

3.4 L'algorithme ECHO

L'algorithme Echo est un algorithme distribué que l'on peut utiliser pour traverser un graphe connexe quelconque de manière parallèle et pour diffuser des données d'un initiateur à tous les processeurs. L'idée de base est qu'un seul processeur se déclenche et inonde la topologie avec une info et veut recevoir la confirmation que tous ont reçu l'info.

Cet algorithme peut servir à synchroniser des processeurs. Il généralise Chang & Roberts pour n'importe quelle topologie (physique ou logique).

Remarque :

- lorsque l'info fait le tour complet de l'anneau, on est bien dans le cas Chang & Roberts.
- c'est aussi un algo total (fait intervenir tous les processus).

3.4.1 Idée de l'algo Echo :

Le processus initiateur envoie un message à tous ses voisins. Ceux qui reçoivent l'info pour la première fois la transmettent eux-même à leurs voisins (moins le père), mais aussi, mémorisent de quel lien vient l'info (père) pour pouvoir renvoyer un acquittement (un écho).

⇒ Cela correspond à la construction d'un arbre couvrant orienté sur la topologie.

Dès qu'un processus reçoit une info, il devient **engagé**, et propage l'info.

Lorsqu'un processus a reçu un écho ou une info de tous ses voisins, ce processus (autre que initiateur) est **désengagé**, et renvoie écho à son père.

Chaque processus a une variable initiateur initialisée à FAUX.

Un processus ayant reçu un message (info ou écho) de tous ses voisins, sait que tous ses voisins ont reçu l'info, et donc fait remonter écho vers son père.

L'algorithme Echo est terminé lorsque l'initiateur a reçu un message (info ou écho) de tous ses voisins. A ce moment là, il est sûr que tous les processus ont eu l'info.

P_i

I_i : {¬engagé et leader = moi}

// au plus un processus exécute son I_i

initiateur := *Vrai*;

engagé := *Vrai*;

$NI = 0$;

// compte le nb de voisins

envoyer < *info* > aux voisins.

X_i : {un message est arrivé de P }

// message est de type info ou echo

si ¬engagé alors

// mess = info

engagé := *vrai*;

$NI := 0$;

$PRED := P$;

envoyer < *info* > à {voisins - $PRED$ }

$NI := NI + 1$;

si $NI = |\text{voisin}|$ alors

engagé := *Faux*;

si initiateur alors < *terminé* >

sinon, envoyer < *echo* > à $PRED$

Pour cet algorithme on suppose qu'un seul processus devient initiateur. Il suffit d'avoir élu précédemment un leader.

Cet algorithme peut être utilisé pour :

- Construire un arbre couvrant sur une topologie quelconque.
- Faire de la synchronisation entre les processus \leftrightarrow application de type maître-esclave.
- Pour mettre en œuvre la détection de la terminaison.

cad réaliser l'équivalent de la mise en place de l'algo des 4 compteurs = collecter $\sum s$ et $\sum r$ de la part de l'initiateur

3.4.2 Complexité

- Nombre total de message avec un initiateur

Au plus 2 messages transitent par lien. Si chaque processus a au plus E voisins (il y a N processus), le nombre total de message est $2 \times E \times N$.

- Complexité en temps

Le nombre maximum d'enchaînement de messages est au pire, $2 \times N$