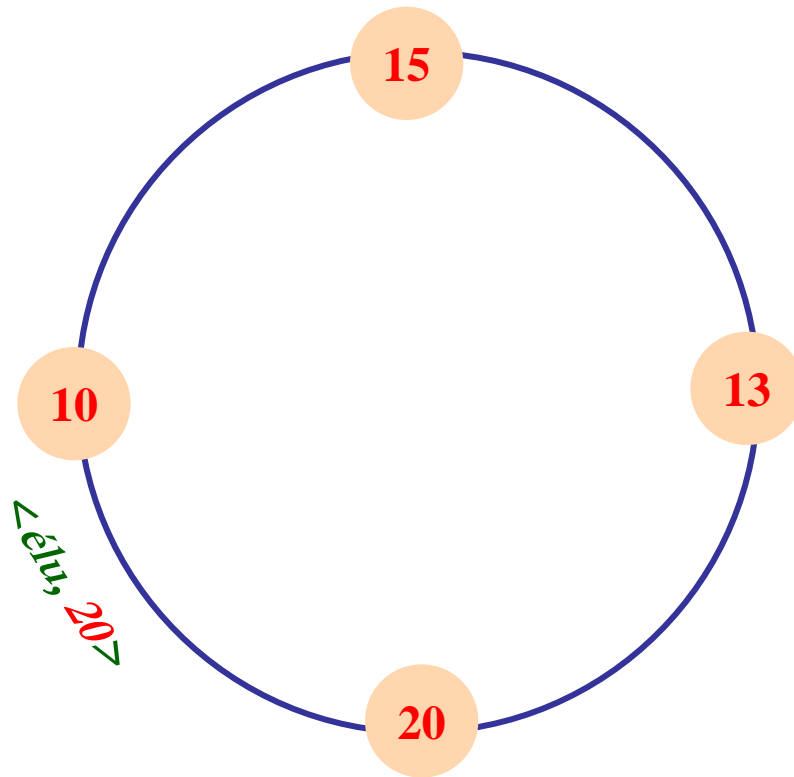
Exemple

- ❑ Processus 15 déclenche l'élection.



Exemple

- ❑ Processus 15 déclenche l'élection.

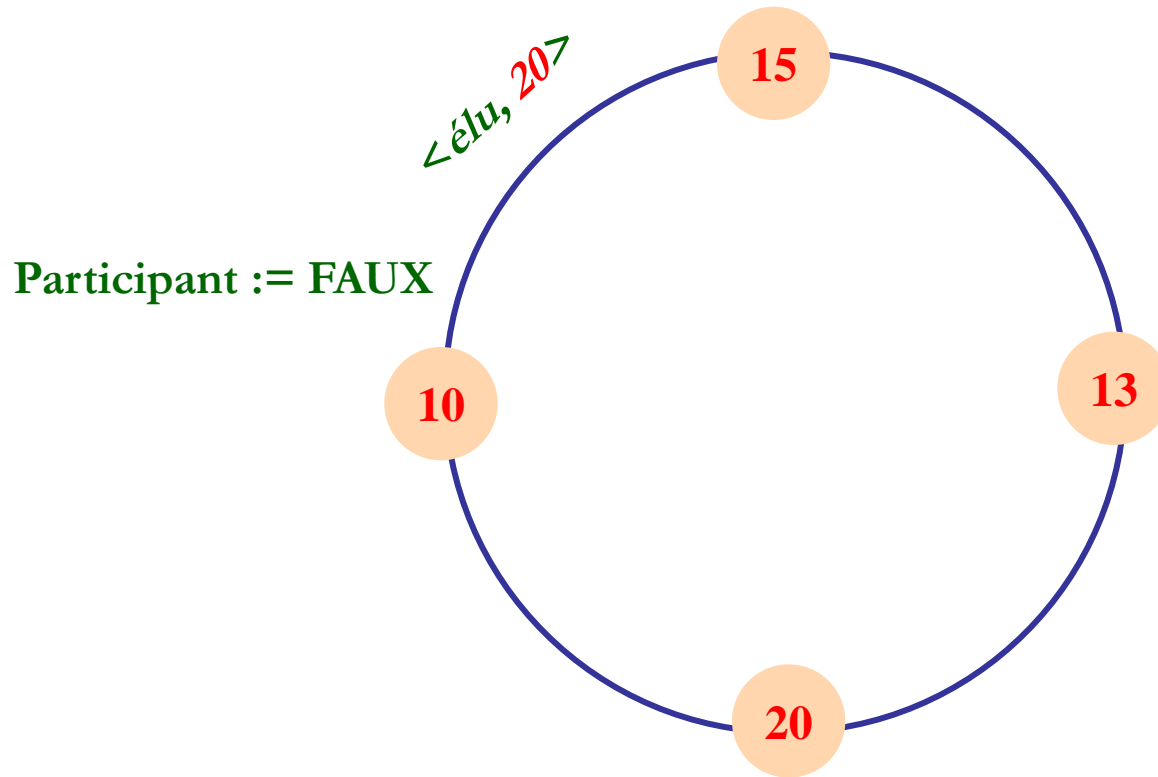


Élu := VRAI

Participant := FAUX

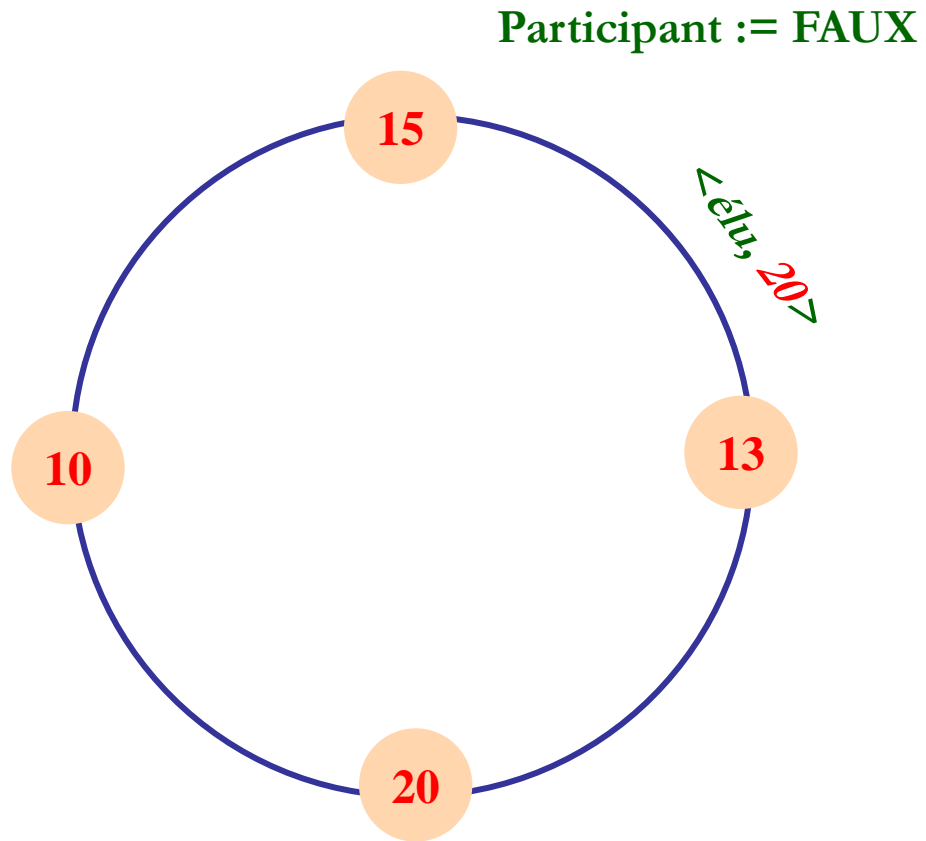
Exemple

- ❑ Processus 15 déclenche l'élection.



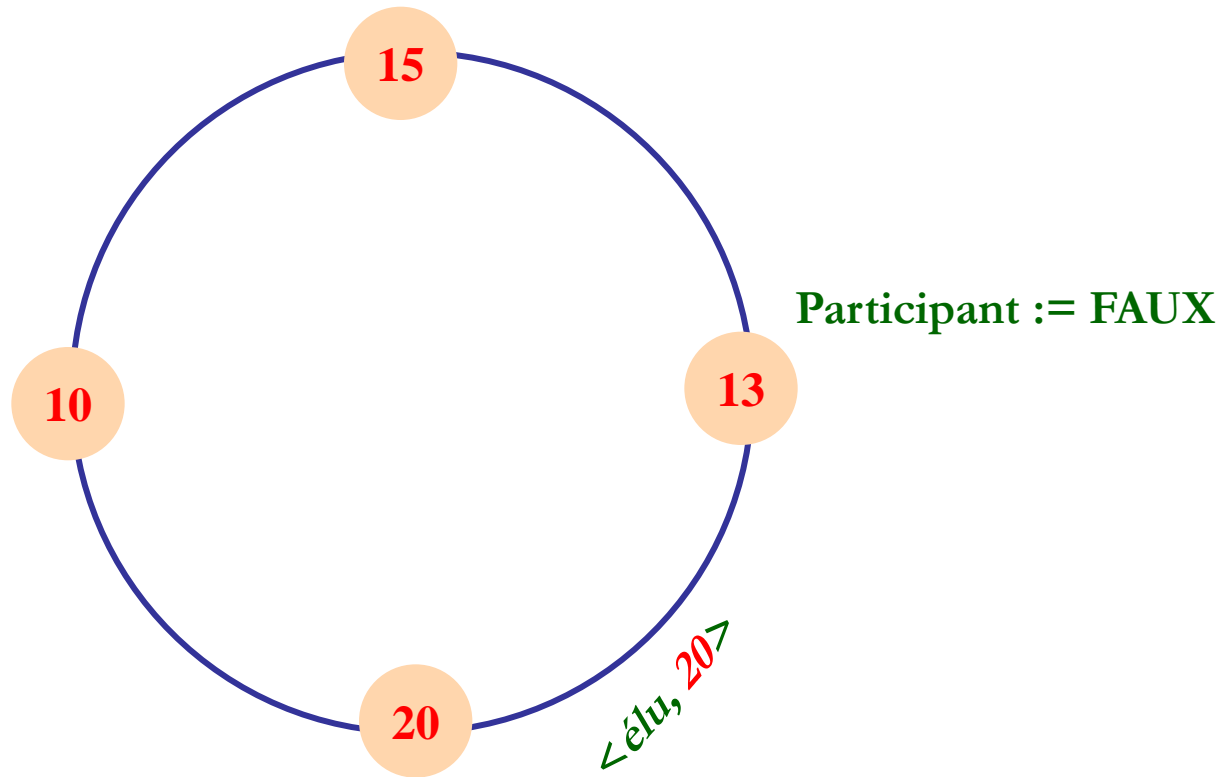
Exemple

- ❑ Processus 15 déclenche l'élection.



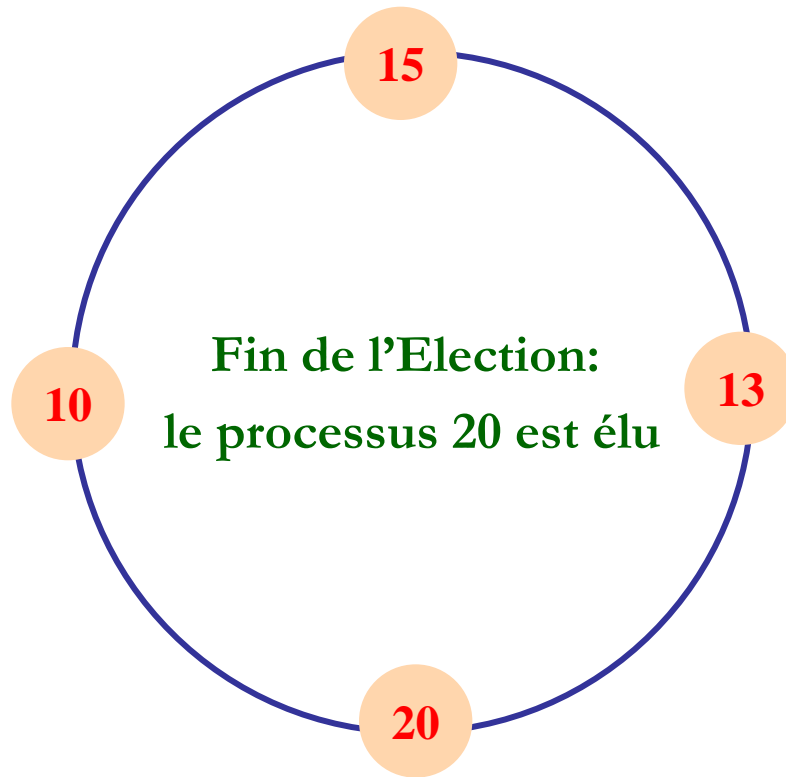
Exemple

- ❑ Processus 15 déclenche l'élection.



Exemple

- ❑ Processus 15 déclenche l'élection.



Algorithme de Chang et Roberts (1979)

Réception d'un message d'élection $\langle \text{élection}, P_i \rangle$ à P_j :

Si $P_i > P_j$ Alors

- Envoie le message $\langle \text{élection}, P_i \rangle$ à son voisin
- Participantj := VRAI

Sinon (//deux cas)

Si $(P_i < P_j \text{ ET } \text{Participantj} = \text{FAUX})$ Alors

- Alors Envoie le message $\langle \text{élection}, P_j \rangle$ à son voisin
- Participantj := VRAI

Sinon $(P_i = P_j)$

- Éluj := VRAI
- Participantj := FAUX
- Envoie le message $\langle \text{élu}, P_j \rangle$ à son voisin (initié la proclamation des résultats)

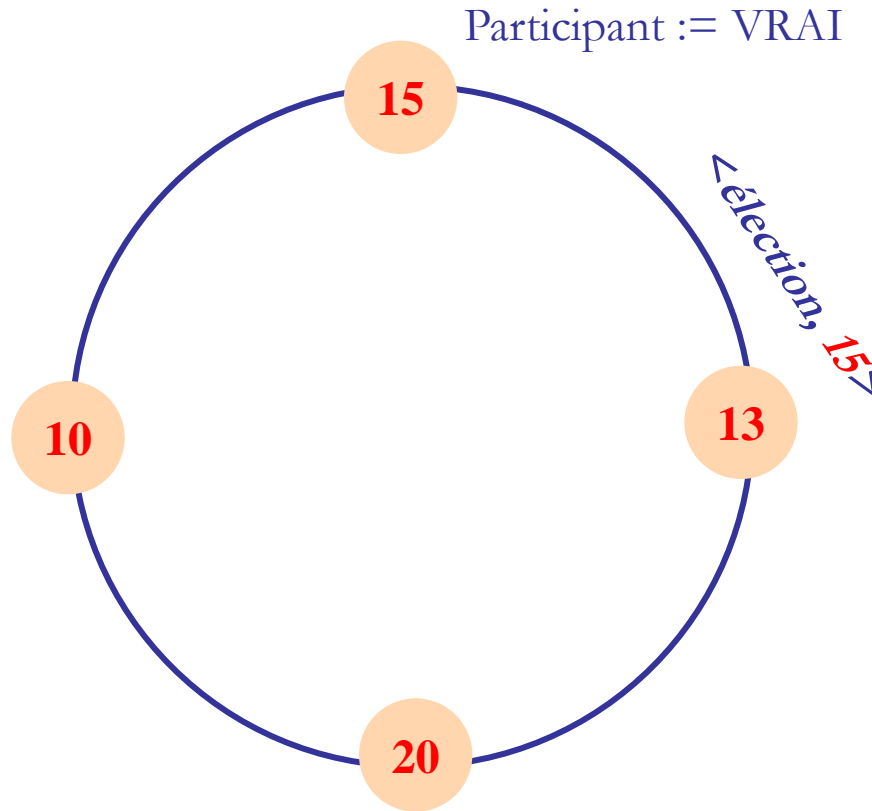
FinSi

FinSi



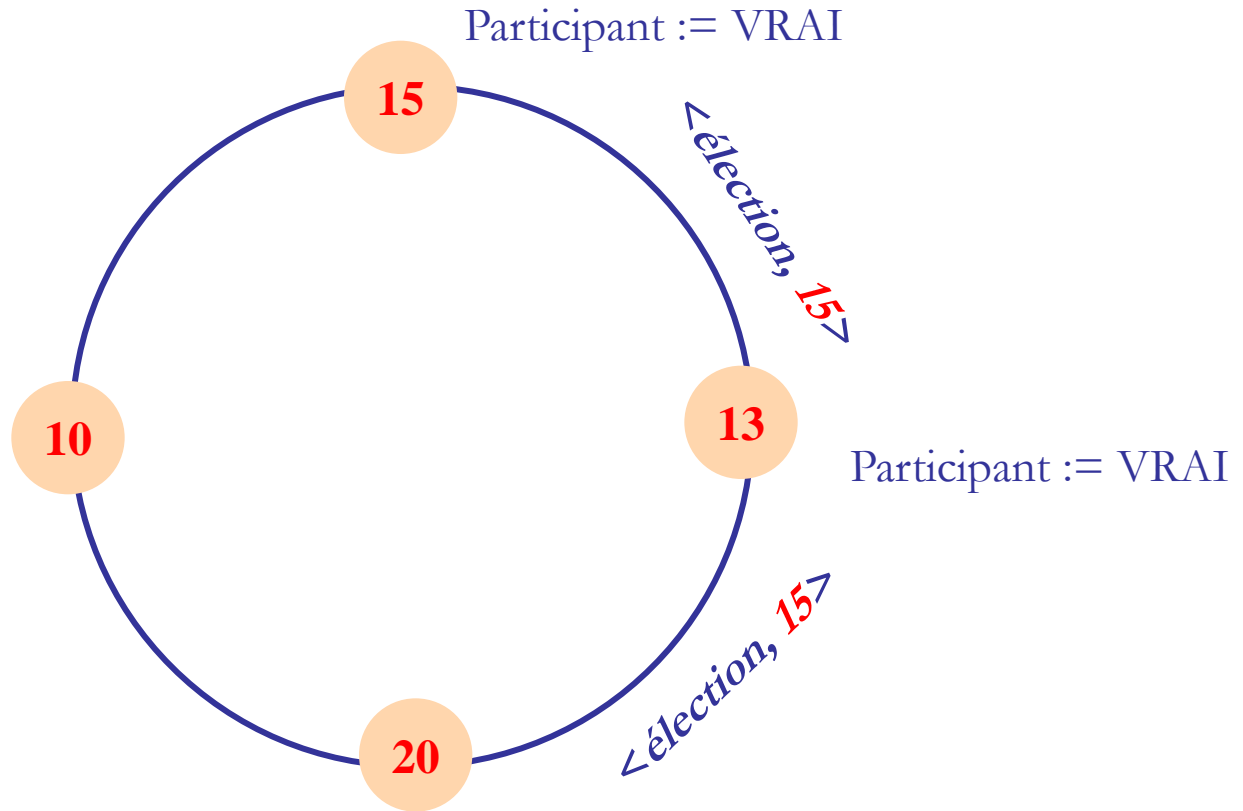
Exemple

- ❑ Processus 15 déclenche l'élection.



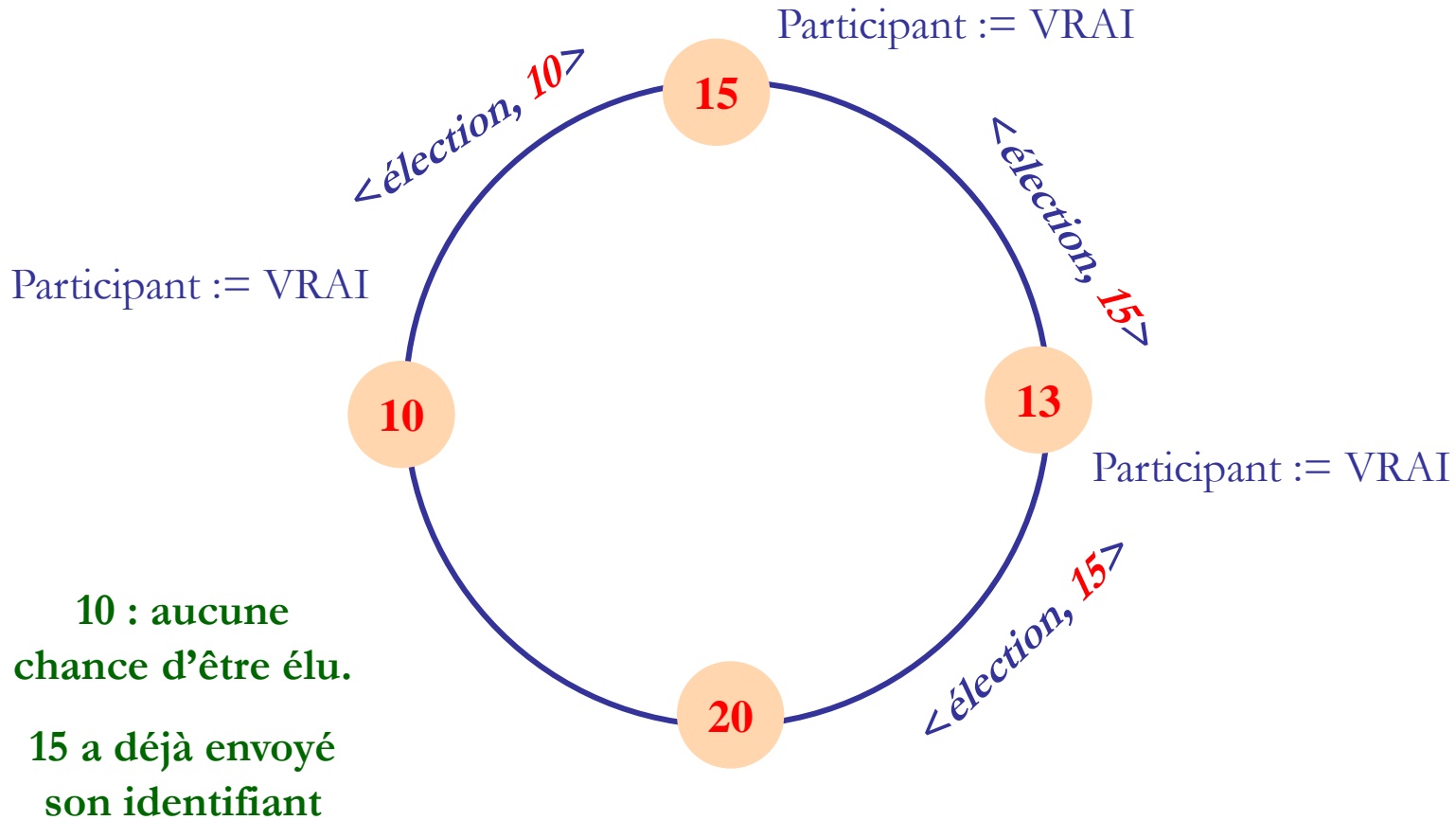
Exemple

- ❑ Processus 15 déclenche l'élection.



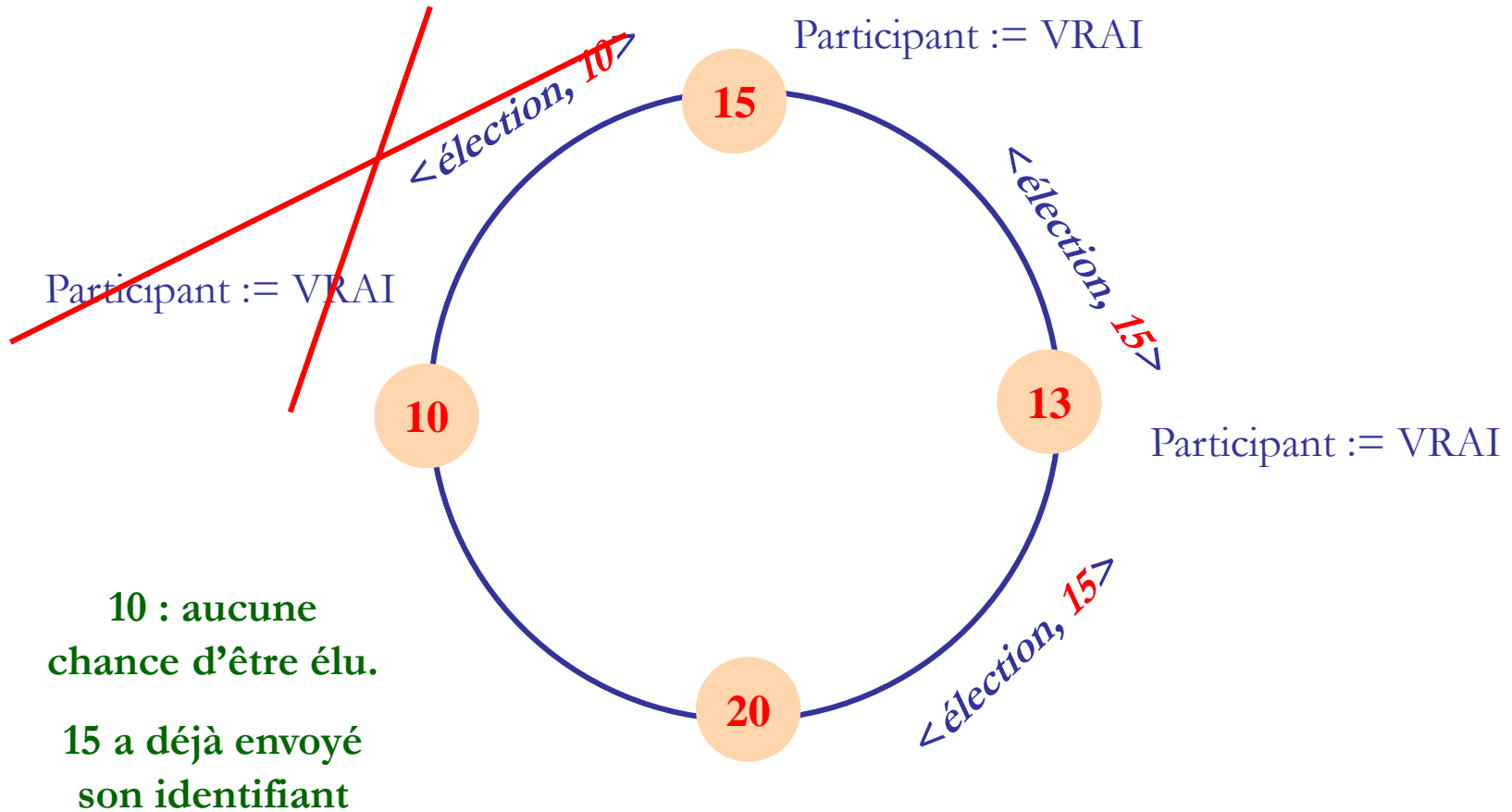
Exemple

- ❑ Processus 10 déclenche une nouvelle élection.



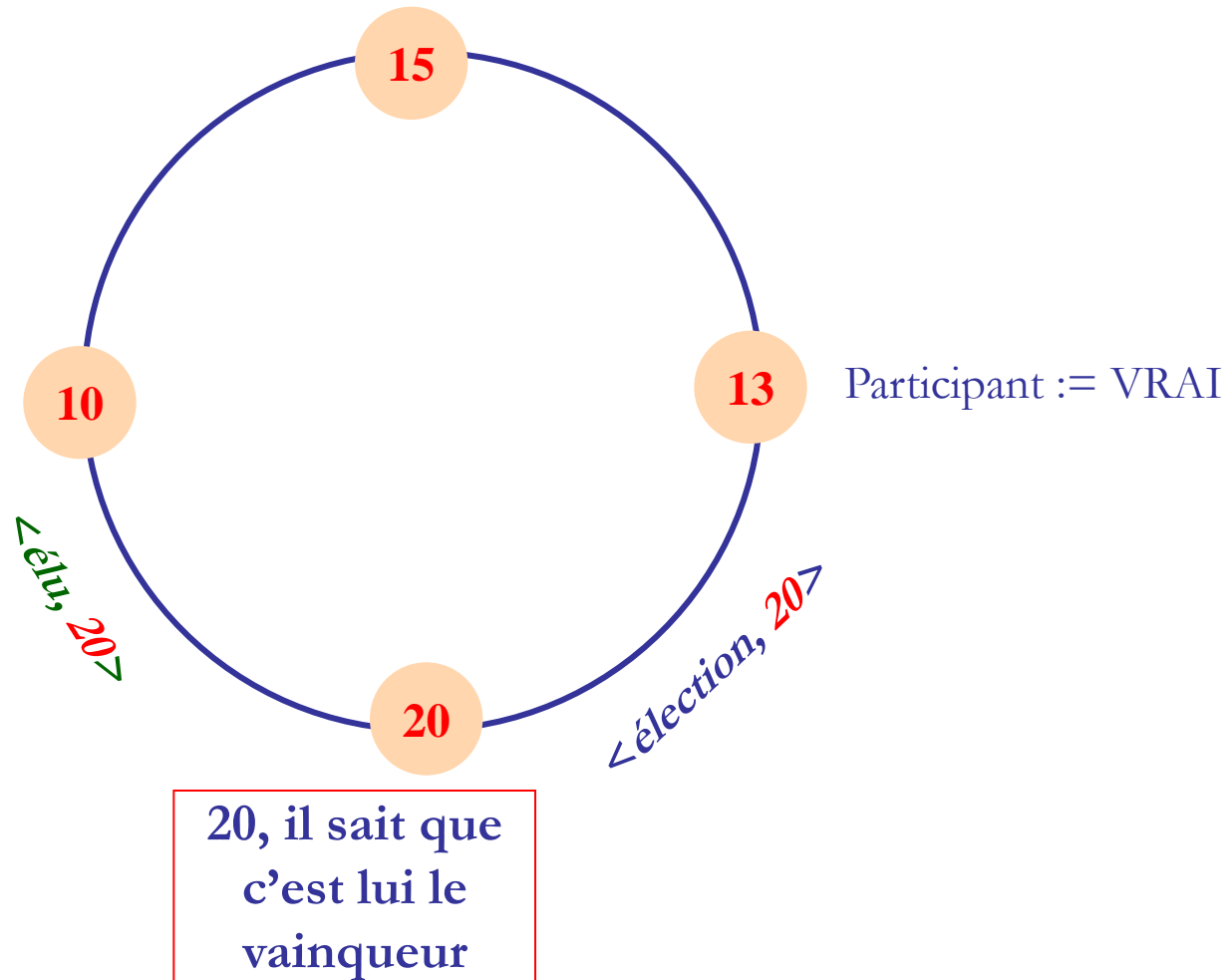
Exemple

- ❑ Processus 10 déclenche une nouvelle élection.



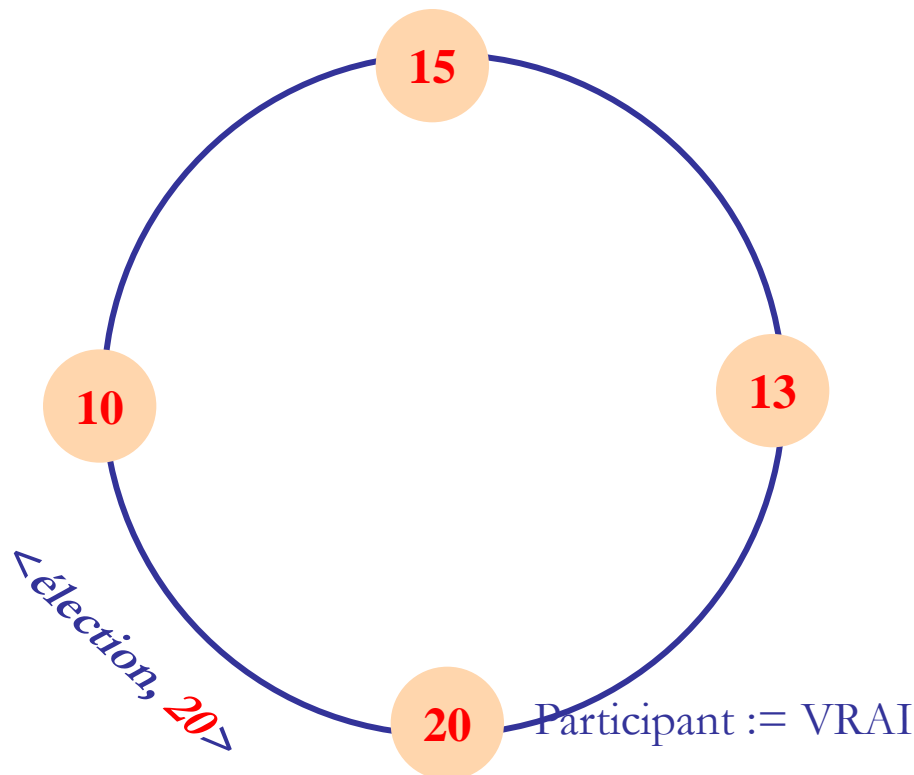
Evaluation

- Du point de vue de la performance en temps, il faut toujours le temps d'un tour d'anneau entre le moment où le vainqueur sait qu'il est vainqueur et la fin du protocole.



Evaluation

- ❑ On peut cependant distinguer deux cas extrêmes :
- ❑ **Le futur vainqueur lance en premier** (éventuellement en même temps que d'autres processus) **une élection** ; il faut alors le temps de la transmission de n messages (le tour de l'anneau) pour qu'il sache qu'il est vainqueur.
 - ❑ **$2n$ messages pour atteindre la fin de l'élection.**



Evaluation

- ❑ On peut cependant distinguer deux cas extrêmes :
- ❑ **Le futur vainqueur est à droite d'un unique processus déclenchant l'élection** ; il faudra alors le temps de $n - 1$ messages pour que le futur vainqueur devienne participant puis le temps de n messages pour que celui-ci découvre qu'il est vainqueur, puis le temps de n messages pour atteindre la fin du protocole d'élection (envoyer le message élu pour proclamer le résultat) soit le temps de la transmission de $3n - 1$ messages.
- ❑ **Dans tous les cas, le temps requis est en $O(n)$.**

