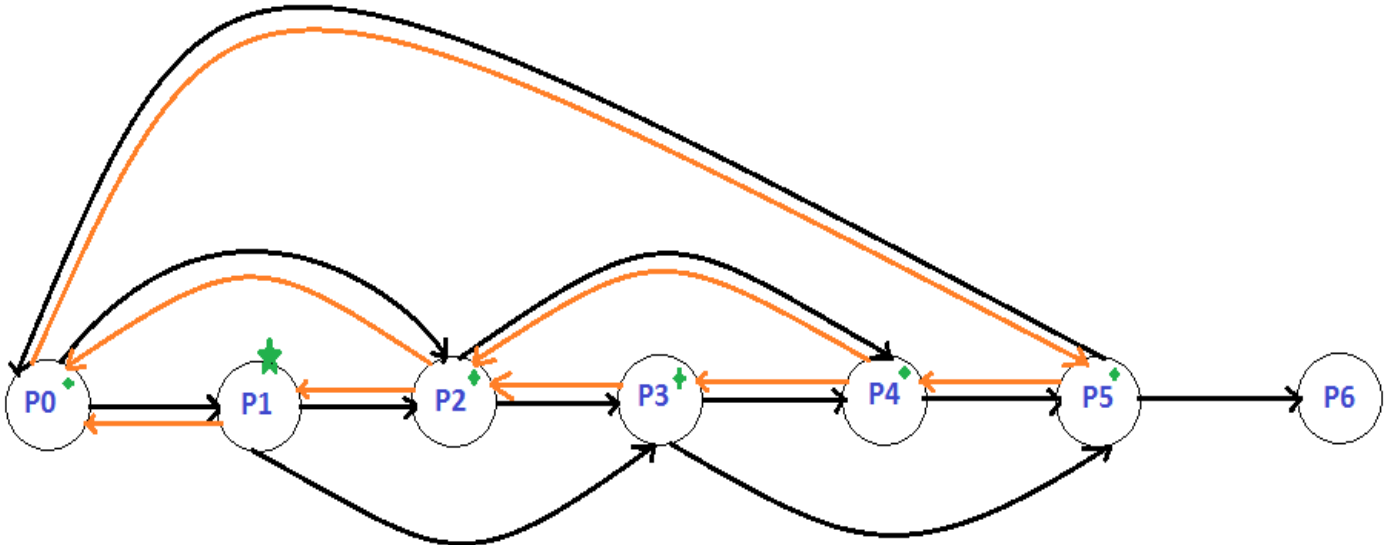
AN ALGORITHMIC APPROACH OF DISTRIBUTED SYSTEMS

EXERCISE 6:

COMMUNICATION DEADLOCK DETECTION ALGO BY CHANDY, MISRA AND HAAS



P1 va initialiser l'envoi de « probe » pour détecter si la communication est « deadlock »: P1→ P2 et P3;

Ce probe est le premier « probe » reçu par ces 2 processus, ils vont marquer P1 comme parent.

A leur tour, P2 et P3 vont envoyer « probe » à ses voisins:

P2→ P3 et P4

P3→ P4 et P5

P3 reçoit le « probe » provenant de P2, et voit que c'est un 2è « probe » vu qu'il est déjà marqué par P1. Il renvoie un « ack » à P2: P3→ P2

Supposons que P4 reçoit en 1er le « probe » de P2 avant celui de P3. P4 va marquer P2 comme parent, tandis qu'il renvoie un « ack » à P3: P4→ P3. P4 fait passer le « probe » à P5: P4→ P5.

Supposons que P5 reçoit en 1er le « probe » de P3 avant celui de P4. P5 va marquer P3 comme parent, tandis qu'il renvoie un « ack » à P4: P5→ P4. P5 fait passer le « probe » à P0 et à P6: P5→ P0 et P6.

P4 reçoit le « ack » de P5 et renvoie un « ack » à son parent P2: P4→ P2.

P2 renvoie un « ack » à son parent P1 puisque P2 a eu un ack de ses successeurs: P2→ P1

P0 reçoit le « probe » de P5 et marque celui-ci comme parent. P0 envoie le « probe » à P1 et P2 (P0→ P1 et P0→P2), et reçoit en retour des « ack » vue que P1 et P2 sont déjà marqués, P1 étant l'initiateur: P2, P1→ P0. Alors P0 renvoie un « ack » à son tour vers son parent P5 car P0 a eu des « ack » provenant de ses successeurs: P0→ P5.

P6, au vue du « probe » envoyé par P5, l'ignore puisque P6 n'attend aucun autre processus. P6 sait juste que P5 attend un message de lui.

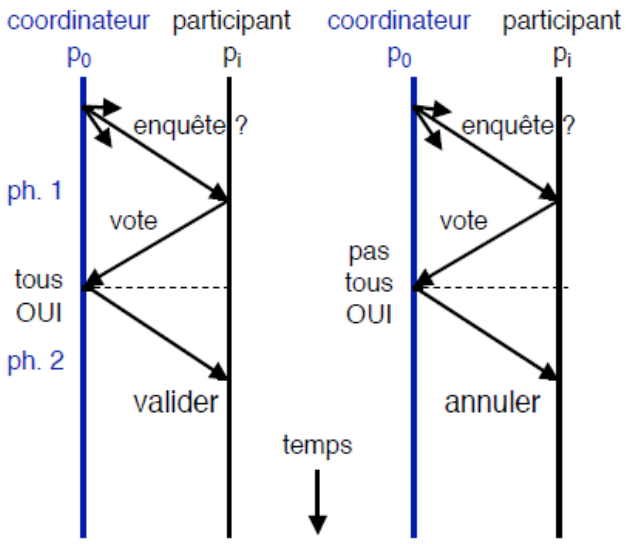
De ce fait, P5 ne pourra pas renvoyer un « ack » vers son parent P3, ni P3 vers P1.

Vu que P1 n'a pas reçu un « ack » de l'un de ses successeurs, P1 ne détecte pas une « communication deadlock »

EXERCISE 6:

TWO-PHASE AND THREE-PHASE COMMIT PROTOCOLS

2PC:

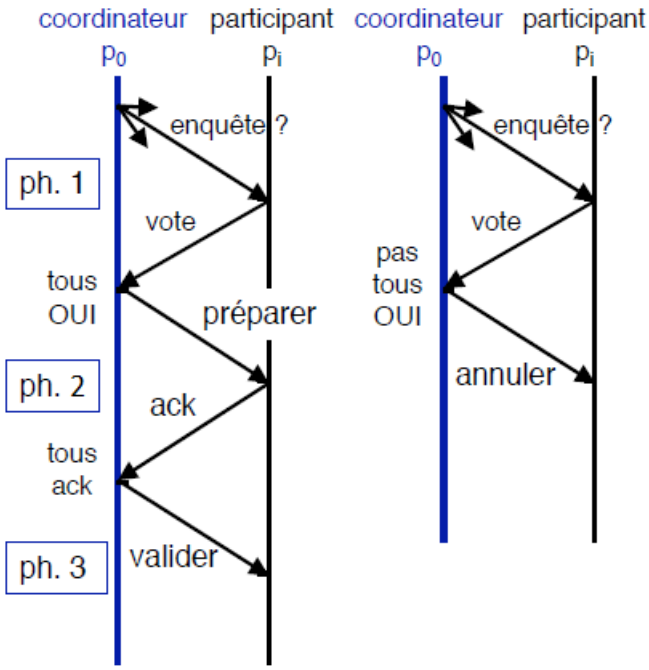


Entre le moment où il a voté OUI et le moment où il reçoit une réponse du Coordinateur, un Participant est dans une zone d'incertitude (un autre Participant peut se trouver soit en phase de validation soit en phase d'annulation, et les deux issues sont possibles). Cela peut conduire à un scénario de blocage :

- Pi a voté OUI et se trouve dans la zone d'incertitude
- Le Coordinateur P0 tombe en panne après avoir envoyé une décision valider ou annuler à certains participants
- Ces Participants tombent en panne à leur tour. Une décision a été prise mais Pi (correct) n'a aucun moyen de la connaître (au moins jusqu'à un éventuel redémarrage du Coordinateur P0)

La possibilité de blocage provient de l'incertitude sur l'état global (pas de connaissance partagée). C'est un modèle en crash-stop.

3PC:



Le nouveau terme introduit « préparer / ack» différencie 3PC du 2PC au moment où tous les Participants Pi ont voté OUI. En effet, :

- le processus qui va faire « ack » après réception du « préparer », sait que tous savent que tous ont voté OUI;
- Dès lors, si le Coordinateur P0 tombe en panne, il est plus facile de faire un consensus entre les Participants corrects:
    - élire un nouveau Coordinateur;
    - prise de décision après interrogation des autres Pi.
- A la reprise après défaillance, le Coordinateur P0 prend les décisions précises selon les cas des Pi

Il n'y aurai pas de situation où Pi soit bloqué du fait que tous connaît l'état de chacun. La connaissance est partagée entre tous les processus.

Références:

« Two-phase commit protocol », « Three-phase commit protocol », « Atomic commit », from Wikipedia

Eric C Cooper, « Analysis of Distributed Commit Protocols », Computer Science Division, EECS Departement, University of California, Berkeley, CA 94720,175-183

Jim GRAY, Leslie LAMPORT, « Consensus on Transaction Commit », 2006,ACM Transactions on Database Systems, Vol.31, No.1,133-160

Sacha Krakowiak, « Validation atomique », 2004, Projet Sardes (INRIA et IMAG-LSR), Université Joseph Fourier, Support de cours, Master 2 Recherche « Systèmes et Logiciel » 2003-2004

Dale Skeen, « Nonblocking Commit Protocols », Computer Science Division, EECS Departement, University of California, Berkeley, CA 94720, 133-143