

Modèle de l'algorithmique distribuée

M. Mosbah

I/ Notion d'algorithme distribué

- Un algorithme distribué est un système distribué S est une collection de transitions locales à réaliser séquentiellement par chaque process de S
- Ces transitions peuvent s'effectuer suite à des événements (internes ou échange de message)

- Un algorithme désigne celui d'un seul process et protocole désigne l'ensemble des algos (mais on ne fait cette distinction)
- Événement: interne (changement d'état), envoi de message, réception de message
- Process p : $(\text{etat}_i, \text{evenmt}_i)$ à $(\text{etat}_{i+1}, \text{evenmt}_{i+1})$
- Simulation d'un mémoire partagée: 1 process à part qui est chargé uniquement de la communication
- Terminaison: explicite vs implicite

II/ Modèle du réseau

- Process = sommet ; Canal = arête
1. Types de réseaux
 - A. Réseaux centralisés: Il existe un seul process dans l'état centraliseur (ou élu) et tous les autres dans l'état battu
 - B. Réseaux avec identités: (named network)
Chaque process de S est distingué par l'attribution d'une identité unique, distincte de tous les autres et connue par lui seul
 - C. Réseaux anonymes: les process sont symétriques du point de vue de l'exécution de l'algorithme et ne possèdent pas d'identités (connues d'eux). Ils sont indiscernables.

2. Propriétés du réseau

- Le graphe sous-jacent est connexe
- Les messages sont acheminés le long des canaux. Les délais sont finis mais non bornés (peuvent être arbitrairement longs)
- Les messages reçus par un process sont traités dans leur ordre d'arrivée (si en //, ordre arbitraire)
- Pour tout couple (p_i, p_j) , l'ordre de réception des messages est le même que celui de transmission (FIFO)

• Modes de transmission

- simplex : unidirectionnel
- half duplex : pas de croisement
- full duplex : croisement possible (mode général, graphe non orienté)
- Les états peuvent être codés par des étiquettes
- Absence de pannes (pas de perte de message)
- Topologie du réseau (arbre, anneau, complet,...)

III/ Modèle de communication (synchronisme)

1. Synchrone

- horloge globale
- L'exécution est partitionnée en rounds (un round correspond à un envoi et à une réception de tous les messages. Envoi instantané ou borne fixe)
- Les process ont la même vitesse

2. Asynchrone

- Pas d'horloge globale
- Pas de borne sur le délai de transmissions de messages
- Les process sont asynchrones (les horloges sont locales)

IV/ Mesures de complexité

Modèle synchrone :

- Nombre de messages : depuis le début jusqu'à la fin de l'algo.
- Taille d'un message : complexité en nombre de bits
- Temps : nombre de rounds

- Modèle asynchrone :
 - Nombre de messages
 - Taille d'un message
 - Temps (moins évident): généralement, on suppose que le temps de transmission sur un canal de communication mesure une unité (ou bornée par une constante)
 - Complexité en espace : espace pour un process
 - Calcul de complexité au pire : max sur toutes les exécutions possibles (généralement le pire des exécutions synchrones)
 - Complexité moyenne : valeur moyenne de la complexité (selon une distribution de probabilités)

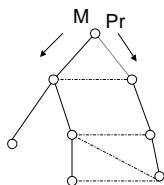
IV/ Quelques algorithmes de base

1. Un algorithme de diffusion (Broadcast, PI)

- Hypo: un arbre recouvrant donné, un message M
- Un sommet P_r distingué (racine)
- P_r envoie M à tous ses fils
- Si un sommet reçoit le message M de son père, il l'envoie à ses fils

```

Algo pour  $P_i \quad i \neq r$ 
  receive (pere( $P_i$ ),M)
  pour tout fils  $P_j$  de  $P_i$ 
    send ( $P_j$ ,M)
  terminer
  
```



- Exercice: Calculer la complexité en nombre de messages ? et la complexité en temps ?

2. Inondation et arbre recouvrant :

un graphe G, un message M à envoyer à tous les sommets

2.1/ Inondation, Diffusion, PI (Propagation d'Information)

- Un sommet distingué (racine) P
- P envoie M à tous ses voisins
- Un process P_i qui reçoit M de P_j pour la première fois, l'envoie à tous ses voisins sauf à P_j

Exercice:

Montrer que la complexité en messages est de $2m-n+1$

2.1/ Application à la construction d'un arbre recouvrant

Quand P_i reçoit M pour la première fois de P_j , il envoie à P_j un message <parent> et <refus> à tous les autres.

Si P_j reçoit <parent> de P_i alors $\text{fils}(P_j) = P_i$