

## TD – Consensus – M2 ARA

### Coordinateur tournant (Chandra-Toueg 1996)

On considère un système à  $n$  processus  $\{1, 2, \dots, n\}$ . Les processus peuvent subir des fautes franches et les canaux de communication sont fiables. Le nombre de fautes ( $f$ ) est inférieur à  $n/2$ . Les primitives R-Broadcast et R-deliver implémentent une diffusion fiable.

Soit l'algorithme de Coordinateur Tournant dont le pseudo-code est le suivant pour le processus  $p$  :

```
1. procedure propose( $v_p$ )
2.  $estimate_p = v_p$ 
3.  $state_p = undecided$ 
4.  $r_p = 0$ 
5.  $ts_p = 0$ 
6. while  $state_p = undecided$ 
7.    $r_p = r_p + 1$ 
8.    $c_p = (r_p \bmod n) + 1$ 
9.   Phase 1:
10.    send ( $p, r_p, estimate_p, ts_p$ ) to  $c_p$ 
11.   Phase 2:
12.    if  $p = c_p$  then
13.      wait for  $(n+1)/2$  estimations ( $q, r_p, estimate_q, ts_q$ ) from processes  $q$ 
14.       $msgs_p[r_p] = \{(q, r_p, estimate_q, ts_q) \text{ received}\}$ 
15.       $t = \text{largest } ts_q \text{ such that } (q, r_p, estimate_q, ts_q) \in msgs_p[r_p]$ 
16.       $estimate_p = \text{select one } estimate_q \text{ such that } (q, r_p, estimate_q, t) \in msgs_p[r_p]$ 
17.      send ( $p, r_p, estimate_p$ ) to all
18.   Phase 3:
19.    wait until received ( $c_p, r_p, estimate_{cp}$ ) from  $c_p$  or  $c_p \in suspected_p$ 
20.    if received ( $c_p, r_p, estimate_{cp}$ ) from  $c_p$  then
21.       $estimate_p = estimate_{cp}$ 
22.       $ts_p = r_p$ 
23.      send ( $p, r_p, ack$ ) to  $c_p$ 
24.    else
25.      send ( $p, r_p, nack$ ) to  $c_p$  /*  $p$  suspects that  $c_p$  crashed */
26.   Phase 4:
27.    if  $p = c_p$  then
28.      wait for  $(n+1)/2$  ( $q, r_p, ack$ ) or ( $q, r_p, nack$ ) from processes  $q$ 
29.      if  $(n+1)/2$  ( $q, r_p, ack$ ) received then
30.        R-Broadcast ( $p, r_p, estimate_p, decide$ )
31.   R-deliver( $q, r_q, estimate_q, decide$ )
32.   if  $stat = undecided$  then
33.     decide( $estimate_q$ )
34.      $state_p = decided$ 
```

### Question 1

Rappelez les propriétés de la diffusion fiable et du consensus

Réponses

Diffusion fiable :

Validité : 1 processus correct p R-Broadcast m, p délivre m

Accord : 1 processus correct délivre m, tous les corrects délivrent m

Intégrité uniforme : R-deliver au plus 1 fois un message m et ce message a été diffusé par un R-Broadcast

Consensus

Terminaison : Tous les processus corrects décident

[Intégrité] : Tout processus décide 1 fois

Accord : 2 processus corrects ne peuvent pas décider différemment

Validité : Décision sur une valeur proposée

### Question 2

Proposez un algorithme simple de diffusion fiable.

Réponses

R-broadcast(m)

Send m to all

Upon reception of m :

if m received first time

if sender(m) != p

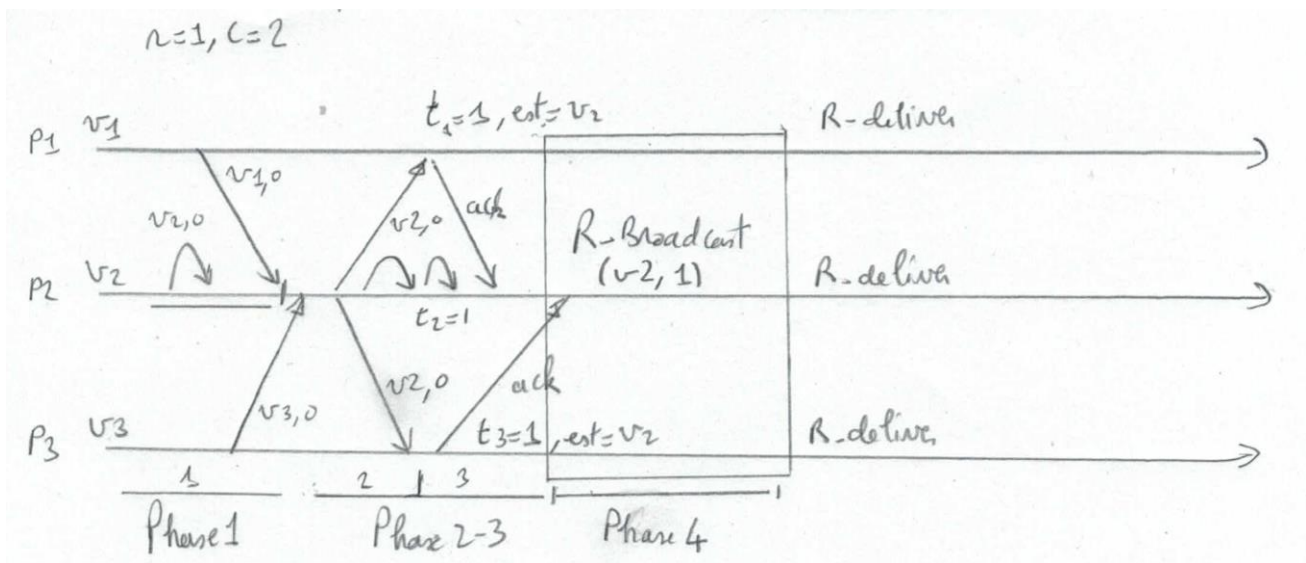
Send m to all

R-deliver(m)

### Question 3

Exécutez l'algorithme sans faute en considérant 3 processus.

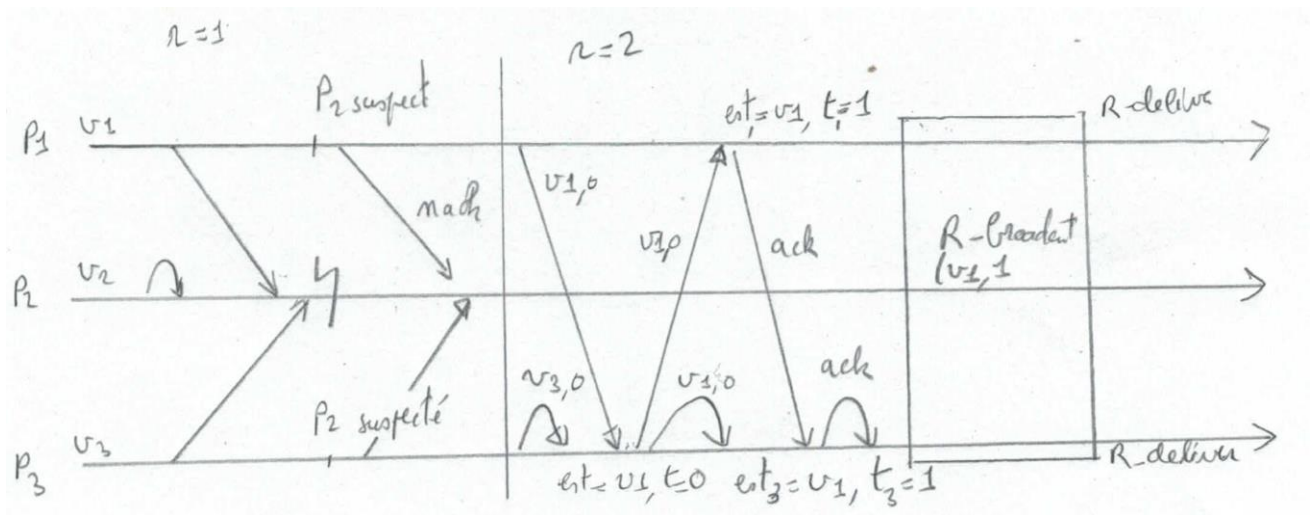
Réponses



### Question 4

Exécutez l'algorithme en considérant 3 processus avec le processus 2 qui tombe en panne.

Réponses

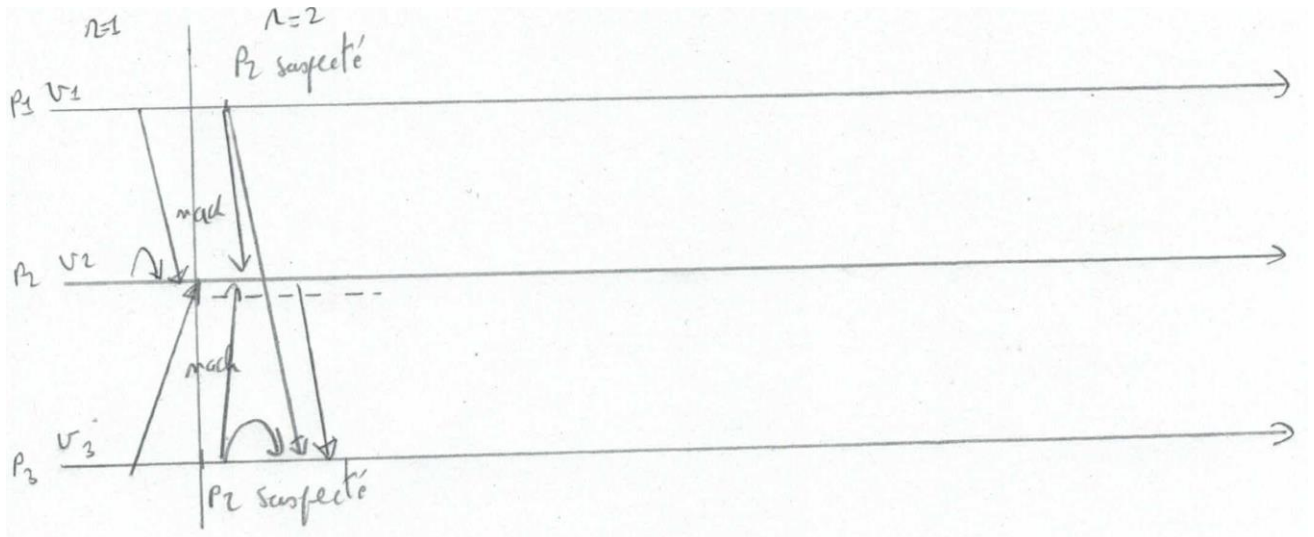


### Question 5

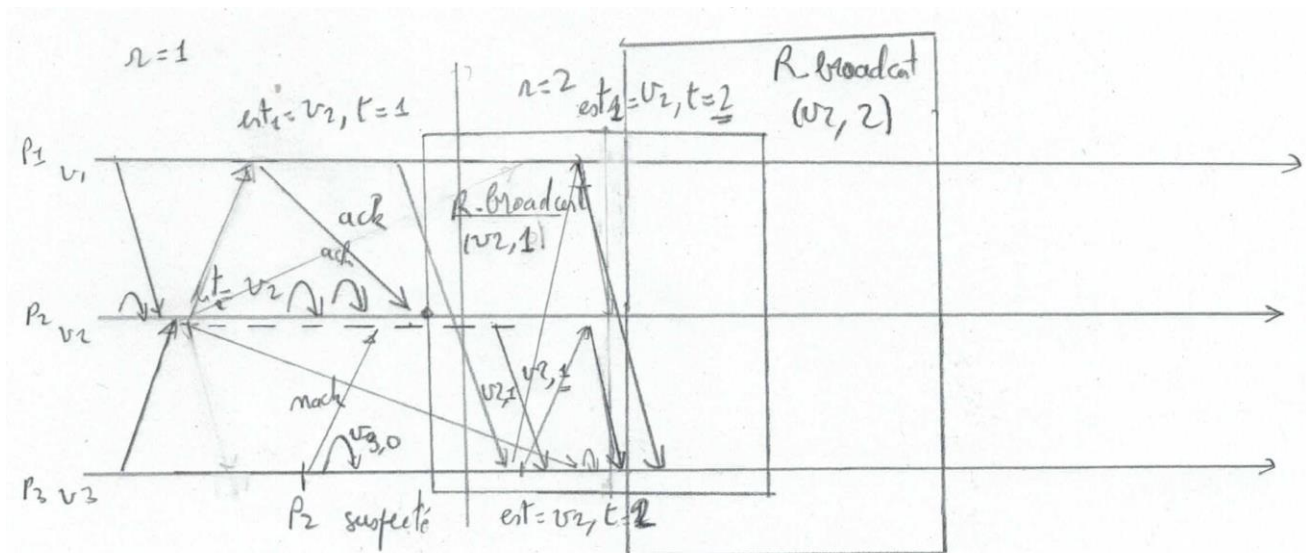
Quel est l'impact d'une fausse suspicion. Est-il possible que deux coordinateurs différents diffusent une décision ? Illustrez votre réponse par un scénario.

#### Réponses

1 majorité suspecte le coordinateur



1 minorité suspecte



### Question 6

Qu'est-ce qu'un accord uniforme ? Démontrez quel type d'accord assure cet algorithme.

Réponses

### Question 7

Quelle est l'utilité du message nack ?

Réponses

Utile pour la terminaison : évite le blocage d'un coordinateur faussement suspecté (ligne 28)

### Question 8

On souhaite maintenant démontrer la terminaison de l'algorithme. On considère différents cas :

- Un processus correct décide : quelle propriété de la diffusion fiable assure la terminaison ?
- En vous appuyant sur la complétude forte du détecteur  $\diamond S$ , montrez que les processus ne peuvent rester indéfiniment bloqués dans une ronde.

c) En vous appuyant sur la justesse faible de  $\delta S$ , montrez qu'il existe un temps à partir duquel, tous les processus décident.

Réponses

a) Un processus correct décide  $\Rightarrow$  1 R-deliver  $\Rightarrow$  accord tous les correct vont faire R-deliver  $\Rightarrow$  tous décide

b) Blocage dans une ronde impossible :

Coordinateur : réception d'une majorité d'estimation (maj. correct + canaux fiable), réception d'une majorité de ack ou nack

Autre processus : soit le coordinateur est suspecté  $\Rightarrow$  changement de ronde, soit on reçoit un message  $\Rightarrow$  changement de ronde

c) Il existe une ronde où 1 coordinateur va accumuler une majorité de ack grâce à la justesse de  $\delta S \Rightarrow$  1 processus n'est à terme plus faussement suspecté  $\Rightarrow$  plus de nack lorsque ce processus ne sera plus coordinateur