Université de Versailles – Saint-Quentin-en-Yvelines
Algorithmique Distribuée

TD 5 - Election

Exercice 1. *– Algorithme de Lelann et son amélioration par Chang et Roberts –*

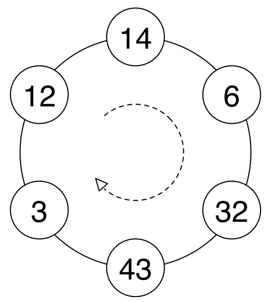
1. Calculer la complexité en messages de l'algorithme de Lelann.
2. Écrire l'algorithme de Chang et Roberts. Cet algorithme est une amélioration de l'algorithme de Lelann. Ici, tous les messages ne font pas le tour de l'anneau. En effet, quand un site reçoit un message, il ne le transmet que si l'identifiant du message est plus petit que l'identifiant du leader calculé jusqu'à maintenant. Parmi tous les messages reçus, un site ne conserve donc que l'identifiant minimum.
3. En supposant que tous les sites sont initiateurs, quelle est la complexité en messages de l'algorithme de Chang et Roberts dans le meilleur et dans le pire cas ? Vous donnerez la position relative des identifiants qui permet d'obtenir la complexité au mieux et celle au pire (vous pouvez supposer que les identifiants des n sites sont $1, 2, \ldots, n$), puis vous calculerez cette complexité.
4. Quelle est la complexité en temps de cet algorithme (en synchrone) ?

Exercice 2. *– Analyse de l'algorithme de Peterson, Dolev, Rodeh et Klawe vu en cours –*

1. On considère une version simplifiée de cet algorithme, dans laquelle un site commence par recevoir les identifiants de ses deux prédécesseurs, puis décide de rester actif ssi son propre identifiant est plus petit que les 2 identifiants reçus. Expliquer pourquoi cet algorithme ne fonctionne pas à travers un exemple.

2. On considère l'anneau unidirectionnel ci-dessous. Compléter le tableau représentant les variables au début du premier tour d'exécution de l'algorithme de Peterson, Dolev, Rodeh et Klawe (donné en annexe).

p	ci_p	acn_p	q	$state_p$
14				active
6				active
32				active
43				active
3				active
12				active



3. Poursuivre l'exécution de l'algorithme en remplissant pour chaque tour de l'algorithme un tableau du même type que le précédent.

4. Calculer la complexité en messages de cet algorithme.

5. Donner une configuration initiale de l'algorithme tel qu'il s'exécute en un maximum de tours (raisonner dans le cadre général).

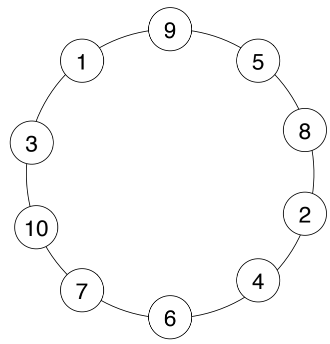
6. Donner une configuration initiale pour laquelle l'algorithme s'exécute en seulement 2 tours sans se soucier du nombre de sites initiateurs de l'algorithme.

Exercice 3. – *Algorithme DoubleToken* –

Hypothèses : Le système est asynchrone. Chaque site est identifié avec un identifiant unique et chaque site est initiateur. La topologie est un anneau bidirectionnel non orienté. Il n'y a pas de fautes.

Principe : L'algorithme fonctionne par phase. A la phase ℓ, un noeud reste candidat (c'est-à-dire qu'il continuera à envoyer des messages avec son identifiant à la phase suivante) si tous ses voisins de droite à distance inférieure ou égale à 2^ℓ et tous ses voisins de gauche à distance inférieure ou égale à 2^ℓ portent un identifiant plus petit que le sien. En répétant les phases, l'algorithme élit ainsi le noeud d'identifiant maximum dans l'anneau.

Plus précisément, chaque site opère en phase 0, 1, 2, ... Lors de chaque phase ℓ, le site i envoie deux tokens contenant son identifiant dans deux directions. Ils sont supposés voyager à distance 2^ℓ, puis retourner à leur origine i. Si les deux tokens lui reviennent, le site i reste candidat à l'élection et passe donc à la phase suivante. Cependant, il se peut que les tokens ne reviennent pas. Si c'est le cas, le site devient passif et ne crée alors plus aucun message portant son identifiant. Tant qu'un token est dans sa phase d'éloignement, chaque site j sur son chemin compare à i son identifiant. Si $i < j$, le site j élimine simplement le token et si $i > j$, il le relaie. Si $i = j$, le site se déclare leader. Tous les sites relaient un token rentrant.



1. On considère l'anneau de la figure de droite.
   (a) Qui survit à la phase 0 ? à la phase 1 ? à la phase 2 ?
   (b) Quel site est finalement élu ? A quelle phase le leader sait-il qu'il est leader ?
2. Écrire la version formelle de cet algorithme.
3. Quelle est la complexité en messages de cet algorithme dans le meilleur et dans le pire cas ? Dans le cas au mieux, vous donnerez la position relative des identifiants qui permet d'obtenir cette complexité (vous pouvez supposer que les identifiants des n sites sont $1, 2, \ldots, n$).
4. Cet algorithme est-il correct si seul un sous ensemble des sites est initiateur ? Si oui, qui est élu à la fin, sinon pourquoi l'algorithme n'est pas correct ?

Exercice 4. – *Algorithme Vitesse* –

On suppose un graphe de communication en anneau. Chaque site i initialise un token qui parcourt l'anneau en transportant l'identifiant de l'initiateur. Différents tokens voyagent à des vitesses différentes. En particulier, un token transportant l'identifiant v voyage au rythme d'une émission tous les 2^v rounds, c'est à dire qu'un site recevant ce token attend 2^v rounds avant de le retransmettre.

Chaque site garde la valeur du plus petit identifiant qu'il a vu et ne transmet pas les tokens qui transportent un identifiant plus grand. Si un token revient à son initiateur, celui-ci se note leader.

On suppose ici que tous les sites sont activés initialement.

1. Qui est élu ?
2. Quelles sont les hypothèses faites sur le système distribué dans cet algorithme ?
3. Écrire la version formelle de cet algorithme.
4. En considérant que les identifiants sont dans l'ensemble $\{1, \ldots, n\}$, quelle est la complexité en messages et en temps (synchrone) de cet algorithme dans le pire cas ?