

### Exercice 1 : protocole de validation atomique

1. Transparent 33 : diagramme d'états coté processus. Dans l'état "choisi oui", si le délai de garde expire, on suit une transition qui revient sur "choisi oui" et qui lance le protocole de terminaison comme sur le diagramme d'états du transparent 41 pour la 3PC.

Le fait que le protocole à 3 phases termine tient plus du fait que le processus qui ne sait pas quoi faire relance le coordinateur pour terminer l'algorithme que de l'existence d'une 3ème phase. On peut donc utiliser aussi ce protocole de terminaison dans le protocole à 2 phases.

2. Diffusion fiable : un message diffusé finit par être délivré par tous les processus corrects. Si certains processus reçoivent le message, ils le rediffusent aux autres et tout le monde finira par le recevoir et par le délivrer.

On pourrait utiliser un protocole de diffusion fiable pour envoyer la décision globale "annuler" ou "valider" à tous les processus, ce qui résoudrait le problème du plantage du coordinateur pendant qu'il diffuse la décision et où certains l'ont reçue et pas d'autres.

Si le délai de garde a expiré sur le processus (quand il était dans l'état "choisi oui") et qu'il n'a pas reçu la décision globale soit directement du coordinateur soit indirectement d'un autre processus, alors le processus sait que personne n'a reçu la décision. La seule raison à cela est que le coordinateur a planté avant de la prendre ou de commencer à la diffuser et tous les processus arrivent forcément à cette même conclusion. Dans ce cas, chaque processus peut décider d'annuler localement en sachant que tout le monde fera la même chose : il y aura une cohérence de décision entre tous les processus.

### Exercice 2 : diffusion fiable

Diffusion fiable dans un contexte avec perte de messages pour les 4 cas à étudier.

Note préalable : d'un point de vue théorique, il est impossible de définir un algorithme distribué qui assure la terminaison avec un contexte de panne incluant des pertes de messages. Par principe, on a le cas extrême où tous les messages sont perdus et rien ne fonctionne en conséquence. On va définir ici des algorithmes qui marchent "au mieux" (best-effort).

Rappel sur la définition de la diffusion fiable : tout message diffusé finit par être délivré (une seule fois) par tous les processus corrects (non plantés).

1. Pas de panne de processus et système synchrone

Le diffuseur envoie le message à chaque processus un par un : il attend un acquittement de chaque processus et s'il ne reçoit pas un acquittement, il renvoie le message au processus qui n'a pas acquité et il fait cela tant qu'il n'a pas reçu tous les acquittements (possiblement, ça peut ne jamais arriver

pour la raison évoquée ci-dessus). Il sait combien de temps attendre car on est en synchrone : il sait au bout de combien de temps il devra recevoir l'acquittement (le temps d'aller-retour).

Au niveau d'un récepteur, on peut recevoir les messages en double : si notre acquittement précédent a été perdu, le diffuseur va nous renvoyer le même message car il ne sait pas différencier la perte de son message avec la perte de l'acquittement. Il faut gérer le duplicat des messages reçus pour ne pas délivrer deux fois le même message. Par contre, à chaque duplicat reçu, on renvoie l'acquittement.

## 2. Pas de panne de processus et système asynchrone

On peut appliquer le même fonctionnement que pour le cas précédent sauf qu'en asynchrone, on ne peut pas estimer de manière exacte le temps que mettra l'acquittement à arriver. On va faire une estimation du délai à attendre mais peut-être que des acquittements arriveront après et qu'on aura déjà renvoyé le message au processus. Le problème se gère facilement en détectant les acquittements identiques reçus plusieurs fois par le diffuseur.

## 3. Pannes de processus et système synchrone

Le principe de rediffusion à tout le monde vu en cours permet de gérer la panne de processus (le crash pendant la boucle de diffusion). Ici, on va mélanger les deux systèmes :

- Rediffusion du message la première fois qu'on le reçoit pour gérer la panne de processus.
- Pour toutes les diffusions réalisées (l'initiale ou les rediffusions) : gestion d'acquittement avec renvoi de message pour gérer la perte de messages comme dans le cas 1. Avec une différence importante : on ne redemande pas indéfiniment un acquittement à un processus qui ne répond pas car il peut être planté et ne répondra donc jamais. On limite à un nombre fixe le nombre de redemandes d'acquittement et au bout d'un moment on considère le processus qui ne répond pas comme planté (mais sans être totalement certain que c'est le cas, ça peut aussi être une perte de tous les messages alors qu'il fonctionne correctement).

## 4. Pannes de processus et système asynchrone

Idem que pour 3 sauf qu'on utilise le principe de redemandes de 2 et non pas de 1 pour gérer les duplicats d'acquittement en plus.