

# Système Distribué

## TD01 : Gestion d'horloges

### 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 ne correspond pas à l'envoi ou à la réception d'un message, 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 :

$$\text{si } H_i < H_j \text{ alors } H_i = H_j + 1 \text{ sinon } H_i = H_i + 1$$

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 a 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.

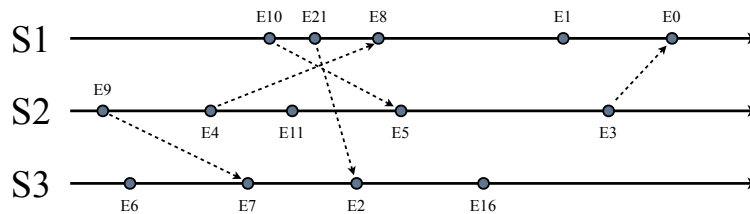


FIGURE 1 – Événements répartis entre 3 sites avec dépendances causales

**Corrections :**

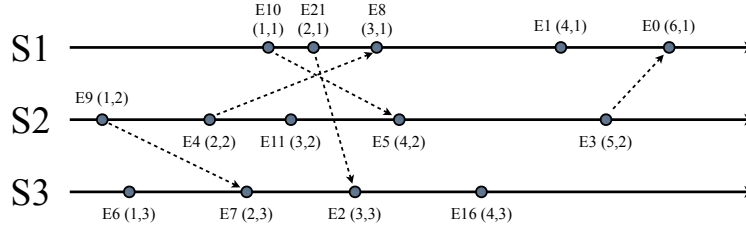


FIGURE 2 – Corrections

2. Donnez deux événements qui vérifient la relation  $H(e) < H(e')$  mais qui n'ont pas de dépendance causale.

**Corrections :**

Les événements  $E_2$  et  $E_1$  vérifient la relation  $(3,3) < (4,1)$  mais n'ont pas de dépendances causales. Idem pour les événements  $E_4$  et  $E_{16}$ .

3. En vous basant sur les estampilles de Lamport, donnez un ordre total des événements illustrés sur la figure 1. Qu'en déduisez-vous ?

**Corrections :**

Pour obtenir un ordre total, il suffit d'ordonner les événements en fonction de leurs estampilles. Ainsi, nous obtenons l'ordre suivant :

$$E_{10} \rightarrow E_9 \rightarrow E_6 \rightarrow E_{21} \rightarrow E_4 \rightarrow E_7 \rightarrow E_8 \rightarrow E_{11} \rightarrow E_2 \rightarrow E_1 \rightarrow E_5 \rightarrow E_{16} \rightarrow E_3 \rightarrow E_0$$

Nous pouvons en déduire que les horloges de Lamport induisent un chaînage de tous les événements, c'est-à-dire que des événements indépendants sont artificiellement ordonnés. Cela vient du fait que l'horloge de Lamport ne respecte pas la réciproque de l'ordre causal ( $H(e) < H(e') \not\Rightarrow e \rightarrow e'$ ).

## 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. A chaque événement  $E$ ,  $V_i[i]$  est incrémenté de la manière suivante :

- $V_i[i] = V_i[i] + 1$  (à chaque événement sur  $i$ )
- si l'événement  $E$  correspond à l'envoi d'un message  $M$  par un site  $S_i$ ,  $M$  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  $V_m$ ,  $S_i$  incrémente 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 3.

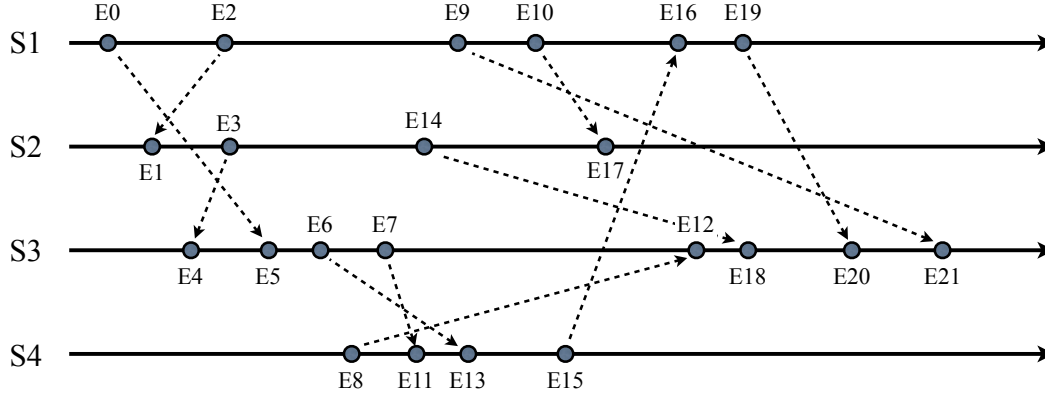


FIGURE 3 – Événements répartis entre 4 sites avec dépendances causales

### Exercices :

1. Montrez que les événements  $E_{10}$  et  $E_{15}$  sont indépendants. Qu'en est-il de  $E_2$  et  $E_{15}$  ?

### Corrections :

Événements	Horloges vectorielles
E0	(1,0,0,0)
E1	(2,1,0,0)
E2	(2,0,0,0)
E3	(2,2,0,0)
E4	(2,2,1,0)
E5	(2,2,2,0)
E6	(2,2,3,0)
E7	(2,2,4,0)
E8	(0,0,0,1)
E9	(3,0,0,0)
E10	(4,0,0,0)

Événements	Horloges vectorielles
E11	(2,2,4,2)
E12	(2,2,5,1)
E13	(2,2,4,3)
E14	(2,3,0,0)
E15	(2,2,4,4)
E16	(5,2,4,4)
E17	(4,4,0,0)
E18	(2,3,6,1)
E19	(6,2,4,4)
E20	(6,4,6,4)
E21	(6,4,7,4)

L'estampille de l'événement  $E_{10}$  est égale à (4,0,0,0) et celle de l'événement  $E_{15}$  est égale à (2,2,4,4). Nous voyons donc que la première colonne de l'estampille de  $E_{10}$  est supérieure à celle de  $E_{15}$  alors que les colonnes suivantes de l'estampille de  $E_{10}$  sont inférieures à celles de l'estampille de  $E_{15}$ . Cela signifie que l'équation :

$$V_{E_{10}}[k] \leq V_{E_{15}}[k] \text{ ou } V_{E_{10}}[k] \geq V_{E_{15}}[k] \forall k \text{ et } \exists j \text{ tel que } V_{E_{10}}[j] < V_{E_{15}}[j] \text{ (respectivement } V_{E_{10}}[j] > V_{E_{15}}[j])$$

n'est pas vérifiée. Il n'y a donc pas de causalité entre les événements  $E_{10}$  et  $E_{15}$ . Par conséquent, ils sont indépendants.

Le même raisonnement s'applique pour les événements  $E_2$  et  $E_{15}$ . Or nous voyons que  $V_{E_{15}}[k] \geq V_{E_2}[k] \forall k$  et  $V_{E_{15}}[1] > V_{E_2}[1]$  ce qui signifie qu'il y a une causalité entre  $E_{15}$  et  $E_2$ . En l'occurrence  $E_2$  est antérieur à  $E_{15}$ .

2. En vous basant sur les estampilles de Mattern, donnez l'arbre des dépendances causales en rapport avec la figure 3. Qu'en déduisez-vous ?

### Corrections :

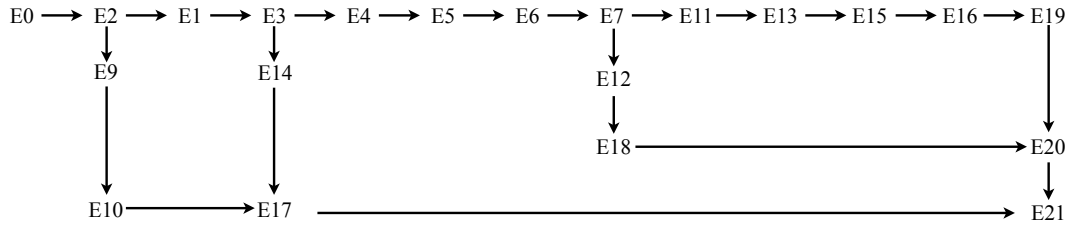


FIGURE 4 – Arbre des dépendances causales

### 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 et 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).

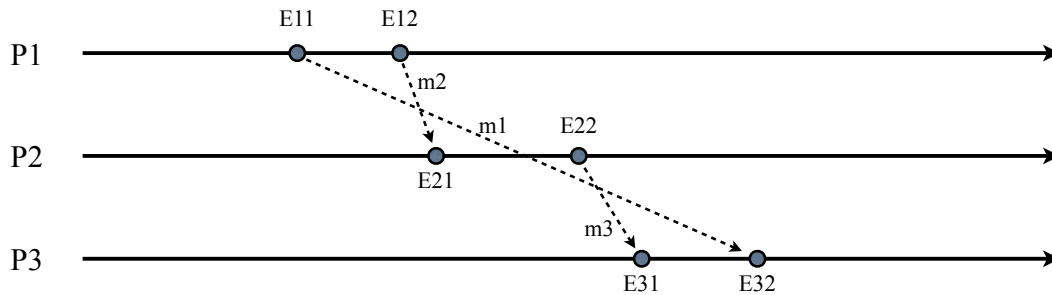


FIGURE 5 – Protocole d'ordre causal

#### Exercices :

1. Mettez en évidence le non-respect des dépendances causales en émission pour le chronogramme exposé sur la figure 5. Placez les événements de délivrement des messages sur le chronogramme en respectant ces dépendances causales.

#### Corrections :

A la vue du chronogramme, l'émission du message  $m_3$  dépend de la réception du message  $m_2$  qui a été envoyé après le message  $m_1$ . Or nous pouvons observer que le processus  $P_3$  reçoit en premier le message  $m_3$  avant le message  $m_1$ . Si les messages sont délivrés dès leur arrivée sur les sites, alors la relation de causalité entre les émissions de  $m_1$  et  $m_3$  est violée. La figure 6 contient les événements de délivrement des messages qui permettent de respecter l'ordre causal.

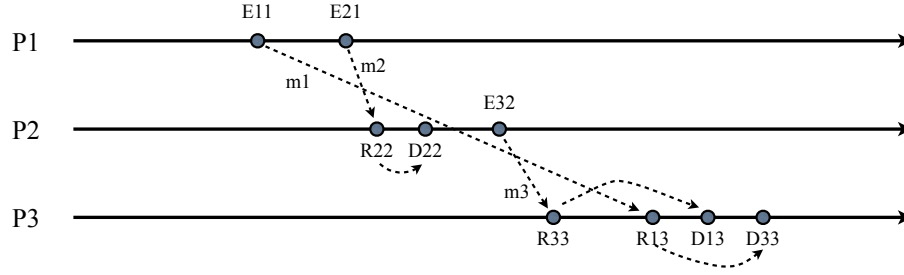


FIGURE 6 – Dépendances causales respectées avec le délivrement des messages

- 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.

**Corrections :**

Dans un système, on dira que l'ensemble des messages respecte l'ordre causal si :

$\forall P_i, P_j, P_k \forall m_1$  émis sur  $C_{i,j}$ ,  $\forall m_2$  émis sur  $C_{k,j}$  et la relation précède noté  $\rightarrow$  :

$$e_i(m_1) \rightarrow e_k(m_2) \Rightarrow del_j(m_1) \rightarrow del_j(m_2)$$

- 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.

**Corrections :**

Avec des horloges de Lamport, le message  $m_3$  serait estampillé par (4,2) et donc l'événement  $R_{33}$  aurait pour estampille (5,3). A cet instant,  $P_3$  n'a aucun moyen de savoir qu'un autre message ( $m_1$ ) antérieur à  $m_3$  doit être délivré avant  $m_3$ . On ne peut donc pas retarder le délivrement de  $m_3$  avec des horloges de Lamport.

Nous pouvons appliquer le même raisonnement avec les horloges de Mattern : l'événement  $R_{33}$  aurait pour estampille (2,2,1) ce qui signifie que 2 événements ont déjà eu lieu sur  $P_1$  et sur  $P_2$  et il s'agit du premier événement sur  $P_3$ . Or, comme il ne s'agit pas d'une diffusion,  $P_3$  n'a aucun moyen de savoir si les événements précédents de  $P_1$  et  $P_2$  le concernent. On ne peut donc pas retarder le délivrement de  $m_3$  avec des horloges de Mattern.

## 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 7 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  $P_i$  ;

Le chronogramme illustré sur la figure 7 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.

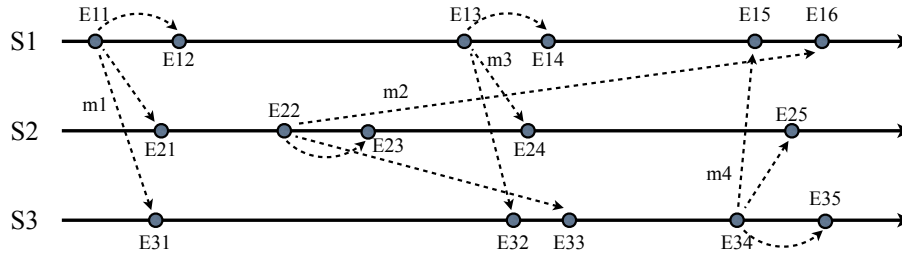


FIGURE 7 – Diffusion de message

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. Déterminez la relation pour accepter un message tout en respectant la diffusion causale.

#### Corrections :

A la réception d'un message  $m$  envoyé par  $P_i$  et estampillé par  $V_m, S_j$  le met en attente jusqu'à ce que :

- $V_m[i] = H_j[i] + 1$  et
- $\forall k \neq i, V_m[k] \leq H_j[k]$

2. Mettez en évidence sur le chronogramme de la figure 7 un non-respect des conditions de la diffusion causale.
3. Datez les événements du chronogramme avec les horloges de Mattern. Est-ce que cette datation permet de déterminer si la diffusion causale est respectée et le cas échéant de retarder le délivrement d'un message ? Pourquoi ?
4. Proposez une version restreinte des horloges de Mattern où tous les événements ne sont pas datés et qui permette d'assurer les conditions de la diffusion causale en pouvant retarder au besoin le délivrement de messages.