

Conception des systèmes répartis

3ième Année Informatique et Mathématiques Appliquées
Corrigé

Décembre 2003

Quelques problèmes de causalité entre flux multimedia

On considère une application échangeant des flux multimedia de type MPEGx par exemple. Un site A diffuse un flux vidéo vers 2 autres sites B et C. Après avoir reçu le début du flux émis par A, le site B émet un flux à son tour (flux audio par exemple) vers les sites A et C. Le chronogramme de la figure (1) donne une vision événementielle du déroulement de l'exécution répartie en ne conservant que les événements de début et fin de flux qu'il s'agisse d'un flux en émission ou d'un flux en réception.

Questions

1. Sur le chronogramme de la figure (1), les paires d'événements suivants sont-elles liées par la relation de causalité : $(A.d1, B.d2)$, $(A.f1, B.f2)$, $(A.d2, B.f2)$, $(A.d2, C.f1)$? Justifiez vos réponses.

Réponse

$A.d1 \prec B.d2$	car	$A.d1 \prec B.d1 \wedge B.d1 \prec B.d2$
$A.f1 \parallel B.f2$	car	$\neg(A.f1 \prec B.f2 \vee B.f2 \prec A.f1)$
$A.d2 \parallel B.f2$	car	$\neg(A.d2 \prec B.f2 \vee B.f2 \prec A.d2)$
$A.d2 \prec C.f1$	car	$A.d2 \prec A.f1 \wedge A.f1 \prec C.f1$

2. Montrer que la réception des flux 1 et 2 sur le site C présente une anomalie par rapport à la causalité en ce qui concerne leur début.

Réponse Le site C voit le flux 2 commencer avant le flux 1 alors que le site B a vu débiter le flux 1 avant de commencer la diffusion du flux 2.

Cette anomalie se traduit en terme de causalité par : $A.d1 \prec B.d2 \wedge C.d2 \prec C.d1$

3. Un protocole ordonné pourrait-il corriger cette anomalie sur l'ordre de démarrage des flux sur le site de réception C? Justifiez votre réponse.

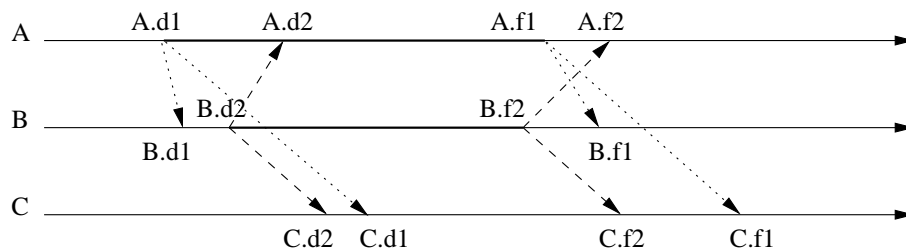


FIG. 1 – Vision événementielle des flux

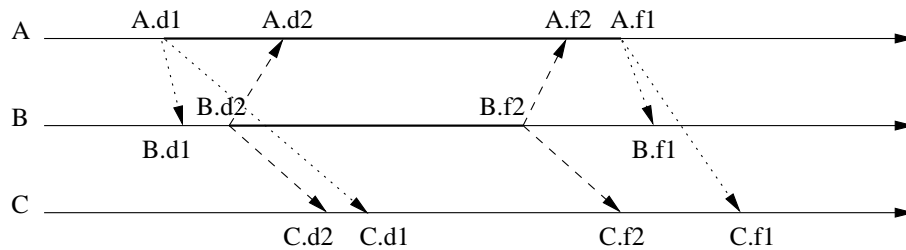
Réponse Oui, puisqu'il assurerait : $A.d1 \prec B.d2 \Rightarrow C.d1 \prec C.d2$. Les événements $C.d1$ et $C.d2$ sont alors les événements de délivrance du premier message de chaque flux au niveau applicatif.

4. De façon similaire, la figure 1 montre que les fins de flux sur les sites A et B se produisent dans un ordre différent : sur A, la fin de l'émission $A.f1$ du flux 1 précède la fin de la réception $A.f2$ du flux 2 et sur B, la fin de l'émission $B.f2$ du flux 2 précède la fin de la réception $B.f1$ du flux 1. Montrez que malgré cette inversion, aucune anomalie causale n'existe.

Réponse Il n'y a pas d'anomalie causale puisqu'il n'y a pas de relation causale entre les événements de fin d'émission de ces deux flux : $A.f1 \parallel B.f2$.

5. Dans le chronogramme étudié, la question précédente conduit à considérer que la fin de réception des flux 1 et 2 sur le site C peut se produire dans n'importe quel ordre. Modifier le chronogramme pour obtenir une situation où la réception des flux 1 et 2 devrait être forcément ordonnée comme dans le chronogramme fourni (dans lequel $C.f2 \prec C.f1$) sans provoquer d'anomalie causale.

Réponse Le chronogramme ci-dessous introduit une causalité entre la fin des émissions des 2 flux, en l'occurrence $B.f2 \prec A.f1$.



Rencontres entre agents mobiles

On considère une application répartie à base d'agents mobiles. Ces agents se rencontrent pour échanger des informations lorsqu'ils se trouvent simultanément sur le même site¹. Une telle rencontre est considérée comme atomique et réalisée toujours entre une paire d'agents².

La figure (2) illustre le déplacement et les rencontres de tels agents désignés par A, B, C, \dots

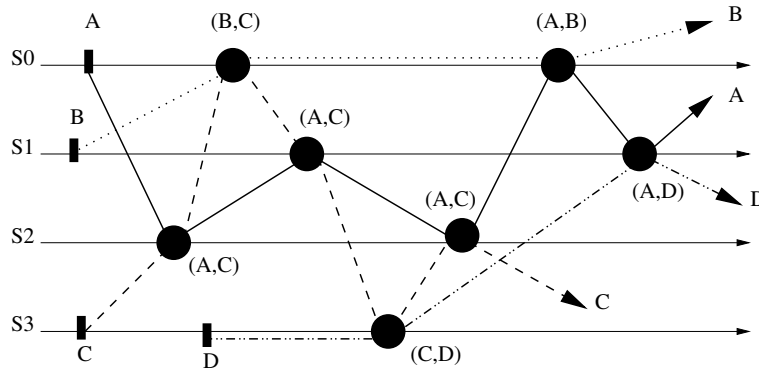


FIG. 2 – Rencontres d'agents mobiles sur des sites

¹On ne s'intéresse pas à leurs communications éventuelles à distance.

²Un agent peut par contre rencontrer successivement plusieurs agents avant de se déplacer vers un autre site.

On se propose de tracer des informations sur les rencontres qui ont lieu durant l'exécution des agents. Pour cela, on peut envisager de collecter des informations soit sur les sites de rencontre, soit dans le contexte local de chaque agent. On s'intéresse plus particulièrement à la datation des rencontres.

Trace sur les sites

On mémorise sur chaque site les dates des rencontres qui ont eu lieu. Le mécanisme de datation s'inspire du système de datation de Mattern. Une date est représentée par un vecteur de dimension N où N est le nombre de sites. Chaque site possède une horloge vectorielle H_i dotée d'une opération permettant de dater les rencontres et les agents vont transporter un vecteur courant unique représentant la date de leur dernière rencontre. On veut que la sémantique d'une date T affectée à une rencontre sur un site i_0 soit la suivante :

$$\text{invariant } (\forall i \neq i_0 : T[i] = \text{Card}(R_i)) \wedge T[i_0] = \text{Card}(R_{i_0}) + 1$$

dans lequel l'ensemble R_i contient tous les événements de rencontre ayant eu lieu sur le site i et qui précèdent causalement la rencontre datée T . À titre d'exemple, la date T affectée à la rencontre (A, D) sur le site S_1 devrait être le vecteur $T = (2, 2, 2, 1)$.

Initialement, l'horloge H_i d'un site S_i est initialisée à zéro : $\forall j :: H_i[j] = 0$. La classe **Horloge** suivante possède une seule opération **Dater** qui ne sert qu'à dater les événements de rencontre.

```
class Horloge {
    int cpt[]; int loc; // "loc" localise le site de l'horloge
    // Datation d'une rencontre
    public int[] Dater( int T1[], int T2[] ) { ... }
    public Horloge(int où, int N) {
        loc = où; cpt = new int[N] ;
        for (int i = 0; i<N; i++) cpt[i]=0 ;
    }
}
```

La sémantique de **Dater** comporte deux paramètres d'entrée, en l'occurrence les deux vecteurs horloges transportés par les agents participant à la rencontre.

Questions

- Donner sous forme d'un triplet de Hoare, la sémantique de **Dater** : $\{H_i.cpt = h\} H_i.Dater(T1, T2)\{\dots\}$ et programmer l'opération **Dater** dans la classe **Horloge**.

Réponse La date de la rencontre est un vecteur ayant pour éléments le maximum de chaque vecteur et du vecteur courant du site sauf pour l'élément i qui comptabilise la nouvelle rencontre à partir de la valeur courante de l'élément i de l'horloge locale. ³

$$\begin{aligned} & \{H_i.cpt = h\} \\ & H_i.Dater(T1, T2) \\ & \{(\forall k \neq i : H_i.cpt[k] = \max(h[k], T1[k], T2[k])) \wedge H_i.cpt[i] = h[i] + 1\} \end{aligned}$$

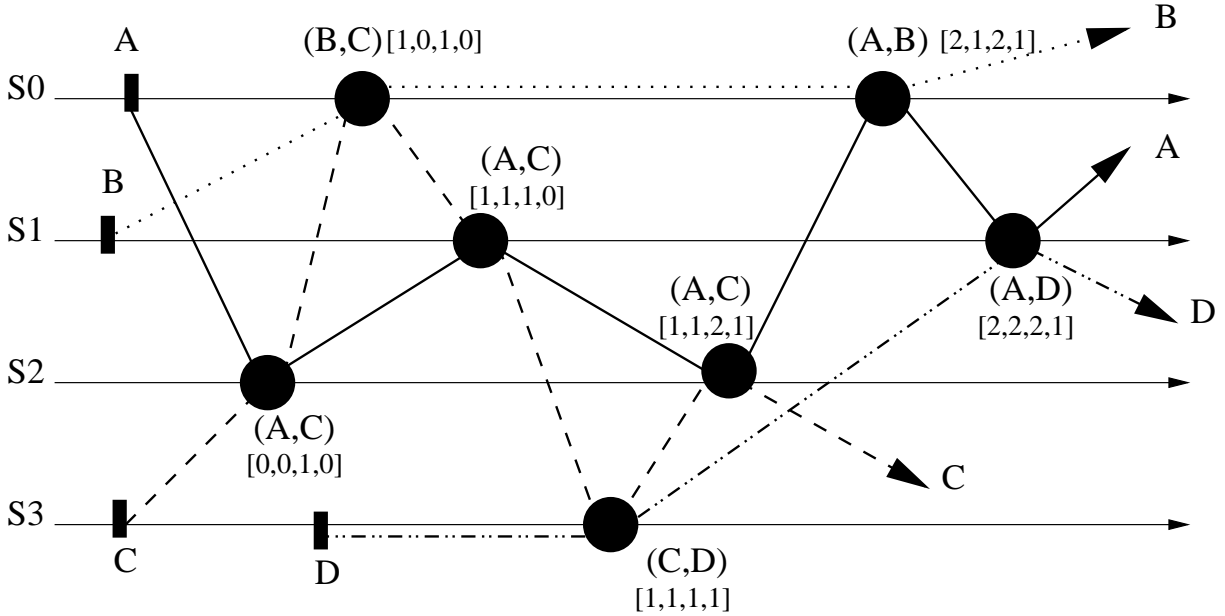
L'opération **Dater** est donc :

```
public int[] Dater( int T1[], int T2[] ) {
    for (int i = 0; i<cpt.length; i++) cpt[i]=max(cpt[i], T1[i], T2[i]) ;
    cpt[loc]++ ;
    int[] res = new int[cpt.length]; System.arraycopy(cpt, 0, res, 0, cpt.length);
    return res ;
}
```

³On a toujours pour une rencontre sur le site i : $H_i.cpt[i] \geq T1[i] \wedge H_i.cpt[i] \geq T2[i]$.

7. Décorer la figure avec les dates affectées aux rencontres en utilisant la feuille fournie en annexe.

Réponse Les dates obtenues sont :



Trace dans les agents

Cette fois-ci, chaque agent a transporte une horloge vectorielle locale H_a (horloges embarquées dans les agents). Chaque horloge locale est initialisée à zéro : $\forall a \in Agents : \forall i \in 1..M : H_a[i] = 0$. Une date est donc représentée par un vecteur de dimension M où M est le nombre d'agents⁴. Par contre, les sites n'ont plus d'horloges locales. On doit donc trouver une implantation différente de l'opération `Dater`. On suppose que chaque agent fournit en paramètre à l'opération `Dater` son indice et son vecteur horloge local. La classe `Datation` définit la nouvelle opération `Dater` :

```
class Datation {
    // Datation d'une rencontre
    static public int[] Dater(int a, int Ha[], int b, int Hb[]) { ... }
}
```

On veut que la sémantique d'une date T affectée à une rencontre entre un agent a et un agent b soit la suivante :

$$\text{invariant } (\forall k \neq a, b : T[k] = \text{Card}(R_k)) \wedge T[a] = \text{Card}(R_a) + 1 \wedge T[b] = \text{Card}(R_b) + 1$$

dans lequel un ensemble R_x contient tous les événements de rencontre auxquels l'agent x a participé et qui précèdent causalement la rencontre datée T . À titre d'exemple, la date T affectée à la rencontre (A, C) sur le site S_1 devrait être le vecteur $T = (2, 1, 3, 0)$.

⁴On suppose que le nombre maximum d'agents est connu pour simplifier et conserver des vecteurs.

Questions

8. En désignant, dans la postcondition, le vecteur date résultat par T_r , donner sous forme d'un triplet de Hoare, la sémantique de **Dater** :

$$\{H_a = h_a \wedge H_b = h_b\} \text{ Datation.Dater}(a, Ha, b, Hb)\{\dots\}$$

et programmer l'opération **Dater** dans la classe **Datation**.

Réponse La solution consiste à prendre une nouvelle fois le maximum des éléments de chaque vecteur et à incrémenter de un les deux éléments concernant les agents de la rencontre :

$$\begin{aligned} & \{H_a = h_a \wedge H_b = h_b\} \\ & \text{Datation.Dater}(a, Ha, b, Hb) \\ & \{\forall i \neq a, b : T_r[i] = \max(h_a[i], h_b[i]) \wedge T_r[a] = h_a[a] + 1 \wedge T_r[b] = h_b[b] + 1\} \end{aligned}$$

L'opération **Dater** correspondante est la suivante⁵ :

```
static public int[] Dater(int a, int Ha[], int b, int Hb[] ) {
    int[] res = new int[Ha.length];
    for (int i = 0; i<Ha.length; i++) res[i]=max(Ha[i],Hb[i]) ;
    Ha[a]++ ; Hb[b]++ ;
    return res ;
}
```

9. Montrer que l'on a l'invariant :

$$\mathbf{invariant} \forall T : \left(\sum_{x=1}^{x=M} T[x] \right) / 2 = nr_T + 1$$

dans lequel le terme nr_T est égal au nombre total de rencontres (quel que soit les participants) qui précèdent causalement la rencontre datée par T .

Pour cela, montrer que si le prédicat invariant est vrai avant l'exécution d'une opération **Dater**, il est encore vrai près l'exécution de l'opération.

Réponse D'après la sémantique de l'opération **Dater**, on a, puisque les éléments $H_a[a]$ et $H_b[b]$ sont maximaux :

$$\sum_{x=1}^{x=M} T[x] = \sum_{x=1}^{x=M} \max(H_a[x], H_b[x]) + 2$$

Il reste donc à prouver que :

$$2 * nr_T = \sum_{x=1}^{x=M} \max(H_a[x], H_b[x])$$

Pour les éléments $H_a[a]$ et $H_b[b]$, nous avons vu que :

$$\max(H_a[a], H_b[a]) = H_a[a] \text{ et } \max(H_b[b], H_a[b]) = H_b[b]$$

Ces compteurs comptabilisent donc le nombre de rencontres de l'agent A (resp. B) avec n'importe quel autre agent, rencontres causalement antérieures à la rencontre courante.

Les éléments $\max(H_a[x], H_b[x])$, $x \neq a, b$ enregistrent les rencontres de l'agent X perçues par les agents A ou B. Supposons que l'agent A a enregistré $H_a[x]$ rencontres de X et l'agent B a enregistré $H_b[x]$, avec par exemple $H_a[x] < H_b[x]$.

⁵On remarque que, comme pour l'opération **Dater** précédente, on a : $\forall k : H_a[a] \geq H_k[a] \wedge H_b[b] \geq H_k[b]$

Puisque **TOUTES** les rencontres successives de l'agent X avec les autres sont causalement liées, cela signifie simplement que l'agent B a rencontré X **après** l'agent A et que les rencontres de X avec A sont donc toutes déjà comptabilisées. En tout état de cause, il suffit donc de prendre le maximum des deux éléments pour capter exactement le nombre de rencontres de l'agent X qui précèdent cette rencontre entre A et B.

10. Décorer la figure avec les dates affectées aux rencontres en utilisant la feuille fournie en annexe.

Réponse Les dates obtenues sont :

