Université de Strasbourg
UFR Mathématique et Informatique

M1 Informatique
2021/2022

Algorithmes Distribués

TD nº 1 : Temps Logique (Horloges & Diffusion)

CORRECTION

I 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 ($e \overset{\bullet}{\to} e' \Rightarrow H(e) < H(e')$) mais non sa réciproque ($H(e) < H(e') \not\Rightarrow e \overset{\bullet}{\to} e'$).

Sur chaque site S_i on trouve une variable entière H_i dite horloge locale, initialisée à 0. 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 recale 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.

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

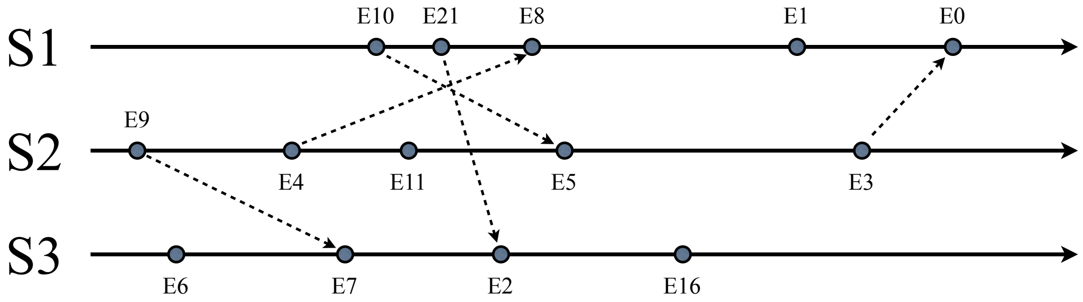

FIGURE 1 – Evénements répartis entre 3 sites avec dépendances causales

Q 2. Donnez les événements qui précèdent causalement l'événement E_4, E_2, puis E_0.

$\{E_9\} \to E_4$, $\{E_6, E_9, E_7, E_{21}, E_{10}\} \overset{\bullet}{\to} E_2$. Tous les événements E_x se produisant sur S_1 et S_2 vérifient $E_x \overset{\bullet}{\to} E_0$.

Q 3. Donnez deux événements qui vérifient la relation $H(e) < H(e')$ mais qui n'ont pas de dépendance causale.

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}.

Q 4. En vous basant sur les estampilles de Lamport, donnez un ordre total des événements illustrés sur la figure 1. Qu'en déduisez-vous ?

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

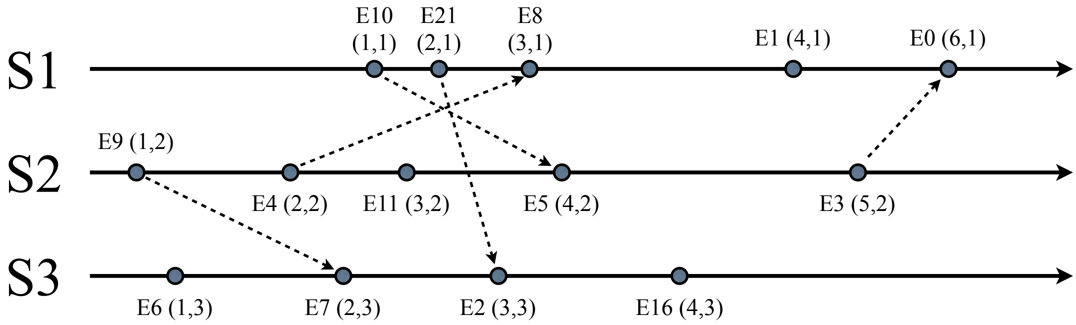

FIGURE 2 – Correction avec estampilles de Lamport

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

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 la précédence causale $(H(e) < H(e') \not\Rightarrow e \overset{\bullet}{\rightarrow} e')$.

II 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 de savoir si deux événements sont concurrents (i.e. parallèles ou non dépendants causalement). Par contre elle ne définit pas un ordre total. Pour finir, alors qu'elle fournit toutes les informations nécessaires pour la définition de la précédence causale (la dépendance causale pour la gestion de la réception des messages), elle ne suffit pas pour garantir l'ordre causal dans le cas général (pour la livraison des messages).

Sur chaque site S_i avec $i = 1, ..., n$ on définit une horloge vectorielle comme un vecteur $V_i[1...n]$ initialisé à 0. Á 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 :

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

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.
Q 1. Datez les événements E0 - E21, puis montrez que les événements E10 et E15 sont indépendants. Qu'en est-il de E2 et E15 ?

Evénements	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)

Evénements	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	(6,3,7,4)
E21	(6,3,8,4)

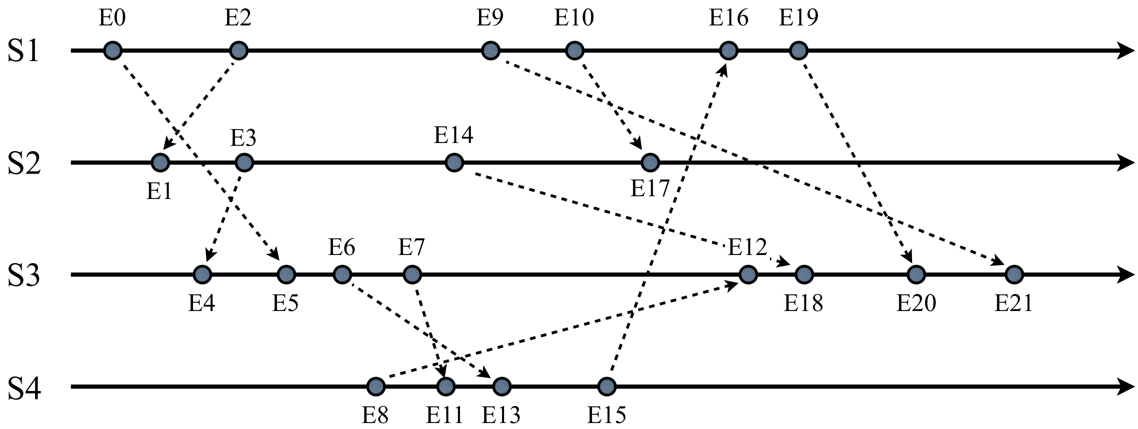

FIGURE 3 – Evénements répartis entre 4 sites avec dépendances causales

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'occurence E_2 est antérieur à E_{15}.

Q 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 sur n sites, vérifient $e_i \overset{\bullet}{\to} e_j$? ou bien que e_i et e_j sont concurrents ? Montrez, dans le cas général, que précédence directe et indirecte ne sont pas distinguables avec seulement deux estampilles. Commentez.
$e_i \overset{\bullet}{\to} e_j := \forall k | 1 \leq k \leq n, V_i[k] \leq V_j[k]$.
$e_j || e_i := \exists k | 1 \leq k \leq n, V_j[k] < V_i[k] \wedge \exists m | 1 \leq m \leq n, V_i[m] < V_j[m]$.
Il suffit de produire un contre exemple (avec les mêmes estampilles mais avec les deux situations) et cela implique qu'une troisième horloge/estampille est nécessaire pour conclure ou se contenter de cas particuliers.

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

Ce graphe est orienté et acyclique, i.e., *Directed Acyclic Graph* (DAG). On observe qu'après certains événements, plusieurs autres sont possibles (d'après les estampilles qu'ils portent), sans qu'on puisse les ordonner.

Exemple : après $E2$, $E1$ et $E9$ se produisent sans qu'on puisse ordonner ces deux événements.

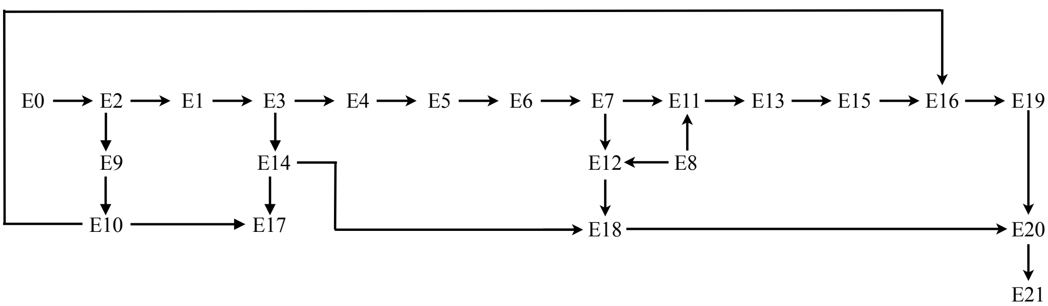

FIGURE 4 – Graphe des dépendances causales

Par conséquent, on peut en déduire que les estampilles de Mattern ne permettent pas d'établir un ordre total.

Q 4. Démontrez la propriété fondamentale des estampilles vectorielles (Charron-Bost) : $e \overset{\bullet}{\to} e' \Leftrightarrow EV_e \subseteq EV_{e'}$

Commençons par l'implication $e \xrightarrow{\bullet} e' \Rightarrow EV_e \subseteq EV_{e'}$ que nous pouvons démontrer simplement en utilisant les définitions :

$$e \xrightarrow{\bullet} e' \Rightarrow Passé(e) \subseteq Passé(e') \Rightarrow \forall i, EV_e[i] \leq EV_{e'}[i] \Rightarrow EV_e \subseteq EV_{e'}$$

Démontrons maintenant l'autre sens de l'équivalence en utilisant sa contraposée, i.e. $e||e' \Rightarrow EV_e||EV_{e'}$:
Soient S_i et S_j les deux sites auxquels appartiennent respectivement e et e'. On sait que $i \neq j$ car $e||e'$, notons alors $Passé_j(e)$ l'ensemble des éléments du passé de e sur S_j. Soit $x(e,j)$ le plus grand événement de $Passé_j(e)$ (le plus proche dans son passé). On a $x(e,j) \xrightarrow{\bullet} e'$ car e' et $x(e,j)$ ont lieu sur S_j et $e' \notin Passé(e)$. En effet, en raisonnant par l'absurde, on sait que $e' \xrightarrow{\bullet} x(e,j)$ implique par transitivité $e' \in Passé(e)$. Donc $Passé_j(e) = Passé_j(x(e,j)) \subset Passé_j(e') \Rightarrow EV_e[j] < EV_{e'}[j]$. Comme le même raisonnement sur e et $y(e',i)$ nous amène à conclure de manière analogue que $EV_{e'}[i] < EV_e[i]$ on a bien $EV_e||EV_{e'}$.

III Protocole d'ordre causal

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

Pour un message m, on notera $e(m)$ sa date d'émission, $r(m)$ sa date de réception et $l(m)$ sa date de livraison, c'est-à-dire le moment où le message sera réellement livré au processus récepteur (la livraison peut être décalée dans le futur par rapport à la réception).

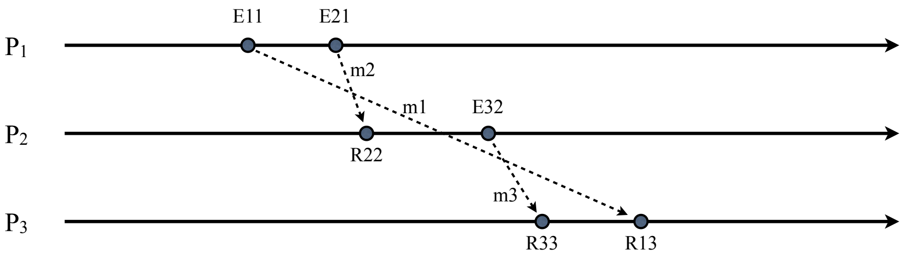

FIGURE 5 – Protocole d'ordre causal

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

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 livrés dès leur arrivée sur les sites, alors la relation de causalité entre les émissions de m_1 et m_3 n'est pas satisfaite. La figure 6 contient les événements de livraison des messages qui permettent de respecter l'ordre causal.

Q 2. Déterminez la relation générale entre les événements associés à deux messages pour que l'ordre causal de leur émission soit respecté lors de leur livraison.

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 de précèdence notée $\xrightarrow{\bullet}$:

$$e_i(m_1) \xrightarrow{\bullet} e_k(m_2) \Rightarrow l_j(m_1) \xrightarrow{\bullet} l_j(m_2)$$

Q 3. Montrez que les horloges de Lamport (scalaires) et celles de Mattern (vectorielles) ne permettent pas de détecter le non-respect des dépendances causales en émission et de bien ordonner les livraisons des messages.

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 livré avant m_3. On ne peut donc pas retarder la livraison 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 deux événements ont déjà eu lieu sur P_1 et sur P_2 et il s'agit du premier

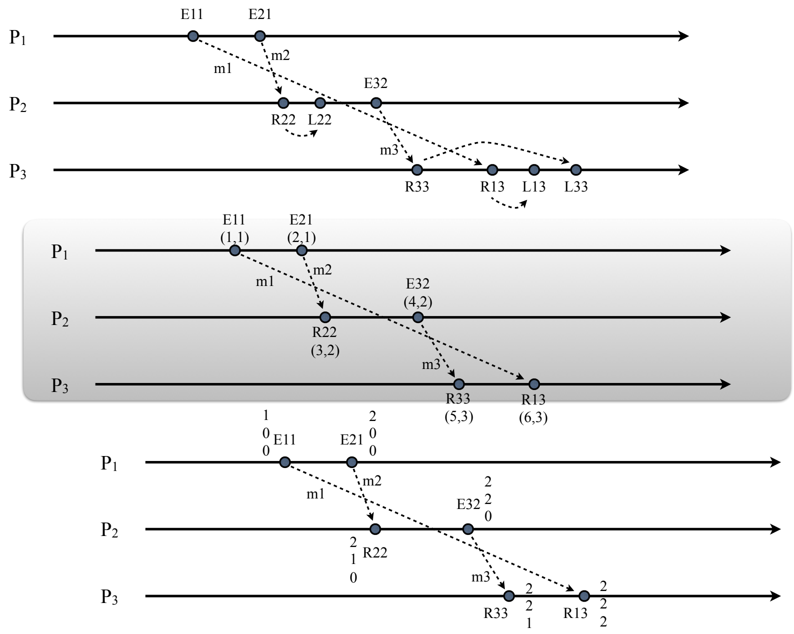

FIGURE 6 – Dépendances causales satisfaites avec la livraison des messages

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 la livraison de m_3 avec des horloges de Mattern.

IV 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. Le chronogramme de la figure 7 représente trois processus qui communiquent par diffusion de message (quatre diffusions sont réalisées ici : messages m_1, m_2, m_3 et m_4). On considérera dans cet exercice des processus fiables en terme de communication.

Un message m diffusé est caractérisable par les événements suivants :
- $e_i(m)$ qui est la date d'émission du message (sa diffusion depuis P_i) ;
- pour chaque processus P_i, $r_i(m)$ est la date de réception du message par son système de communication ;
- pour chaque processus P_i, $l_i(m)$ est la date de livraison du message (sur P_i) par son système de communication.

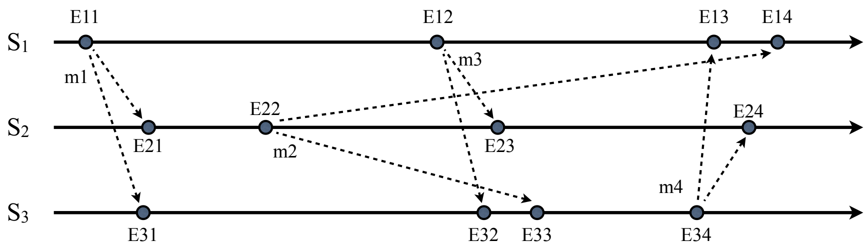

FIGURE 7 – Diffusion de message

Le chronogramme illustré sur la figure 7 représente uniquement les événements d'émission et la livraison immédiate d'un message dès sa réception, sans se préoccuper d'assurer des contraintes au niveau de l'ordre de livraison des messages ($l_i(m) = r_i(m)$).

La diffusion causale est un modèle de communication qui assure que si l'émission par diffusion d'un message m' dépend causalement de la diffusion d'un message m, alors tout processus livrera m avant m'.

Q 1. Mettez en évidence sur le chronogramme de la figure 7 un non-respect des conditions de la diffusion causale.

La diffusion du message m_4 dépend causalement de la diffusion du message m_2. Par conséquent, le processus S_1 devrait livrer le message m_2 avant le message m_4 ce qui n'est pas le cas sur la figure 7.

Q 2. Dans le cas des horloges vectorielles, déterminez la condition pour qu'un site S_j livre un message m reçu, daté par V_m et envoyé par un site S_i, tout en respectant la diffusion causale.

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

— $V_m[i] = H_j[i] + 1$ et

— $\forall k \neq i, V_m[k] \leq H_j[k]$

Autrement dit, si un message reçu de S_i porte une estampille en avance sur S_j alors S_j attend au moins d'atteindre cette date-ci (sauf pour FIFO pour le canal entre i et j où un écart de 1 est nécessaire) pour livrer (et donc traiter) le message.

Q 3. Datez les événements du chronogramme selon l'algorithme **CBCAST** en remplissant le tableau suivant. Comment ce système d'horodatage permet de déterminer si la diffusion causale est respectée et, le cas échéant, de retarder la livraison d'un message ?

Evt	Site	H avant	Mess reçu	Action	Mess émis	H après
E11	S1	H1=[000]		traiter m1, H1[1]++	m1([100],1)	H1=[100]
E21	S2	H2=[000]	m1([100],1)	livrer m1, H2[1]++		H2=[100]
E31	S3	H3=[000]	m1([100],1)	livrer m1, H3[1]++		H3=[100]
E22	S2	H2=[100]		traiter m2, H2[2]++	m2([110],2)	H2=[110]
E12	S1	H1=[100]		traiter m3, H1[1]++	m3([200],1)	H1=[200]
E23	S2	H2=[110]	m3([200],1)	livrer m3, H2[1]++		H2=[210]
E32	S3	H3=[100]	m3([200],1)	livrer m3, H3[1]++		H3=[200]
E33	S3	H3=[200]	m2([110],2)	livrer m2, H3[2]++		H3=[210]
E34	S3	H3=[210]		traiter m4, H3[3]++	m4([211],3)	H3=[211]
E24	S2	H2=[210]	m4([211],3)	livrer m4, H2[3]++		H2=[211]
E13	S1	H1=[200]	m4([211],3)	temporise m4 car V4=[211] > [201]		H1=[200]
E14	S1	H1=[200]	m2([110],2)	livrer m2, H1[2]++		H1=[210]
		H1=[210]	m4([211],3)	livrer m4, H1[3]++ $l_1(m4) \neq r_1(m4)$		H1=[211]

En utilisant des horloges vectorielles datant seulement les émissions des diffusions, il est possible de détecter l'ordre causal de celles-ci. C'est ainsi que l'algorithme détecte *a posteriori* que certains messages déjà diffusés mais non encore reçus doivent être livrés avant le (ou les) messages déjà reçu(s), et procède alors à la temporisation de ce(s) message(s) mis en attente jusqu'à réception du ou des message(s) non encore reçu(s).

V Estampilles matricielles

Représentez l'évolution des horloges matricielles des processus P_1, P_2 et P_3 de l'exemple donné en figure **??**. Annotez directement vos estampilles sur le schéma de la figure **??** depuis l'état initial (événements a, j, u) jusqu'aux événements pour lesquels l'estampille matricielle est déjà fournie (événements e, p, z).
Rappelez pourquoi l'ordre de livraison causale est respectée.

L'ordre de livraison causal est respecté car la livraison d'un message en avance (s'il existe par opposition à l'existence de messages en retard) est retardée jusqu'à ce que les messages en retard soient réceptionnés et traité dans l'ordre et donc avant celui en avance. La définition des états en *avance* et en *retard* est celle de l'ordre de livraison causal rappelée dans le TD et vérifiée au moyen des conditions données ci-dessus. La figure 8 fournit la correction des estampilles matricielles du scénario donné en 8.

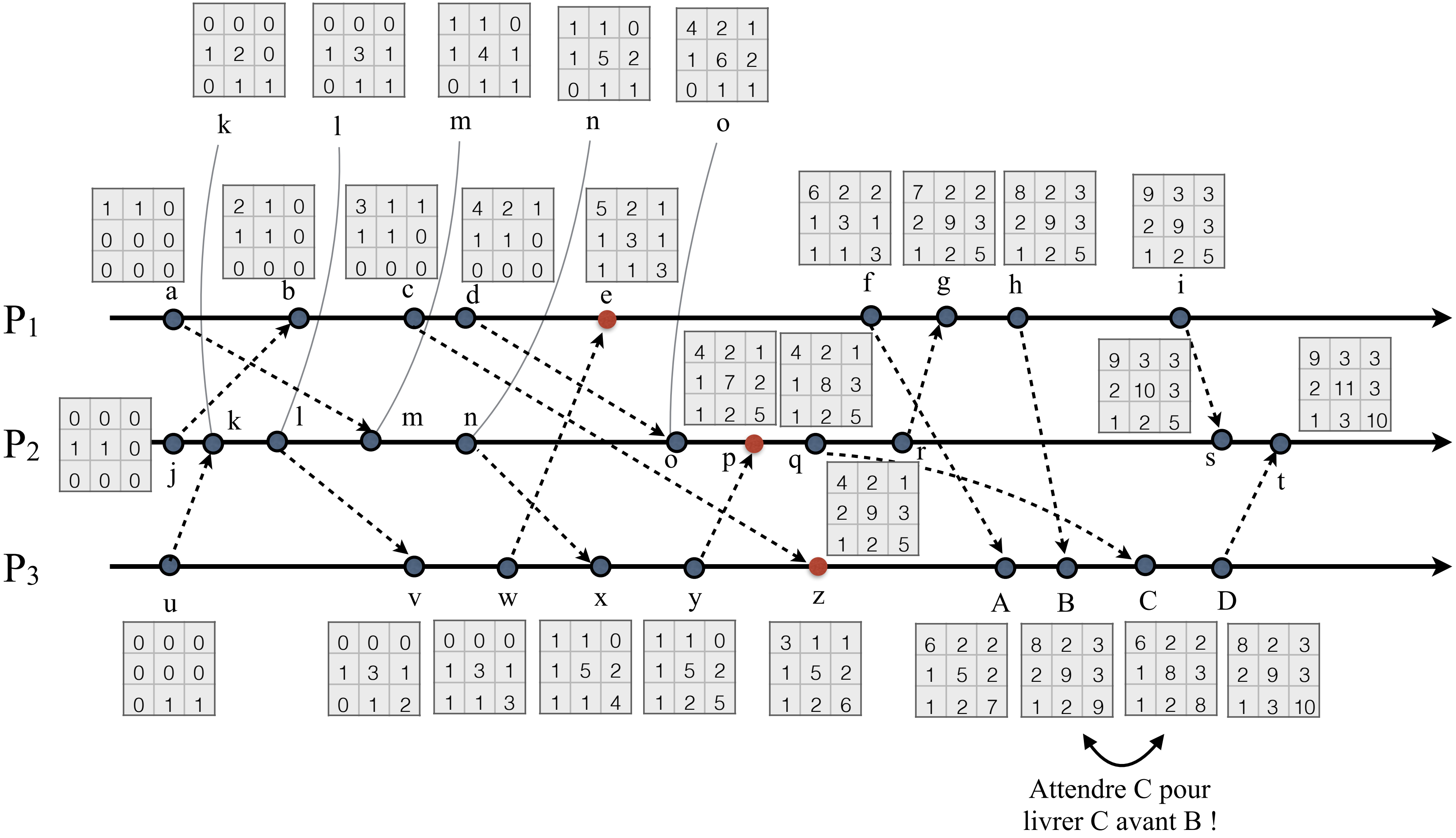

FIGURE 8 – Horloges matricielles - correction