

# Système Distribué

## TD01 : Horloges et Diffusion

### 1 Horloge de Lamport

L'horloge de Lamport a été inventée par Leslie Lamport en 1978. Il s'agit du premier type d'horloge logique introduit en informatique. L'horloge de Lamport respecte la dépendance causale ( $e \rightarrow e' \Rightarrow H(e) < H(e')$ ) mais non sa réciproque ( $H(e) < H(e') \not\Rightarrow e \rightarrow e'$ ).

Sur chaque site  $S_i$  on trouve une variable entière  $H_i$  dite horloge locale. La date locale d'un événement  $E$  est notée  $d(E)$ . Avec cette notation, l'algorithme de Lamport fonctionne comme suit :

- Pour chaque événement  $E$  qui correspond à un calcul local, le site  $S_i$  incrémente son horloge locale  $H_i$  et date l'événement  $E$  par  $d(E) = H_i$ .
- Lors de l'émission d'un message  $M$  par  $S_i$ ,  $S_i$  incrémente son horloge locale  $H_i$  et estampille le message  $M$  par  $(H_i, i)$ . Cet événement (l'envoi du message  $M$ ) est daté par  $d(E) = H_i$ .
- Lors de la réception d'un message  $M$  estampillé  $(H_j, j)$  par  $S_i$ ,  $S_i$  recalcule son horloge de la manière suivante :

$$H_i = \text{Max}(H_i, H_j); \quad H_i = H_i + 1;$$

Cet événement (la réception du message  $M$ ) est daté par  $d(E) = H_i$

La date globale  $D(E)$  d'un événement  $E$  est alors notée  $(d(E), i)$  où  $i$  est le numéro du site où à eu lieu l'événement et  $d(E)$  sa date locale sur ce site.

### Exercices :

1. Appliquez cet algorithme à l'exemple illustré sur la figure 1. Donnez la date globale de chaque événement.

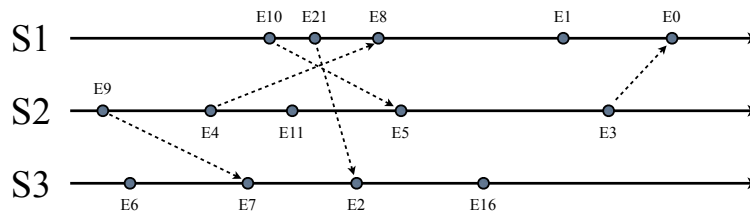


FIGURE 1 – Événements répartis entre 3 sites avec dépendances causales

2. Donnez les événements qui précèdent causalement l'événement  $E_4$ ,  $E_2$ , puis  $E_0$ .
3. Donnez deux événements qui vérifient la relation  $H(e) < H(e')$  mais qui n'ont pas de dépendance causale.
4. En vous basant sur les estampilles de Lamport, donnez un ordre total des événements illustrés sur la figure 1. Qu'en déduisez-vous ?

## 2 Horloge de Mattern

L'horloge de Mattern a été inventé par Mattern et Fidge dans les années 1989 - 1991. Ce type d'horloge assure la réciproque de la dépendance causale. Elle permet également de savoir si 2 événements sont parallèles (non dépendants causalement). Par contre elle ne définit pas un ordre total global.

Sur chaque site  $S_i$  avec  $i = 1, \dots, n$  on définit une horloge vectorielle comme un vecteur  $V_i[1..n]$  initialisé à 0. À chaque événement  $E$  sur le site  $i$ ,  $V_i[i]$  est modifié de la manière suivante :

- $V_i[i] = V_i[i] + 1$  (quel que soit l'événement : calcul local, émission ou réception)
- si l'événement  $E$  correspond à l'envoi d'un message  $M$  par  $S_i$ ,  $M$  est estampillé par  $V_m =$  valeur de l'horloge  $V_i$  au moment de l'envoi.
- si l'événement  $E$  correspond à la réception par  $S_i$  d'un message  $M$  estampillé par le vecteur  $V_m$ ,  $S_i$  modifie son vecteur local comme suit :

$$V_i[k] = \max(V_i[k], V_m[k]) \text{ pour } k = 1, \dots, n$$

Dans tous les cas, l'événement  $E$  est daté par l'horloge  $V_i$ .

On considère les échanges entre 4 sites illustrés sur la figure 2.

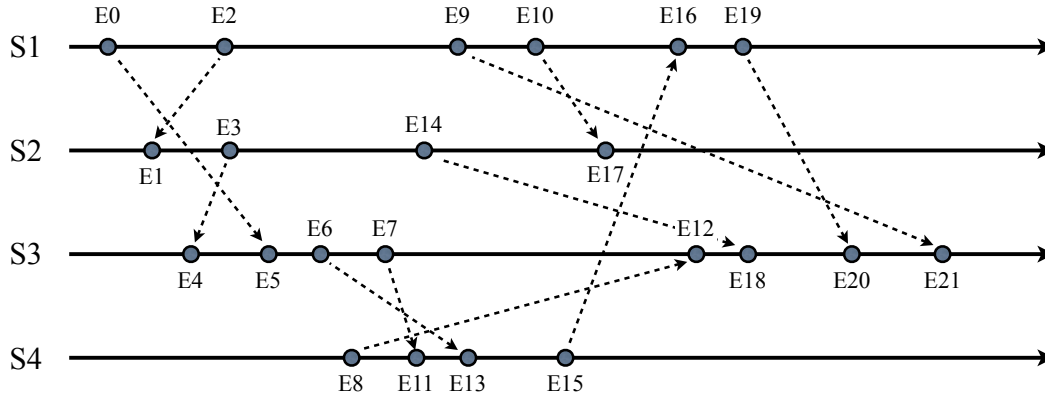


FIGURE 2 – Événements répartis entre 4 sites avec dépendances causales

### Exercices :

1. Datez les événements E0 - E21, puis montrez que les événements E10 et E15 sont indépendants. Qu'en est-il de E2 et E15 ?
2. Donnez l'équation permettant de déterminer si deux événements  $e_i$  et  $e_j$ , estampillés respectivement par les vecteurs  $V_i$  et  $V_j$ , respectent  $e_i \rightarrow e_j$  ?  $e_j \rightarrow e_i$  ? ou bien que  $e_i$  et  $e_j$  sont indépendants ?
3. En vous basant sur les estampilles de Mattern, donnez l'arbre des dépendances causales en rapport avec la figure 2. Qu'en déduisez-vous ?

## 3 Protocole d'ordre causal

Un protocole d'ordre causal est un protocole qui assure que les messages reçus sur un même site sont délivrés en respectant les dépendances causales entre les événements d'émission de ces messages.

Pour un message  $m$ , on notera  $e_m$  son événement d'émission,  $r_m$  son événement de réception et  $d_m$  l'événement de délivrement du message, c'est-à-dire l'événement correspondant au moment où le message sera réellement délivré au processus récepteur (le délivrement peut être décalé dans le futur par rapport à la réception).

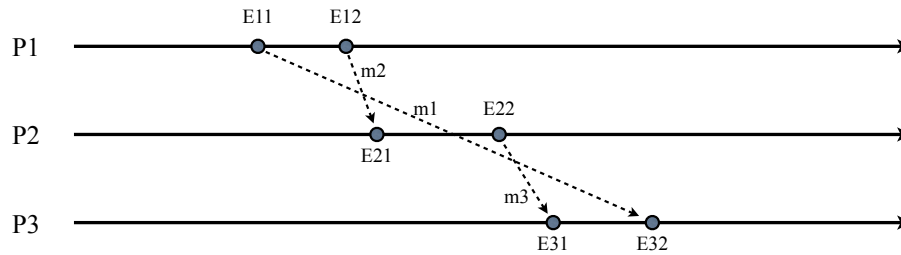


FIGURE 3 – Protocole d'ordre causal

### Exercices :

1. Mettez en évidence le non-respect des dépendances causales en émission pour le chronogramme exposé sur la figure 3. Placez les événements de délivrement des messages sur le chronogramme en respectant ces dépendances causales.
2. Déterminez la relation générale entre les événements associés à 2 messages pour que l'ordre causal de leur émission soit respecté lors de leur délivrement.
3. Montrez que les horloges de Mattern ou de Lamport ne permettent pas de détecter le non-respect des dépendances causales en émission et de bien ordonner les délivrances des messages.

## 4 Diffusion causale

On peut représenter sur un chronogramme la diffusion d'un message comme un ensemble de sous-messages ayant un événement commun d'émission et un événement de réception pour chaque processus (y compris celui qui a lancé la diffusion). Le chronogramme de la figure 4 représente 3 processus qui communiquent via de la diffusion (4 diffusions sont réalisées ici : messages  $m_1, m_2, m_3$  et  $m_4$ ). On considérera dans cet exercice des processus de communications fiables.

Un message  $m$  diffusé est caractérisé par les événements suivants :

- $e_m$  qui est l'émission (la diffusion) du message ;
- pour chaque processus  $P_i$ ,  $r_m^i$  est la réception du message par le système de communication de  $P_i$  ;
- pour chaque processus  $P_i$ ,  $d_m^i$  est le délivrement du message à  $P_i$  par le système de communication de  $P_i$  ;

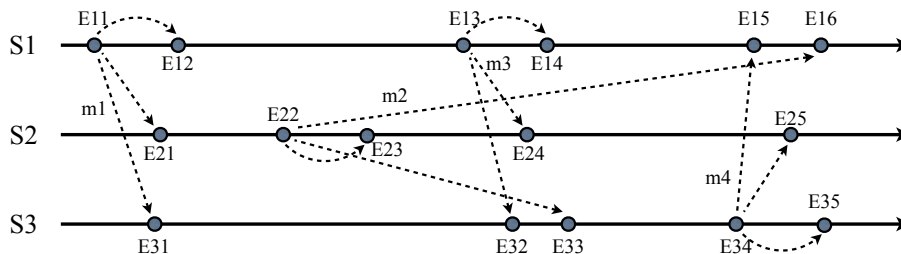


FIGURE 4 – Diffusion de message

Le chronogramme illustré sur la figure 4 représente uniquement les événements d'émissions et le délivrement immédiat d'un message dès sa réception, sans se préoccuper d'assurer des contraintes au niveau de l'ordre de délivrement des messages.

La diffusion causale est une diffusion qui assure que si la diffusion (l'émission) d'un message  $m'$  dépend causalement de la diffusion d'un message  $m$ , alors tout processus délivre  $m$  avant  $m'$ .

### Exercices :

1. Mettez en évidence sur le chronogramme de la figure 4 un non-respect des conditions de la diffusion causale.
2. Dans le cas des horloges vectorielles, déterminez la condition pour qu'un site  $S_j$  délivre un message  $m$  reçu, daté par  $V_m$  et envoyé par un site  $S_i$ , tout en respectant la diffusion causale.
3. Datez les événements du chronogramme selon l'algorithme **CBCAST** en remplissant le tableau suivant. Comment cette datation permet de déterminer si la diffusion causale est respectée et le cas échéant de retarder le délivrement d'un message ?

| Evt | Site | H avant | Mess reçu | Action | Mess émis | H après |
|-----|------|---------|-----------|--------|-----------|---------|
| E11 | S1   |         |           |        |           |         |
| E12 | S1   |         |           |        |           |         |
| E21 | S2   |         |           |        |           |         |
| E31 | S3   |         |           |        |           |         |
| E22 | S2   |         |           |        |           |         |
| E23 | S2   |         |           |        |           |         |
| E13 | S1   |         |           |        |           |         |
| E14 | S1   |         |           |        |           |         |
| E24 | S2   |         |           |        |           |         |
| E32 | S3   |         |           |        |           |         |
| E33 | S3   |         |           |        |           |         |
| E34 | S3   |         |           |        |           |         |
| E35 | S3   |         |           |        |           |         |
| E25 | S2   |         |           |        |           |         |
| E15 | S1   |         |           |        |           |         |
| E16 | S1   |         |           |        |           |         |
|     |      |         |           |        |           |         |

### Rappel :

L'algorithme **CBCAST** (Causal Broadcast) fonctionne comme suit :

- Chaque site  $S_i$  utilise une horloge vectorielle  $H_i[N]$  (N sites).
- Pour chaque message  $m$  à diffuser par  $S_i$ ,  $S_i$  traite  $m$  puis incrémente son horloge vectorielle  $H_i[i]++$ , et estampille  $m$  par  $V_m = H_i$ .
- À la réception d'un message  $m$  diffusé par  $S_i$  et estampillé par  $V_m$ , le récepteur  $S_j$  le met en attente avant délivrance jusqu'à ce que la condition (2) soit respectée.
- A la délivrance du message  $m$  envoyé par  $S_i$ ,  $S_j$  incrémente son horloge vectorielle comme suit  $H_j[i]++$  (sauf si  $S_j = S_i$  alors on fait rien).