

Algorithmique distribuée – Examen

INFO4, PolyTech’Grenoble

Tous documents autorisés
Temps imparti : 2 heures

Barème indicatif sur 20 points

Hypothèses : Pour tout le sujet, nous considérons un réseau de topologie *bidirectionnelle*, *connexe* et *enracinée*. Cela signifie qu’il existe dans le réseau un unique nœud particulier appelé *initiateur*. Les processus et les canaux sont asynchrones, sans faute et nous ne faisons aucune hypothèse sur l’ordre de traitement des messages dans les canaux.

1 Construction d’arbre

On considère l’algorithme suivant.

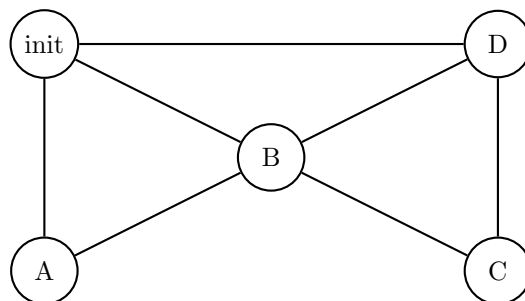
Algorithme 1 Algorithme pour un processus p

Variables

- 1: $dist_p$: un entier naturel ou $+\infty$
- 2: $parent_p$: un pointeur valant soit $\perp$ soit l’un des canaux issus de p

Algorithme pour un processus p

- 3: $parent_p \leftarrow \perp$
 - 4: **Si** p est initiateur **alors** $dist_p \leftarrow 0$ **Sinon** $dist_p \leftarrow +\infty$ **Fin Si**
 - 5: **Si** p est initiateur **alors** Envoyer $\langle A, dist_p \rangle$ à tous les voisins p **Fin Si**
 - 6: **Pour toujours**
 - 7: Réceptionner $\langle A, d \rangle$ de q
 - 8: **Si** $d + 1 < dist_p$ **alors**
 - 9: $dist_p \leftarrow d + 1$
 - 10: $parent_p \leftarrow q$
 - 11: Envoyer $\langle A, dist_p \rangle$ à tous les voisins différents de q
 - 12: **Fin Si**
 - 13: **Fin Pour toujours**
-



Question 1 (2 points) Faire un exemple d'exécution de l'algorithme 1 sur la topologie précédente.

- (a) L'algorithme termine-t-il ? Si oui, donner la configuration que vous obtenez.

Observation sur l'exemple : si l'algorithme termine,

- (b) quelle est la valeur $dist_p$ pour un nœud p donné ?
(c) que représentent les pointeurs $parent_p$ pour l'ensemble des nœuds p ?
(d) que pouvez vous dire d'un chemin entre un nœud p et l'initiateur et de sa longueur ?

On s'attache maintenant à démontrer les constatations que vous avez pu faire à la question précédente.

Question 2 (1 point) Pour un processus p donné, les valeurs successives de sa variable $dist_p$ sont décroissantes. Pourquoi ?

Question 3 (3 points) On veut montrer que : (P_1) Pour un processus p donné, si la valeur de sa variable $dist_p$ est différente de $+\infty$ alors il existe un chemin de l'initiateur à p de longueur $dist_p$.

Cette démonstration se fait par récurrence sur l'exécution.

- (a) Au tout début de l'exécution, seul l'initiateur s'est exécuté. Pourquoi la propriété (P_1) est-elle vraie à ce moment là ?
(b) On suppose que l'exécution a maintenu la propriété (P_1) jusqu'à un certain moment et on considère le pas de calcul suivant où plusieurs processus peuvent s'exécuter en même temps. Pourquoi (P_1) est elle encore vraie après ce pas de calcul ?
(c) Conclusion : pourquoi peut on dire qu'à tout moment de l'exécution et pour processus p , la valeur de $dist_p$ est supérieure ou égale à la distance entre l'initiateur et p ?

Question 4 (2 points) Soit x un entier naturel. On veut montrer que : (P_2) Pour tout nœud p à distance x de la racine, $dist_p$ atteint en temps fini la valeur x et $parent_p$ pointe à ce moment là vers un nœud à distance $x - 1$ (ou $\perp$ si x vaut 0).

Montrer cette propriété par récurrence sur x , pour toute valeur de x .

Question 5 (2 points) Dédurre des propriétés ci-dessus que

- (a) l'algorithme termine (c'est-à-dire que pour tout processus p , ses variables $dist_p$ et $parent_p$ ne changent plus de valeur au bout d'un temps fini)
(b) et que, après terminaison, les pointeurs parents forment un arbre couvrant en largeur.

Question 6 (4 points) Complexité. (Vous pourrez utiliser la topologie précédente pour faire un exemple.)

- (a) Quelle est la complexité en temps de l'algorithme ?
(b) On cherche le nombre de messages échangés au maximum avant terminaison.
(i) Pour un nœud p , la valeur $dist_p$, si différente de $+\infty$, vaut au plus $n - 1$, où n est le nombre de nœuds du réseau. Justifier pourquoi.
(ii) En déduire que les lignes 9, 10 et 11 sont exécutées par un nœud p au plus n fois.
(iii) En déduire le nombre de messages échangés au maximum avant terminaison.

2 Routage

On veut se servir de la structure construite par l'algorithme précédent pour faire du routage. Après terminaison de celui-ci, chaque nœud p devra envoyer un message $\langle Route, M_p \rangle$ jusqu'à l'initiateur. Pour des raisons d'efficacité, on souhaite que ce message passe par un plus court chemin entre p et l'initiateur, d'où le fait d'utiliser l'arbre couvrant construit précédemment.

Question 7 (1 point) Écrire un algorithme de routage en supposant l'algorithme précédent terminé, c'est-à-dire que les variables $dist_p$ et $parent_p$ sont correctement calculées et constantes.

Question 8 (2 point) On souhaite maintenant écrire un algorithme complet permettant à chaque nœud p d'envoyer son message $\langle Route, M_p \rangle$ à l'initiateur en combinant l'algorithme de la première partie et celui de la Question 7.

- (a) Quelle est la difficulté de la mise en œuvre ?
- (b) Quels outils connaissez vous pour y remédier ?

Question 9 (4 points) On considère l'algorithme 2, où δ_p représente le degré du nœud p .

- (a) Que fait cet algorithme pour les variables $dist_p$ et $parent_p$? Justifier.

L'algorithme déclenche aussi la fonction *terminé()*. Le principe ici est que chaque nœud compte le nombre de fois où il a envoyé sa distance à des voisins et en attend des accusés de réception. Quand il a reçu tous ses accusés de réception, le nœud considère que son calcul de distance est terminé et fait remonter cette information à son père dans l'arbre. Seul l'initiateur ayant reçu tous les accusés de réception peut décider de la terminaison de l'algorithme observé. Sur la base de ce principe,

- (b) Expliquer les lignes 5, 9, 19 et 24
- (c) À quoi sert la ligne 15 ?
- (d) Expliquer le principe mis en œuvre dans les lignes 25 à 31

Question 10 (1 point) Sans en faire plus la preuve, on admet que l'algorithme 2 déclenche la fonction *terminé()* quand l'ensemble des processus a terminé le calcul de l'arbre couvrant en largeur. Comment alors utiliser cet algorithme pour répondre à la Question 8.

Algorithme 2 Algorithme pour un processus p

Variables

- 1: $dist_p$: un entier naturel
- 2: $parent_p$: un pointeur valant soit $\perp$ soit l'un des canaux issus de p
- 3: cpt_p : un entier naturel (nombre d'accusés de réception attendus)

Algorithme un processus p

```
4:  $parent_p \leftarrow \perp$ 
5:  $cpt_p \leftarrow 0$ 
6: Si  $p$  est initiateur alors  $dist_p \leftarrow 0$  sinon  $dist_p \leftarrow +\infty$  Fin Si
7: Si  $p$  est initiateur alors
8:   Envoyer  $\langle A, dist_p \rangle$  à tous les voisins  $p$ 
9:    $cpt_p \leftarrow \delta_p$ 
10: Fin Si
11: Pour toujours
12:   Réceptionner  $\langle M \rangle$  de  $q$ 
13:   Si  $M = \langle A, d \rangle$  alors
14:     Si  $d + 1 < dist_p$  alors
15:       Si  $parent_p \neq \perp$  alors Envoyer  $\langle ACK \rangle$  à  $parent_p$  Fin Si
16:        $dist_p \leftarrow d + 1$ 
17:        $parent_p \leftarrow q$ 
18:       Envoyer  $\langle A, dist_p \rangle$  à tous les voisins différents de  $q$ 
19:        $cpt_p \leftarrow cpt_p + \delta_p - 1$ 
20:     Sinon
21:       Envoyer  $\langle Ack \rangle$  à  $q$ 
22:     Fin Si
23:   SinonSi  $M = \langle Ack \rangle$  alors
24:      $cpt_p \leftarrow cpt_p - 1$ 
25:     Si  $cpt_p = 0$  alors
26:       Si  $p$  est l'initiateur alors
27:          $terminé()$ 
28:       Sinon
29:         Envoyer  $\langle Ack \rangle$  à  $parent_p$ 
30:       Fin Si
31:     Fin Si
32:   Fin Si
33: Fin Pour toujours
```
