Circulation de jeton dans un réseau quelconque

TD 8 et 9 - Algorithmique distribuée - Polytech Grenoble INFO4

L'algorithme proposé ici fait circuler un jeton dans un graphe à l'aide d'un parcours en profondeur d'abord. Il suppose des processus et canaux asynchrones dans une topologie quelconque (connexe) d'au moins deux nœuds. Il devra n'y avoir qu'un seul initiateur, de plus, les canaux sont étiquetés de 1 à δ_p pour tout processus p .

Algorithme 1 *Circulation* pour tout processus p

Variables

- 1: **Initiateur** : booléen vrai pour un unique processus (l'initiateur)
- 2: *Visite* : table de booléens indexé sur les canaux sortants, servant à marquer le passage du jeton
- 3: *Père* : père du processus dans l'arbre construit par le parcours du jeton

Algorithme pour tout processus p

```
4: Pour tout voisin  $q$  faire
5:    $Visite[q] \leftarrow faux$ 
6: Fin Pour
7: Si Initiateur alors
8:    $Visite[1] \leftarrow vrai$ 
9:    $Père \leftarrow \top$ 
10:  Envoyer  $\langle Jeton \rangle$  à 1
11: Sinon
12:    $Père \leftarrow \perp$ 
13: Fin Si

14: Pour toujours
15:  Réceptionner  $\langle Jeton \rangle$  de  $q$ 
16:  Si  $Père = \perp$  alors
17:     $Père \leftarrow q$ 
18:  Fin Si
19:  Si pour tout voisin  $q'$ ,  $Visite[q'] = vrai$  alors
20:    décide
21:  Sinon Si  $Père \neq q \wedge \neg Visite[q]$  alors
22:     $Visite[q] \leftarrow vrai$ 
23:    Envoyer  $\langle Jeton \rangle$  à  $q$ 
24:  Sinon Si il existe un voisin  $q' \neq q$  tel que  $Père \neq q' \wedge \neg Visite[q']$  alors
25:     $Visite[q'] \leftarrow vrai$ 
26:    Envoyer  $\langle Jeton \rangle$  à  $q'$ 
27:  Sinon
28:     $Visite[Père] \leftarrow vrai$ 
29:    Envoyer  $\langle Jeton \rangle$  à  $Père$ 
30:  Fin Si
31: Fin Pour toujours
```

Question 1 – Implémenter cet algorithme dans un projet SINALGO. On pourra s'aider du squelette de code fourni ici : <http://www-verimag.imag.fr/~altisen/APD/>. Le tester sur différentes topologies de réseaux de façon à bien comprendre le parcours du jeton. Notamment, mettre en place des fonctions qui observent l'état global du système de façon à

1. valider que le jeton reste toujours unique,
2. tracer le parcours du jeton dans le graphe et
3. compter le nombre de messages échangés.

Question 2 – Redonner la spécification que doit valider l’algorithme.

Question 3 – Pour prouver l’algorithme, prouver successivement les assertions suivantes :

1. Il existe au plus un jeton dans le réseau.
2. Il y a au plus une décision.
3. Chaque canal est traversé un nombre fini de fois dans chaque sens. (Préciser combien de fois au plus.)
4. L’exécution termine.
5. L’initiateur finit par décider.
6. Si une décision est prise, alors tous les processus ont été visités par le jeton.

Puis conclure.

Question 4 – On note m le nombre de liens dans le réseau. Etudier le nombre de messages envoyés par l’algorithme et sa complexité en temps.