Université Paris 7-Denis Diderot
M2 informatique:
Algorithmes distribués
2011-2012

Devoir 1:
à rendre pour le 5/3/2011
Chaque devoir peut avoir jusqu'à 3 auteurs

Il s'agit d'étudier un algorithme de vague: *l'algorithme de phases*.

Hypothèses Le réseau est représenté par un graphe orienté $G =< \Pi, V >$ fortement connexe. On notera $d(p,q)$ la longueur du plus court chemin de p à q (attention, en général $d(p,q) \neq d(q,p)$!). D le *diamètre* est défini par $D = \max\{d(p,q) | p,q \in \Pi\}$. Pour cet algorithme on suppose que D est connu des processus. Pour chaque processus p, $In_p = \{q \in \Pi | (q,p) \in V\}$ est l'ensemble des processus qui peuvent envoyer directement à p un message. $\preceq$ est la relation de causalité entre événements. Pour simplifier on supposera que la communication est FIFO (les messages entre deux noeuds sont reçus suivant leur ordre d'émission).

Initialisations:
$\forall q \in In_p : Rec_p[q] := 0$
$Phase_p := 0$
CODE POUR LE PROCESSUS p:

$\{<>$ de q $\} \longrightarrow$ recevoir $<>$;
    $Rec_p[q] := Rec_p[q] + 1$

$\{\forall q \in In_p : Rec_p[q] \geq Phase_p \wedge Phase_p < D\} \longrightarrow$ send$<>$ à tous les $q \in Out_p$
    $Phase_p := Phase_p + 1$

$\{\forall q \in In_p : Rec_p[q] \geq D\} \longrightarrow$ decider

---

Figure 1: Algorithme de phases

Questions

1. On considère le graphe $\Pi = \{1,2,3,4,5\}, V = \{(1,2),(2,1),(2,3),(3,2), (3,4),(4,3), (4,5),(5,4)\}$. Ici le diamètre du graphe est 4.

   Donner un exemple d'exécution de l'algorithme de phase (Figure 1). Sur cet exemple indiquer les évènements suivants: l'émission du premier message envoyé par le processus 1 au processus 2, l'émission du 2ème message envoyé par le processus 2 au processus 3, l'émission du 3ème message envoyé par le processus 3 au processus 4, l'émission du 4ème message envoyé par le processus 4 au processus 5. Dans votre exemple d'exécution quel est l'ordre de ces événements?

   Pouvez vous trouver un exemple d'exécution où ces événements apparaîtraient dans un ordre différent?

2. On veut maintenant montrer que l'algorithme de phases termine et décide. Pour cela, on considère une exécution γ de cet algorithme.

   (a) Combien de messages peuvent être envoyés sur chaque lien de communication? Quel est le nombre maximal de messages qui peuvent circuler?

(b) Soit l'instant t_0 à partir duquel dans γ tous les messages ont été reçus et plus aucun message ne sera envoyé.

   i. Montrer que dans γ à partir de l'instant t_0, on a:
      - $Phase_p > 0$ pour tout processus p,
      - $Rec_p[q] = Phase_q$ pour tout processus p, q tel que $q \in In_p$,
      - $Phase_p = D$ ou $\forall q \in In_p : Rec_p[q] < Phase_p$.

   ii. En déduire que tous les processus décident.

3. On va montrer maintenant que l'algorithme de phases est un algorithme de vague. E_p est l'ensemble des événements du processus p. On note e_p l'événement correspondant au démarrage de l'algorithme en p. Si a est un événement sur p, $ph_p(a)$ est la valeur de la variable $Phase_p$ au moment de cet événement.

   (a) Montrer la propriété suivante (on pourra faire une induction sur la distance $d(q,p)$).

$$\text{si } d(q,p) = i \text{ alors } \forall a \in E_p : ph_p(a) > i \Rightarrow e_q \preceq a \tag{1}$$

   (b) En déduire que l'algorithme est un algorithme de vague.

   (c) Que se passe-t-il si on ne suppose plus que la communication est FIFO?

   (d) Quel est le nombre de messages échangés avant la décision de tous les processus?

4. On veut montrer que si $d(q,p) = i$ alors p entendra parler de q pour la première fois en recevant un message d'un voisin r tel que "$r \to p$" est la dernière étape d'un plus court chemin de q à p et au moment de la réception de ce message, $Rec_p[r] = i$.

   Pour cela, soit q tel que $d(q,p) = i$ (on suppose que $i > 0$) et soit $c = (p_0 = q \to \cdots p_{i-1} = r \to p_i = p$ un plus court chemin de q à p, r le dernier sommet de ce chemin sera noté $last(c)$. Soit $a(q) \in E_p$ l'événement tel que pour *première fois* $e_q \preceq a(q)$. ($\forall x \in E_p : x \prec a(q) \Rightarrow \neg(e_q \preceq a(q)$ et $e_q \preceq a(q))$.

   Montrer que :

   (a) $a(q)$ est un événement de réception

   (b) Si $a(q)$ est la réception d'un message provenant de r alors il existe un chemin minimal c de q à p tel que $last(c) = r$ (i) (on pourra fair une induction sur la longueur des chemins minimaux issus de q).

   (c) Montrer que si $a(q)$ est la réception d'un message issu de r alors après cette réception $Rec_p[r] = i$ (ii)

5. On va utiliser la propriété (ii)pour calculer la matrice des distances minimales:

   (a) Donner un algorithme utilisant la propriété (ii) pour construire pour chaque processus p une table TM_p des distances minimales vers p: TM doit assurer que pour tout q, $TM_p[q] = d(q,p)$. A quel moment cette table est-elle complète?

   (b) Proposer une solution pour obtenir sur p la matrice MD des distances: $MD[x,y] = d(x,y)$. Combien de messages sont échangés pour obtenir cette matrice?

6. On va utiliser la propriété (i) pour construire une table de routage minimal. Une table de routage, R_p, indique pour chaque processus vers qui transmettre les messages: $R_p[q]$ est l'identité du voisin de p vers qui p doit transmettre les messages destinés à q. Pour une table de routage minimal cela signifie que $R_p[q]$ est la première étape d'un chemin minimal de p vers q.

   Pour construire R_p pour chaque p, on va d'abord construire T_p, T_p sera construit par p et sera tel que $T_p[q] = R_q[p]$. En clair, $T_p[q]$ est l'identité d'un processus qui est la première étape d'un chemin minimal de q vers p.

(a) Modifier l'algorithme de phases de façon à ce que chaque p calcule T_p.

(b) Comment peut-on procéder pour qu'à partir des T_p, les processus p obtiennent leur table de routage R_p?

(c) Évaluer le nombre de messages échangés pour construire R_p.

7. Essayer de modifier l'algorithme de phases de façon à obtenir un algorithme de vague même si le diamètre et le nombre de processus sont inconnus.