Systèmes et Applications Répartis

année1996-1997

Ordres, état global, horloges, synchronisation dans les systèmes répartis

C Kaiser

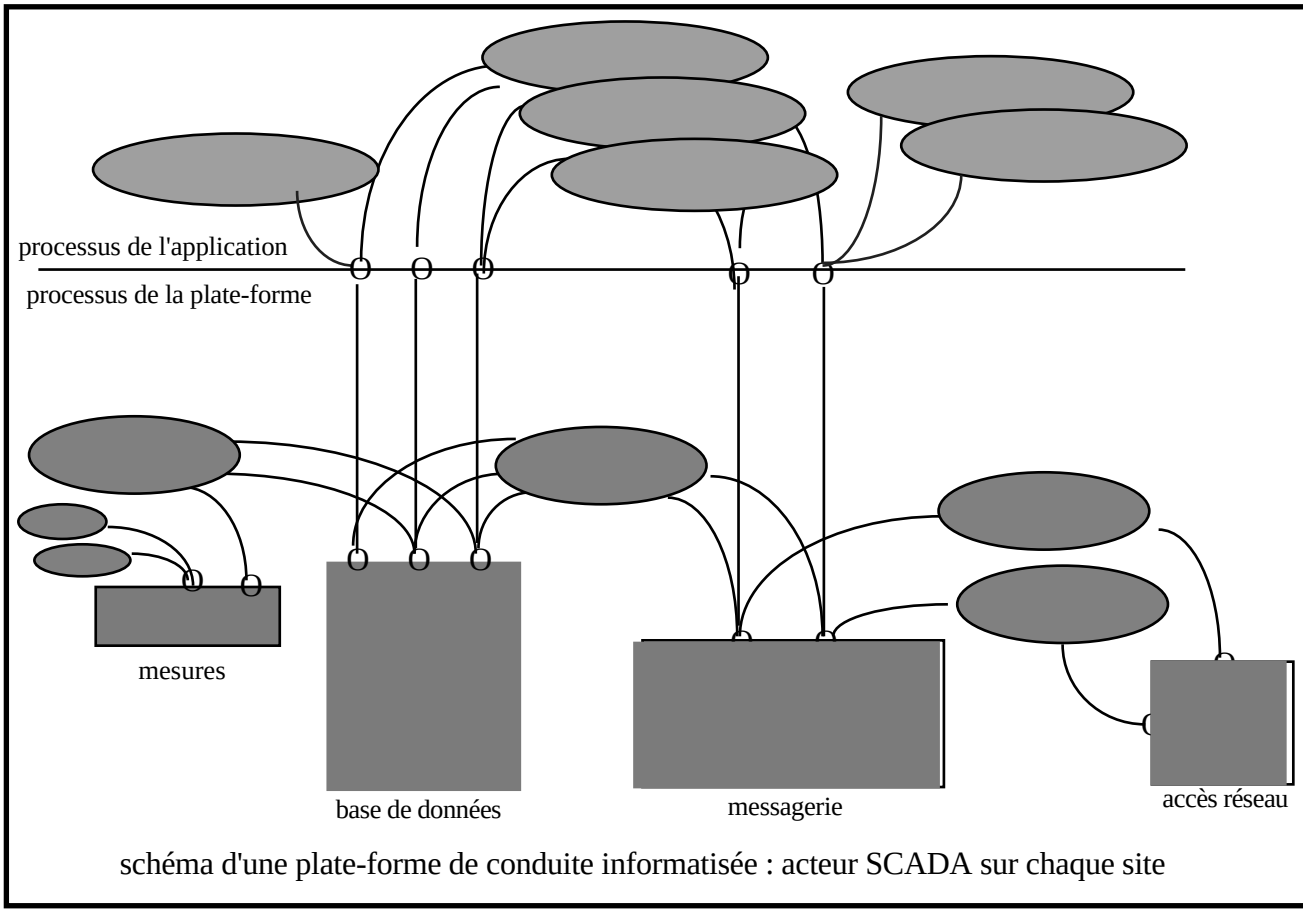
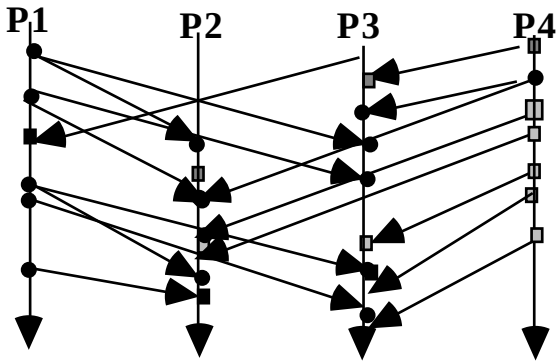


schéma d'une plate-forme de conduite informatisée : acteur SCADA sur chaque site



trace de messages entre 4 acteurs

SYSTEMES ET APPLICATIONS REPARTIS

Ordres, état global, horloges, synchronisation dans les systèmes répartis

Besoin des applications réparties
Réel et modèles de communication élémentaire
Dépendance causale
Modèles de diffusion fiable et communication de groupe
Propriétés d'ordre dans les groupes
Désordre naturel des communications
Etat global d'un système réparti
Passé et coupures cohérentes
Détermination d'un état global cohérent
Datation causale et horloges vectorielles
Horloges vectorielles et coupures cohérentes
Diffusion fiable avec ordre causal
Ordre total par horloges logiques
Exclusion mutuelle répartie, avec horloge logique

Bibliographie :

R.Balter, J.P.Banâtre, S.Krakowiak, éditeurs. Construction des systèmes d'exploitation répartis. Collection didactique INRIA 1991 (350 pages)

G.Coulouris, J.Dollimore, T.Kindberg. *Distributed Sytems (2nd edition)*. Addison Wesley 1995 (601 pages)

S.Mullender. ***Distributed Sytems (2nd ed.).*** **Addison Wesley 1994 (644 p.)**

A.Tanenbaum. ***Distributed Operating Sytems.*** **Prentice Hall 1995 (614 p.)**

BESOINS DES APPLICATIONS REPARTIES

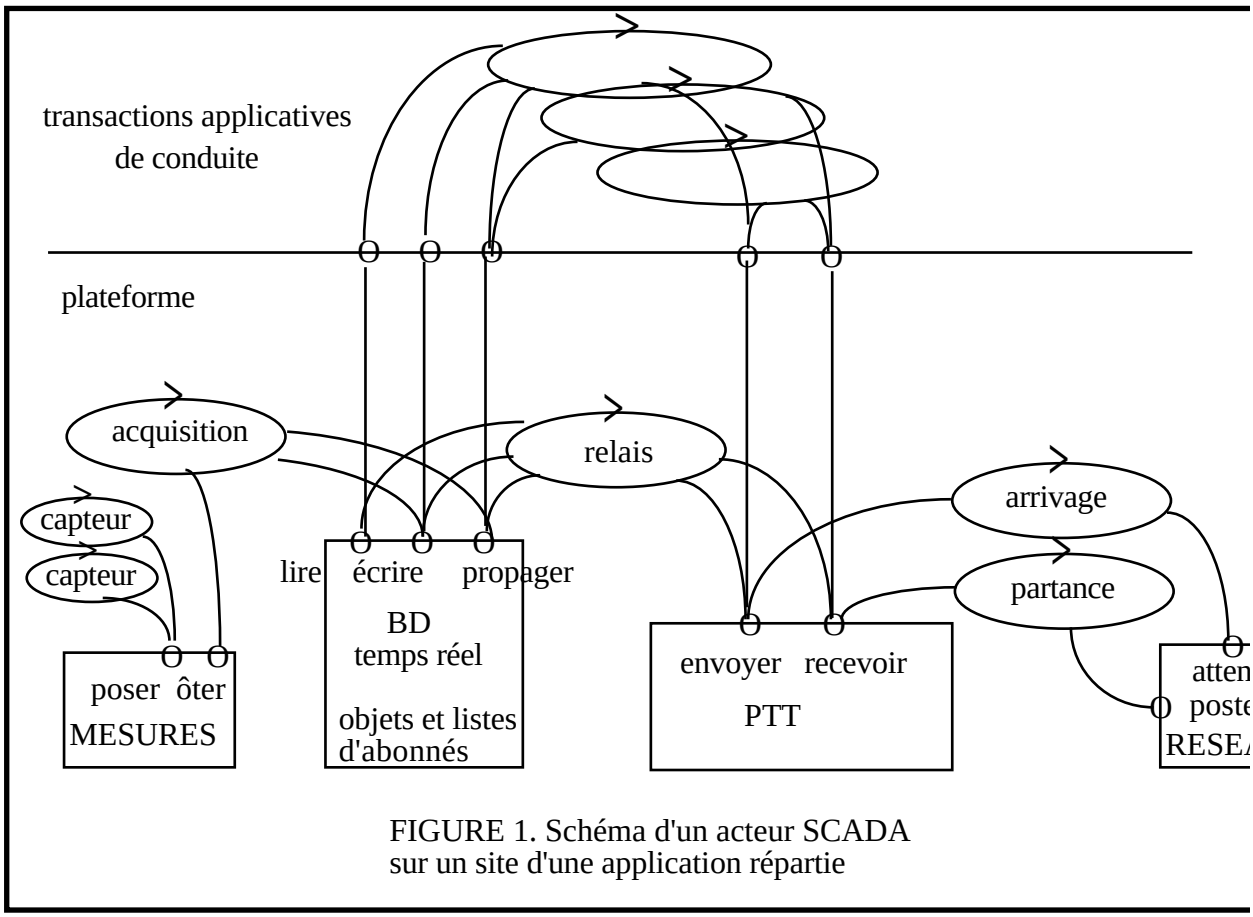


FIGURE 1. Schéma d'un acteur SCADA
sur un site d'une application répartie

APPLICATION REPARTIE = ENSEMBLE DE SITES

CHAQUE SITE SCADA COMPREND

UNE PLATE-FORME AVEC:

des modules d'acquisition : captures concurrentes, acquisition synthétique

des bases de données temps réel : lecteurs rédacteurs en mémoire centrale

une messagerie : producteurs consommateurs

un module réseau : communication de messages intersites

des processus de service

UNE COUCHE APPLICATIVE AVEC :

des processus appelés transactions applicatives

BESOINS DES APPLICATIONS REPARTIES

REPARTITION SIMPLE (CLIENT SERVEUR)

accès local ou distant aux BD TR, chaque BD cohérente individuellement

abonnement à BD primaire: copies secondaires pour lecture, sur autres sites

transactions avec accès à une seule BD primaire à la fois

messagerie entre les processus de divers sites

REPARTITION PLUS COMPLEXE

Transactions avec accès emboîtés à plusieurs BD :
problèmes de cohérence globale et interblocage

Copies multiples d'une même BD avec écritures sur chaque copie : cohérence faible ou forte (problème des caches multiples)

Ensemble de transactions coopérantes. Problèmes de synchronisation:
démarrage dans un état cohérent
coordination par un site fixe, mais si absent (panne, maintenance) alors élection d'un nouveau coordinateur
terminaison de la coopération
mise au point répartie

points de reprise cohérents

diffusion d'information dans un groupe

Diffusion fiable et ordonnée à un groupe de processus sur des sites divers

BESOINS DES APPLICATIONS REPARTIES

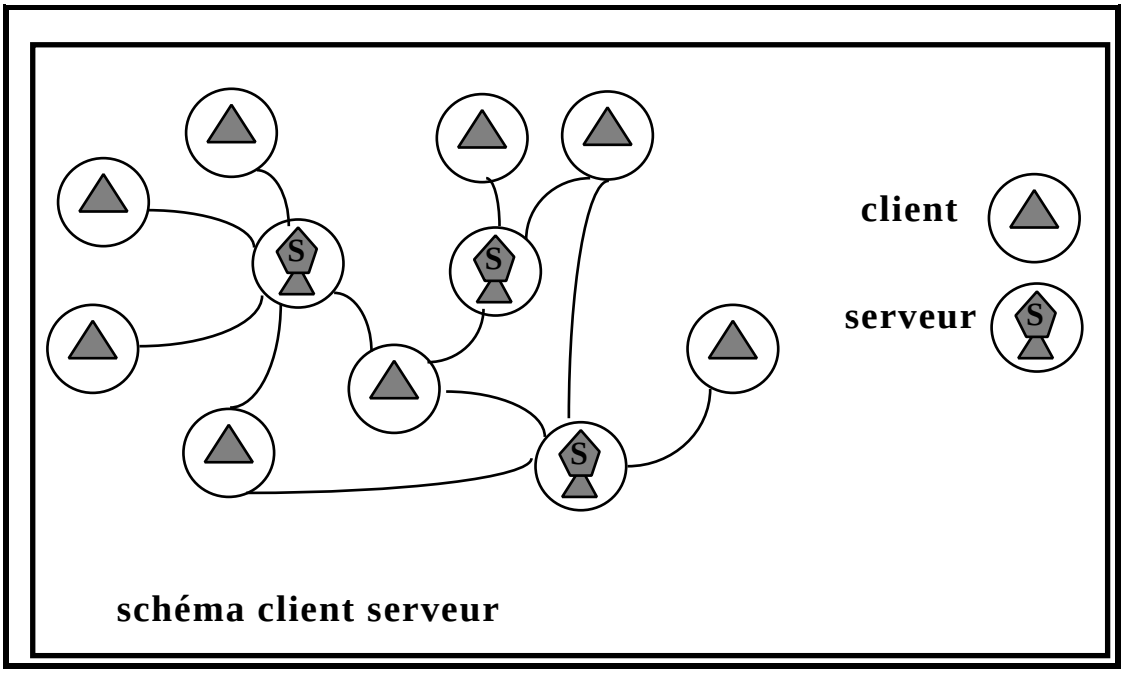


schéma client serveur

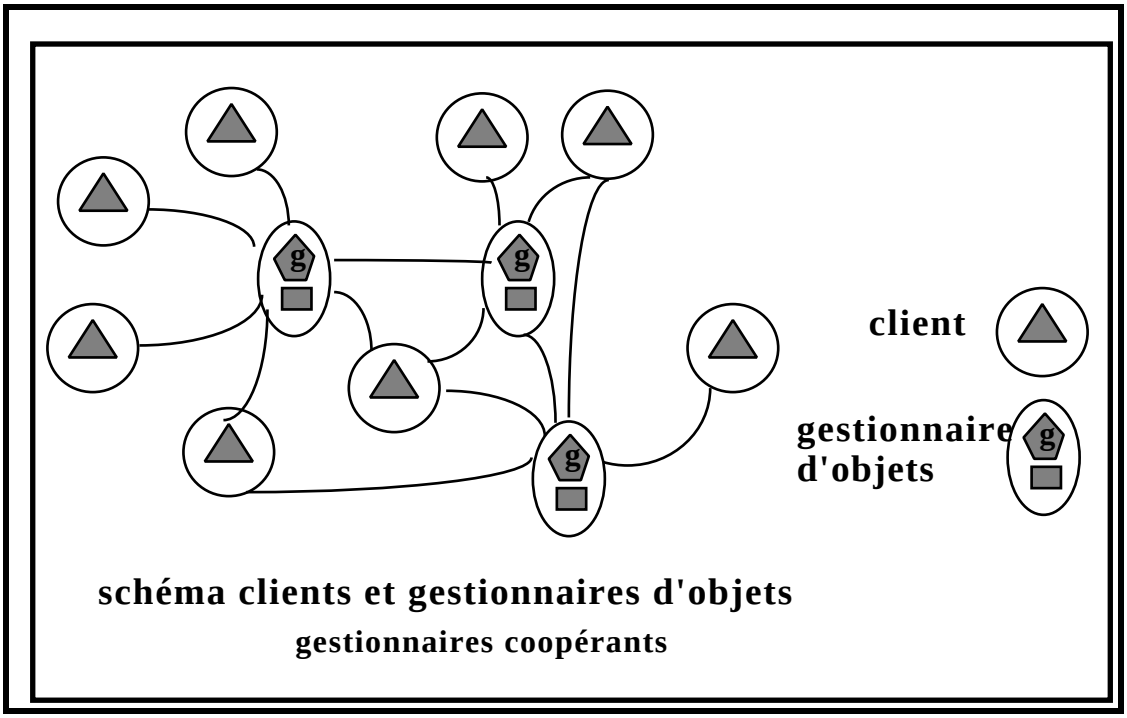
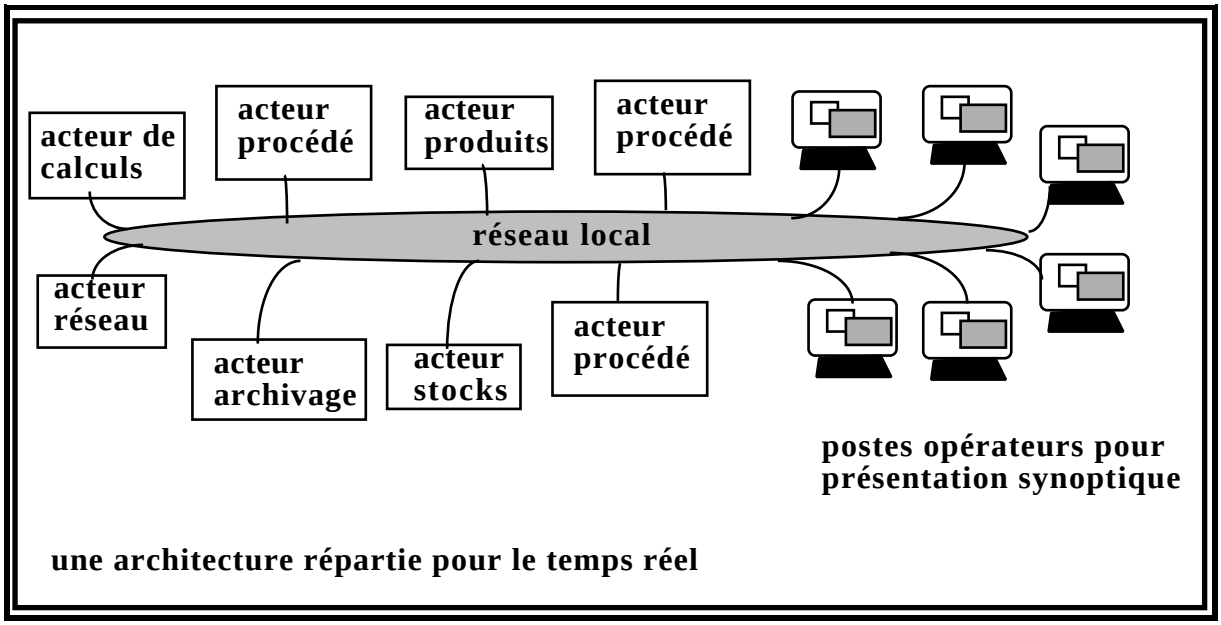


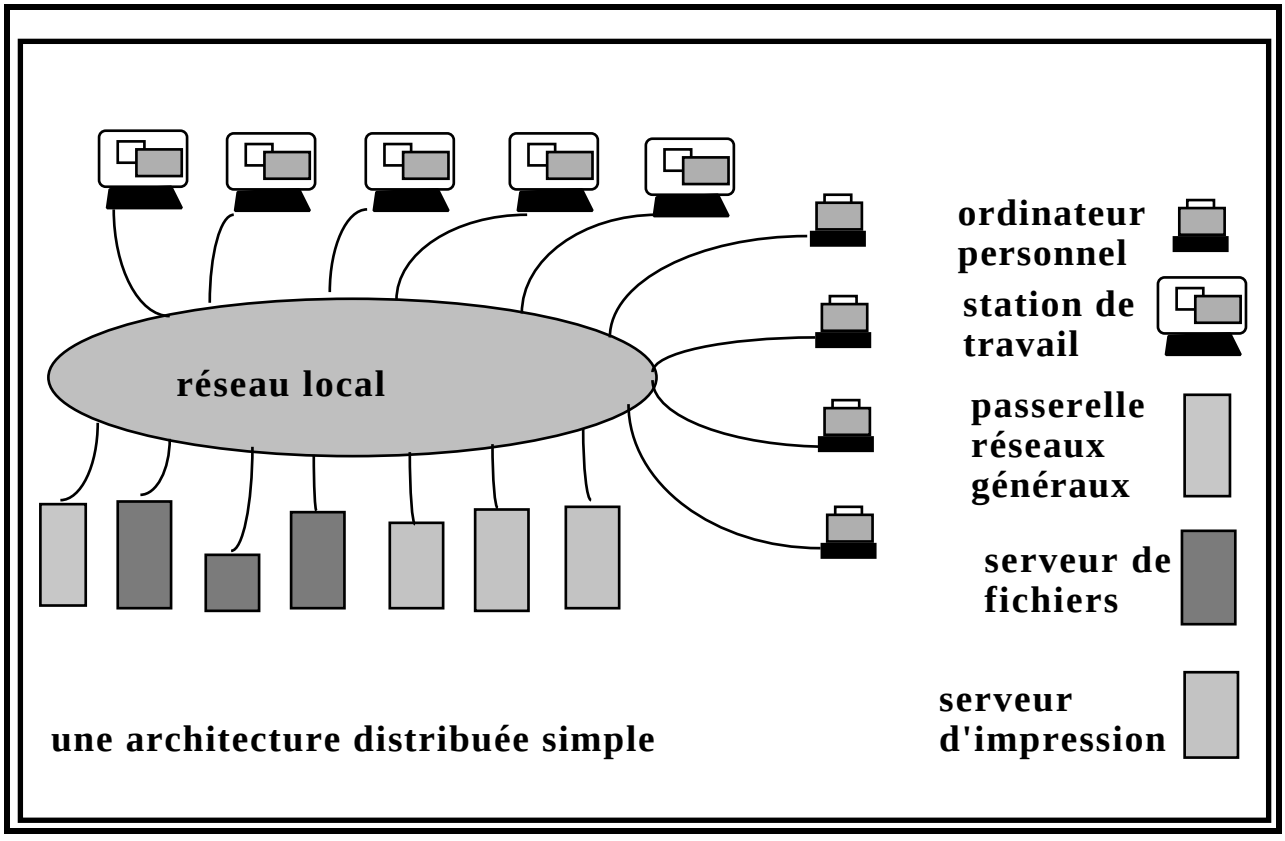
schéma clients et gestionnaires d'objets

gestionnaires coopérants

EXEMPLES D'ARCHITECTURE PHYSIQUE
(ARCHITECTURE ORGANIQUE)



une architecture répartie pour le temps réel



une architecture distribuée simple

LE REEL DES COMMUNICATIONS ELEMENTAIRES

POINT A POINT MODE MESSAGE

message d'un émetteur vers un récepteur sur un canal

perte de message, absence de récepteur (panne, maintenance,...)

contrôle d'erreur par acquit et délais de garde, mais duplication possible

numérotation des messages successifs

réception désordonnée d'une suite de messages

contrôle de flux pour asservir les vitesses de l'émetteur et du récepteur

(producteur-consommateur)

incertitude sur l'état du canal, de l'émetteur et du récepteur

durées de transfert variable, asynchrone (mais il existe des bus synchrones)

POINT A POINT MODE CLIENT SERVEUR

l'émetteur est le client et le récepteur est le serveur

message requête d'un client vers un serveur sur un canal

message réponse du serveur au client

le client n'envoie pas d'autre requête avant d'avoir reçu la réponse

mêmes problèmes d'erreurs, d'incertitudes et de variabilité des délais

DIFFUSION A UN GROUPE

Un émetteur et N récepteurs sur le canal

exemples : Ethernet, Token ring

perte de message pour R récepteurs avec 0 <= R <= N

d'une émission à l'autre perte sur des récepteurs différents

pas le même ordre sur tous les sites pour une suite de réceptions

pas de perception unique de l'ordre d'émission si émetteurs différents

durées de transfert variable

incertitudes sur l'état du canal, de l'émetteur et des récepteurs

incertitude sur la composition du groupe

LES MODELES DE LA COMMUNICATION ELEMENTAIRE

MODELE DE COMMUNICATION SYNCHRONE FIABLE

Tout message arrive avant le délai d_{max}, qu'il soit point à point ou diffusé.

réseau isotrope : même délai d_{max} pour tous les canaux

communication fiable : pas de perte de message, pas de panne de site

réseau connexe : tout site peut communiquer avec tous les autres

les horloges des sites sont aussi synchronisées
(leur écart est borné par dh_{max})

ELECTION EN MODELE SYNCHRONE FIABLE

Les sites S1, S2,..., Si, ..., SN doivent élire un site coordonnateur

Chaque site Si a un identificateur unique uid(i)

et peut diffuser un message <élection, i, uid(i))>

Tous les sites démarrent l'élection en même temps

(à dates fixes, par exemple)

Soit $T = d_{max} + dh_{max}$,

Chaque site Si

(i) attend de recevoir un message d'élection

(ii) s'il n'a rien reçu au bout de T * uid(i) secondes, mesurées sur son horloge, il diffuse son message d'élection.

Résultat : le premier site qui diffuse son message est l'élu

Cet algorithme synchrone élit le site de plus petit uid

commentaire : simple n'est-il pas? mais l'hypothèse synchrone est restrictive (pas de pannes, horloges communes) la "nature" est asynchrone.

MODELE DE COMMUNICATION ASYNCHRONE FIABLE

réseau connexe : tout site peut communiquer avec tous les autres
durées de transfert variables
communication fiable : pas de perte de message, pas de panne de site

PROPRIÉTÉ DE CAUSALITE ELEMENTAIRE

Par la nature physique de la communication, l'émission d'un message sur un site précède nécessairement la réception du message sur le site destinataire. Toute réception d'un message est causée par une émission antérieure.

Cette relation causale permet d'établir, dans un système réparti, une relation d'ordre partiel entre l'événement d'émission d'un message sur un site et l'événement de réception du message sur un autre site destinataire. cette relation se note (on lit précède) :

$$\forall m, \text{EMISSION}(m) \rightarrow \text{RECEPTION}(m)$$

HYPOTHESES PROPRES A UN CANAL C

DEFINITION : m1 double m2 dans le canal C si et seulement si

$$\text{EMISSION}(m2) \rightarrow \text{EMISSION}(m1)$$
$$\text{et RECEPTION}(m1) \rightarrow \text{RECEPTION}(m2)$$

TYPES DE MESSAGES

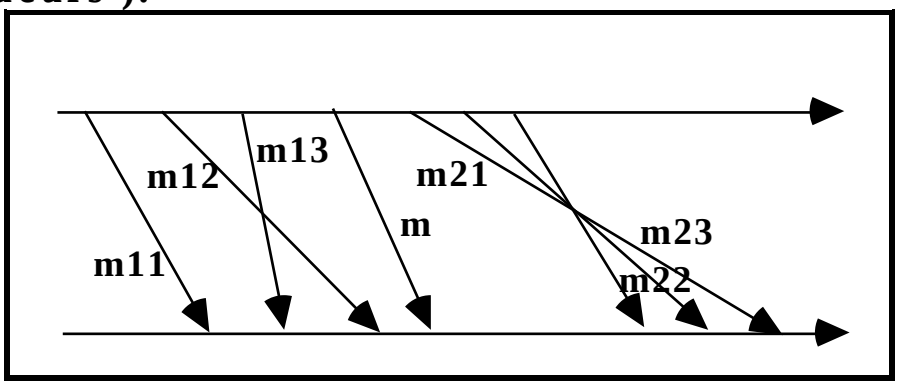
1• un message m de type marqueur ne peut ni doubler ni être doublé sur C

$\forall m1, \forall m2,$

$\text{EMISSION}(m1) \rightarrow \text{EMISSION}(m) \Rightarrow \text{RECEPTION}(m1) \rightarrow \text{RECEPTION}(m)$

$\text{EMISSION}(m) \rightarrow \text{EMISSION}(m2) \Rightarrow \text{RECEPTION}(m) \rightarrow \text{RECEPTION}(m2)$

2• un message m de type ordinaire n'impose pas de condition de réception, mais respecte celles des autres (il ne peut doubler les marqueurs et ne peut être doublé par les marqueurs).



tout marqueur m sépare les messages du canal en deux sous-ensembles

$<_m = \{m1 \mid \text{EMISSION}(m1) \rightarrow \text{EMISSION}(m)\}$

et m est un marqueur ⇒ RECEPTION(m1) → RECEPTION(m)

$>_m$ = {m2 | EMISSION(m) → EMISSION(m2) }

et m est un marqueur ⇒ RECEPTION(m) → RECEPTION(m2)

TYPES DE COMPORTEMENT D'UN CANAL

1• le moins contraint : tous les messages sont ordinaires

2• le plus contraint : tous les messages sont des marqueurs (canal FIFO)

RELATION DE CAUSALITÉ ENTRE DES ÉVÉNEMENTS RÉPARTIS

a) A "précède causalement" A' si A et A' sont des événements qui ont été générés dans cet ordre sur le même site S (ordre local) : A → A'

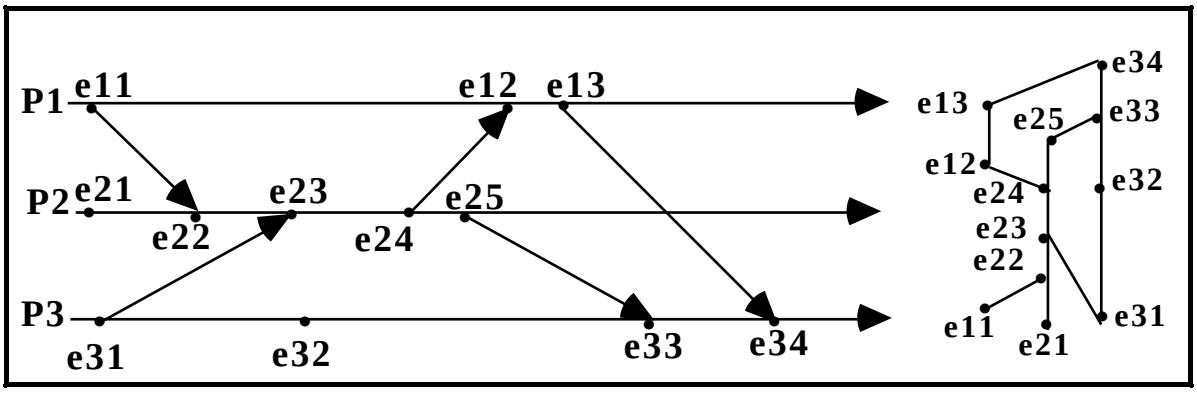
b) A "précède causalement" A' si A est l'événement d'émission d'un message M par le site P et que A' est l'événement de réception du message M sur Q : A → A' (ordre causal pour chaque message)

La relation de causalité dans un système réparti est la fermeture transitive des deux relations précédentes.

si A → B et B → C alors A → C

La dépendance causale est potentielle (si A → B, A peut avoir influencé B).

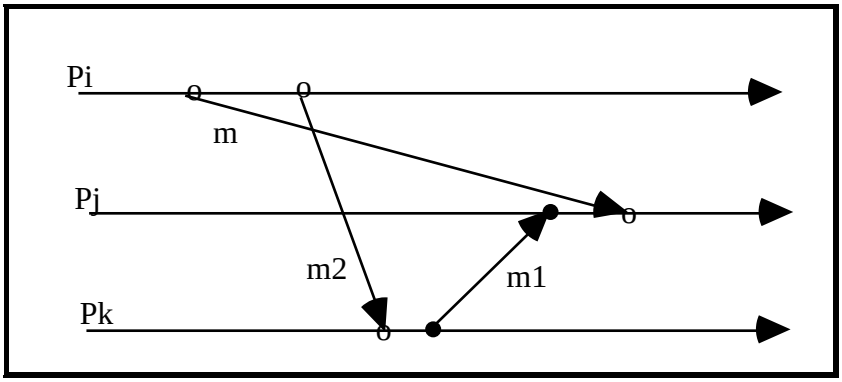
La dépendance causale est un ordre partiel



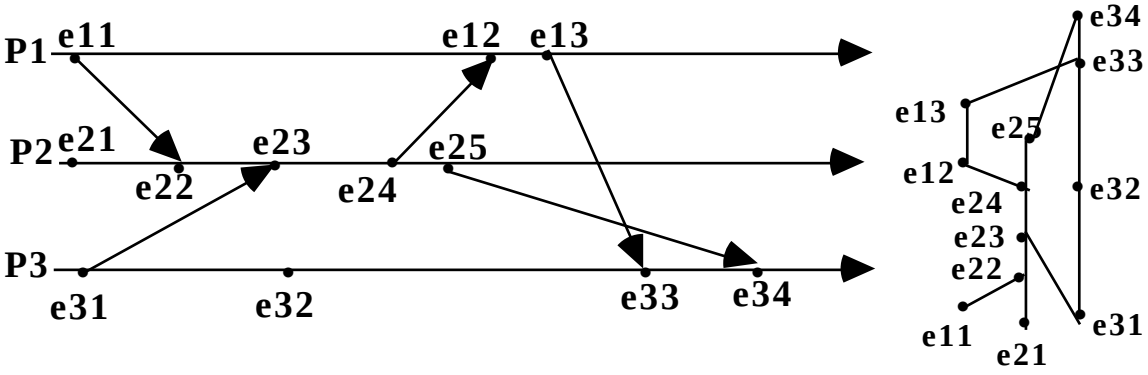
PROPRIÉTÉ D'ORDRE CAUSAL SUR LES MESSAGES

La réception des messages respecte la dépendance causale lorsque :

$\forall Pi, \forall Pj, \forall Pk, \forall m$ émis sur Cij, $\forall m1$ émis sur Ckj, EMISSION $_i(m) \rightarrow$ EMISSION $_k(m1) \Rightarrow$ RECEPTION $_j(m) \rightarrow$ RECEPTION $_j(m1)$



Premier exemple : la réception ne respecte pas la dépendance causale



Deuxième exemple : la réception respecte la dépendance causale
car les émissions e13 et e25 ne sont pas en dépendance causale

MODELES DE DIFFUSION FIABLE ET COMMUNICATION DE GROUPE

Un message émis doit être reçu par n destinataires.

MODELES DE COMMUNICATION

Communication inclusive ou non (l'émetteur reçoit le même message - le sien enrichi par le réseau- que les récepteurs)

Communication interne ou externe (l'émetteur, client du groupe, n'appartient pas au groupe)

CLASSIFICATION SELON LES EMETTEURS ET LES RECEPTEURS (classification OSI)

MODE CENTRALISE ("multicast")

Un seul émetteur (toujours le même) et n récepteurs.

MODE CENTRALISE A CENTRE MOBILE

L'émetteur est unique par périodes.

MODE MULTI-CENTRE

N processus émetteurs peuvent à tout instant effectuer une diffusion vers P récepteurs.

MODE DECENTRALISE OU MODE CONVERSATION

Un ensemble de N sites peuvent être à tout instant émetteurs et sont tous destinataires des messages.

PROPRIETES D'ORDRE DANS LES GROUPES

DIFFUSION RESPECTANT L'ORDRE LOCAL

Pour deux diffusions successives du même processus, les messages sont délivrés dans le même ordre sur chaque site distant

DIFFUSION RESPECTANT L'ORDRE CAUSAL

diffusion + causalité (Birman et Joseph 1987)

Relation de causalité entre les événements répartis (Lamport 78)

+

Relation de causalité entre les messages, généralisée à la diffusion

Toute suite de diffusions de messages en relation de causalité implique la délivrance des messages sur tous les sites destinataires dans la même relation de causalité

∀Pi, ∀Pk, ∀m

DIFFUSION$_i$(m) précède causalement DIFFUSION$_k$(m1)

⇒ RECEPTION$_j$(m) précède RECEPTION$_j$(m1) pour tout Pj.

Exemple d'utilisation de la diffusion causale

A diffuse un courrier électronique à B, C1 et C2 qui contient: 'je demande à B de nous diffuser du travail par courrier électronique'.

Pour respecter l'ordre causal, C1 et C2 ne doivent pas recevoir le courrier de B avant celui de A.

DIFFUSION RESPECTANT UN ORDRE TOTAL SIMPLE

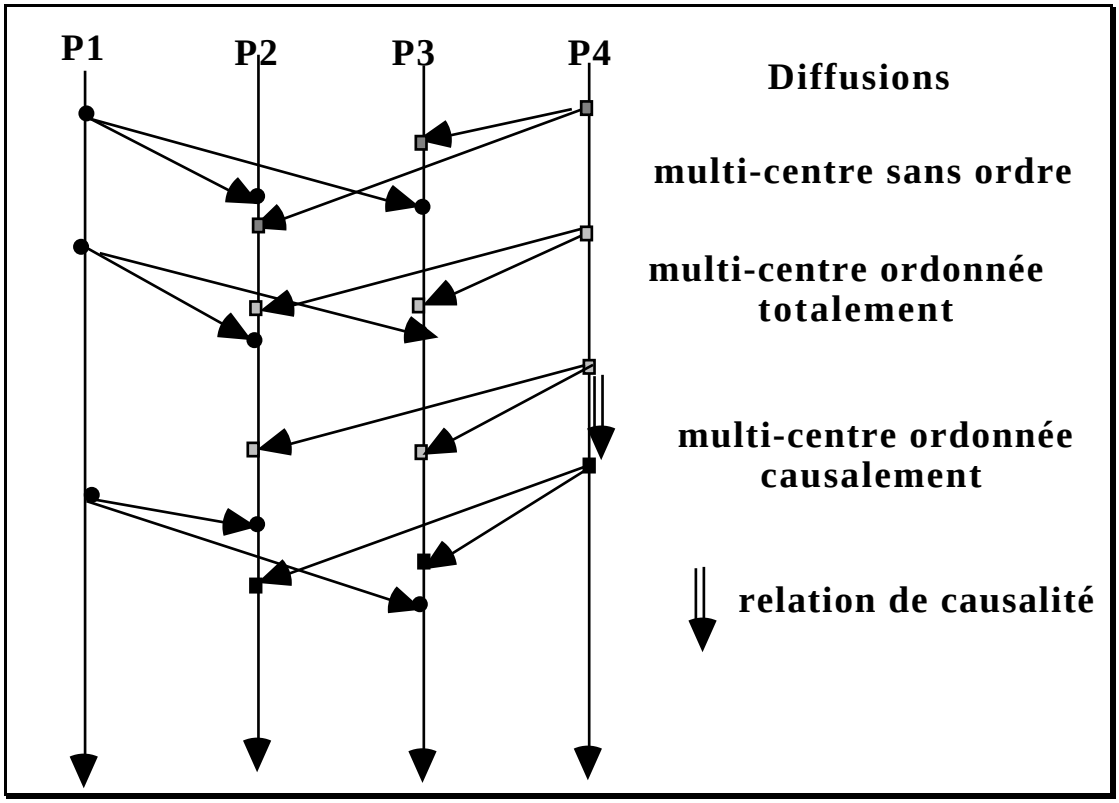
Si plusieurs diffusions ont lieu concurremment de différents processus vers le même groupe de diffusion, alors tous les messages sont délivrés aux applications réceptrices dans le même ordre sur tous les récepteurs.

Exemple d'utilisation : copies multiples.

Le fait que toutes les opérations de modifications d'un ensemble de données en copies multiples soient effectuées dans le même ordre sur toutes les copies suffit à assurer le maintien de la cohérence (faible) des copies.

DIFFUSION RESPECTANT L'ORDRE TOTAL CAUSAL

L'ordre total respecte aussi la relation d'ordre causal entre messages



DESORDRE NATUREL DES COMMUNICATIONS

délais de transmission variables, dispersés, non bornés

pertes de messages, pannes de sites

Exemple : Base de donnée répliquée avec même valeur initiale

T1 : début; traitement local; envoi M1;

T2 : début; traitement local; envoi M2; fin;

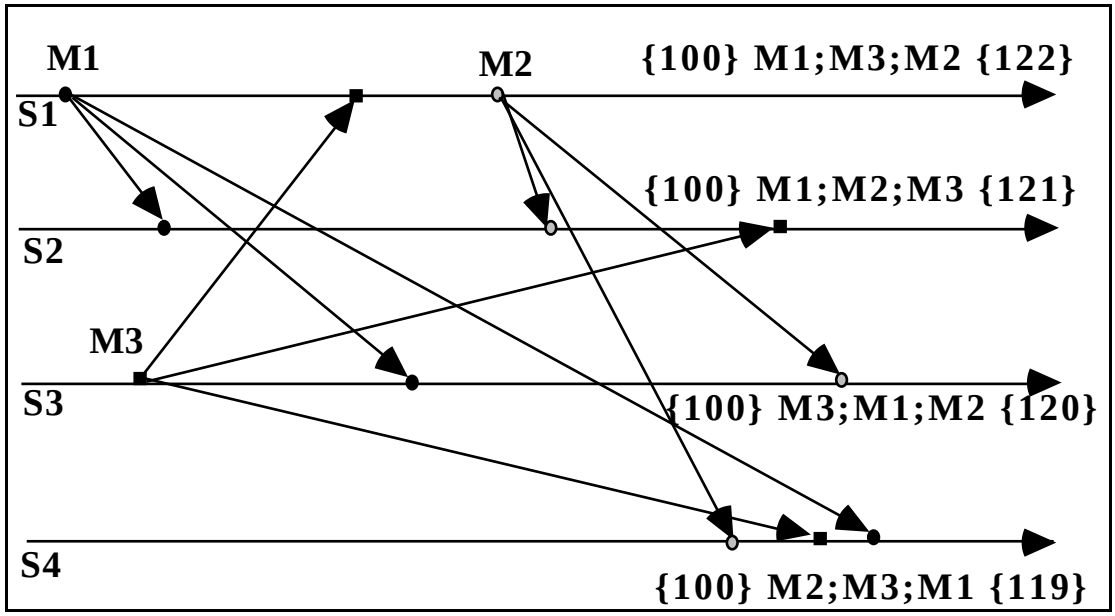
T3 : début; traitement local; envoi M3; fin;

M1, M2, M3 : messages de mise à jour diffusés à toutes les copies.

M1 : j'ai ajouté 20

M2 : j'ai retranché 10

M3 : j'ai augmenté la valeur de 10%



Quelles sont les valeurs des répliques après exécution de T1 et T3 ?

Que faire si perte de message ou panne de site ?

TRAITEMENT EN CLIENT SERVEUR

Site S2 : serveur de base de donnée, répliques comme bases secondaires

Il faut appeler systématiquement le serveur pour tout accès à la base

T1 : début; traitement local; envoi R1; attendre M1;

T2 : début; traitement local; envoi R2; attendre M2; fin;

T3 : début; traitement local; envoi R3; attendre M3; fin;

R1, R2, R3 : requêtes envoyées au serveur

R1 : ajouter 20

R2 : retrancher 10

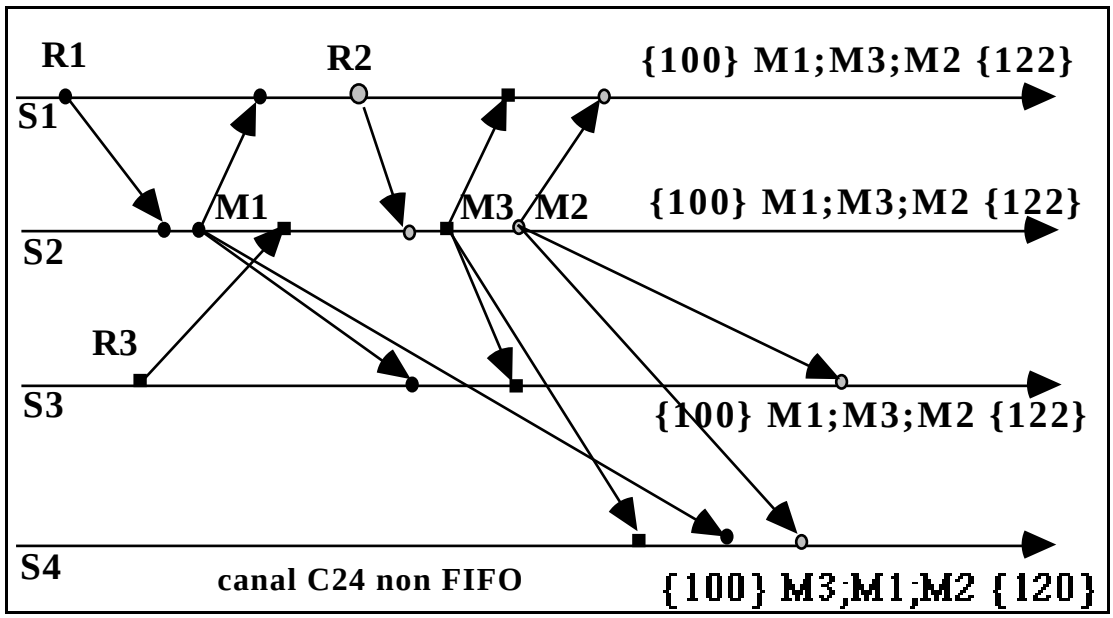
R3 : augmenter la valeur de 10%

M1, M2, M3 : messages de mise à jour diffusés à toutes les copies

M1 : j'ai ajouté 20

M2 : j'ai retranché 10

M3 : j'ai augmenté la valeur de 10%



transferts nécessairement à l'ancienneté (FIFO). Pas de panne de S2.

toutes écritures par S2, lectures concurrentes en cohérence faible

TRAITEMENT AVEC ANNEAU VIRTUEL ET JETON

Base de donnée répliquée. Anneau virtuel : S1 S2 S3 S4 S1 S2 S3...

Il faut modifier les transactions pour attendre systématiquement le jeton J

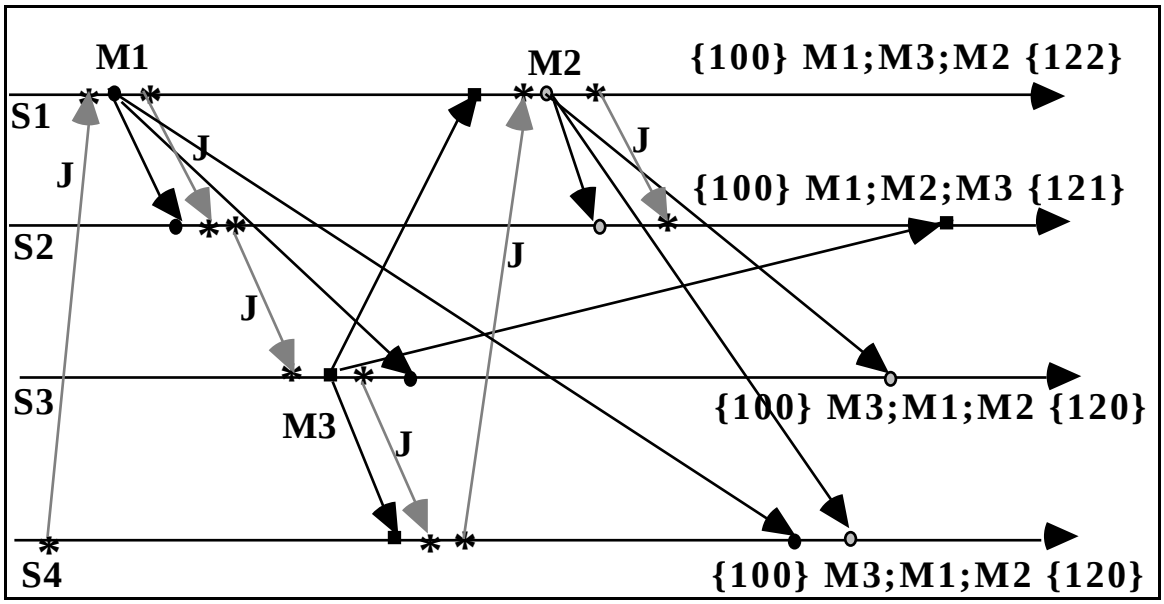
T1 : début; traitement local; attendre J; diffuser M1; envoi J;

T2 : début; traitement local R2; attendre J; diffuser M2; envoi J; fin;

T3 : début; traitement local; attendre J; diffuser M3; envoi J; fin;

attendre J : attendre le jeton du site prédécesseur sur l'anneau virtuel
envoi J : envoyer le jeton au site successeur sur l'anneau virtuel
M1, M2, M3 : messages de mise à jour diffusés à toutes les copies.

M1 : j'ai ajouté 20
M2 : j'ai retranché 10
M3 : j'ai augmenté la valeur de 10%



le passage du jeton induit une causalité entre les émissions en diffusion

émission(M1) → émission(M3) → émission(M2)

l'anneau virtuel et les canaux FIFO ne suffisent pas, il faut en plus :

soit la diffusion causale : réception(M1) → réception(M3) → réception(M2)

soit la diffusion totalement ordonnée (le jeton numérote les émissions)

ETAT GLOBAL D'UN SYSTEME REPARTI

• ETAT LOCAL ELi D'UN SITE Si

état initial et séquence d'événements locaux sur Si

• ETAT LOCAL ECij D'UN CANAL Cij

ensemble des messages en transit sur le canal Cij

émis par Si et non encore reçus par Sj

• EVENEMENTS (ATOMIQUES) FAISANT EVOLUER LE SYSTEME

{ELi} événement interne sur Si {EL'i}

{ELi, ECij} émission de m par Si sur Cij {EL'i, EC'ij = ECij ∪ {m}}

{ELi, ECki} réception de m par Si sur Cki {EL'i, EC'ki = ECij - {m}}

• ETAT GLOBAL S = { pour tout i, j, ∪ ELi, ∪ ECij}

• DIFFICULTES

ELi n'est immédiatement observable que sur Si

ECij n'est jamais directement observable, ni sur Si, ni sur Sj

• OBJECTIF : définir un état global déterminable <u>localement</u> par tout site

REMARQUE : toute définition d'état doit respecter la dépendance causale

nota : on dit indifféremment site ou processus (dans ce cas il y a un processus par site)

PASSÉ D'UN ÉVÉNEMENT

Le passé (ou historique) d'un événement e, c'est par définition :

hist(e) = e ∪ ensemble des événements e' tels que e' → e

Seul le passé de e peut avoir influencé e

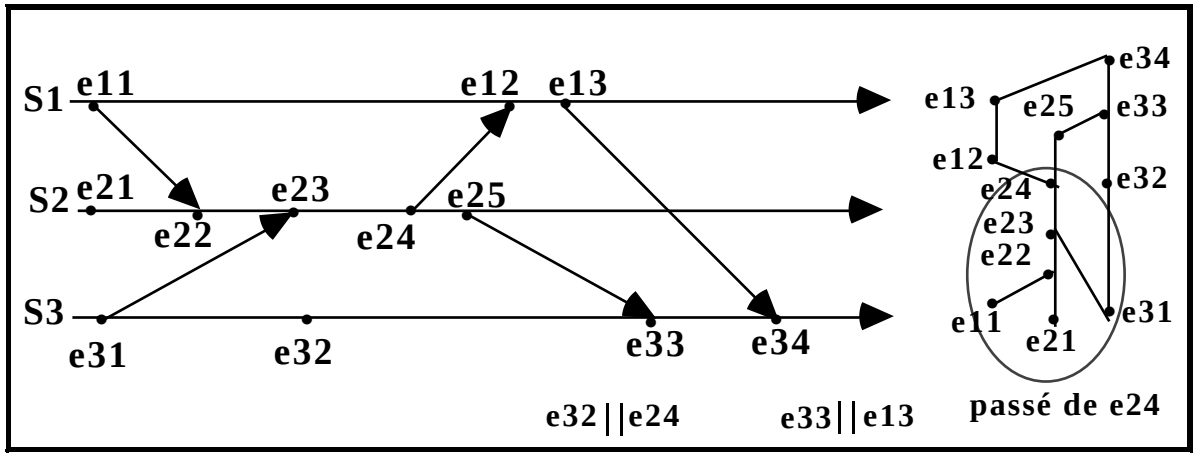
Chaîne causale : $e_0 \ldots e_n$ tels que $e_{i-1} \rightarrow e_i$ pour tout i ∈ (1...n)

Evénements concurrents (ou encore causalement indépendants)

a || b ⇔ non (a → b) et non (b → a)

aucun des 2 événements n'appartient au passé de l'autre

aucun des 2 événements ne peut influencer l'autre



Applications

définition de la cohérence d'un état, d'une observation
mise au point répartie
mesure du parallélisme

COUPURES

soit E un ensemble d'événements constituant une application répartie

Coupure = sous-ensemble fini C de E tel que pour a, b $\in$ E

a $\in$ C et (b précède localement a) $\Rightarrow$ (b $\in$ C)

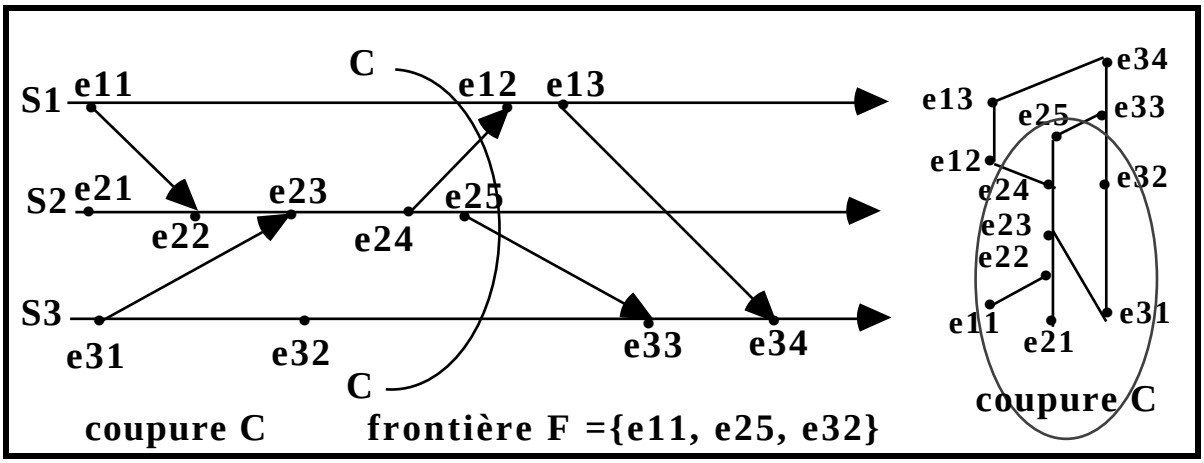
Photographie instantanée d'un système obtenue en prenant un événement par site et tous les événements du site qui le précède.

C'est un sous-ensemble de l'histoire de l'application qui contient toute l'histoire qui le précède : cela permet de définir un passé et un futur
(par rapport à la coupure)

Frontière F d'une coupure C :

ensemble des événements les plus récents de la coupure, un par site

a $\in$ F $\Leftrightarrow$ a $\in$ C et il n'existe pas de b $\in$ C tel que a $\rightarrow$ b


S1
S2
S3
e11
e21
e22
e23
e24
e25
e31
e32
e33
e34
e12
e13
C
C
coupure C
frontière F ={e11, e25, e32}
e34
e13
e25
e33
e12
e24
e32
e23
e22
e11
e21
e31
coupure C

COUPURES COHERENTES

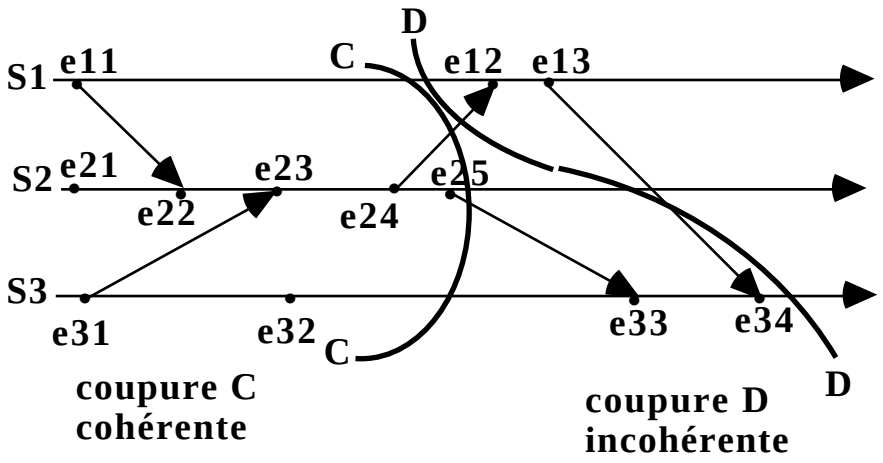
Cohérence = respect de la causalité dans la coupure

une chaîne causale ne peut sortir et re-entrer dans la coupure

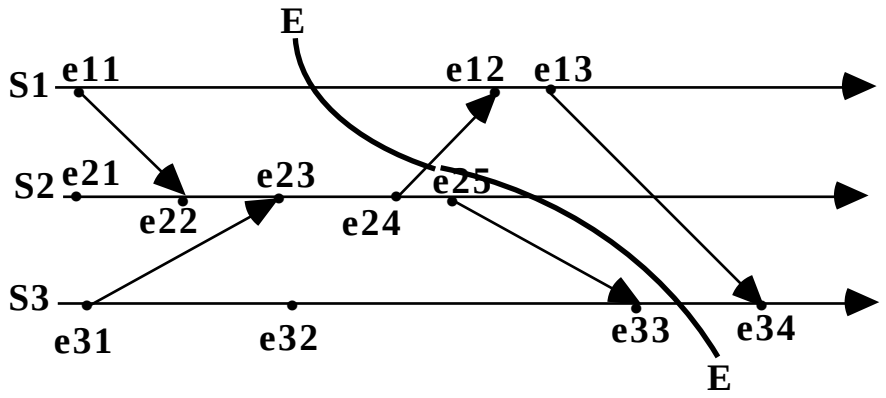
un message ne peut pas venir du futur en franchissant la frontière

Coupure cohérente = coupure fermée par la relation de dépendance causale

$$a \in C \text{ et } (b \rightarrow a) \Rightarrow (b \in C)$$



Etat global cohérent = état associé à une coupure cohérente



coupure E formée par e33 et son passé

Exemple de coupure cohérente : le passé d'un événement

ETAT GLOBAL COHÉRENT D'UN SYSTEME REPARTI

- **soit ELi état local du site Si pour tout i (histoire locale de Si)**
- **soit ECij état local du canal Cij pour tout (i, j)**

état global S = { pour tout i, j, ∪ ELi, ∪ ECij }

coupure associée = { pour tout i, ∪ ELi }

état global S cohérent si coupure associée cohérente

La frontière de la coupure associée définit un passé et un futur

Il en résulte deux conditions nécessaires pour les messages m qui traversent une frontière de coupure

condition C1 :

si EMISSION $_i$(m) $\in$ ELi alors

soit RECEPTION $_j$(m) $\in$ ELj, soit m $\in$ ECij

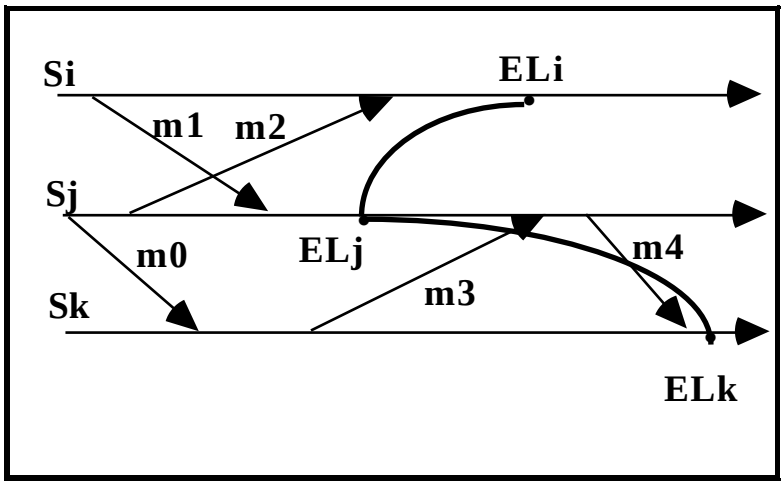
tout message émis dans le passé est soit reçu dans le passé soit en transfert

condition C2 :

si EMISSION $_i$(m) $\notin$ ELi alors RECEPTION $_j$(m) $\notin$ ELj

tout message émis dans le futur reste dans le futur

UNE EXECUTION REPARTIE



état global {ELi, ELj, ELk}

1• EMISSION $_k$(m3) ∈ ELk mais
comme RECEPTION $_j$(m3) ∉ ELj,
il faut que m3 ∈ ECkj
donc la condition C1 est vraie seulement si m3 ∈ ECkj

2• EMISSION $_j$(m4) ∉ ELj mais RECEPTION $_k$(m4) ∈ ELk
la condition C2 est fausse car
C2 : si EMISSION $_j$(m4) ∉ ELj alors RECEPTION $_k$(m4) ∉ ELk

conclusion

{ELi, ELj, ELk} n'est pas un état global cohérent

DETERMINATION D'UN ETAT GLOBAL COHERENT

(Chandy - Lamport 1985)

♦ **Hypothèses**

- **le réseau de communication est connexe**
- **les canaux respectent l'ordre d'émission des messages (canaux FIFO)**
- **un site "élu" particulier lance la détermination d'état global**

♦ **Principe**

- **association d'un message marqueur à chaque enregistrement d'état local d'un site pour que les autres sites puissent repérer les messages qui sont avant ou après cet enregistrement (les messages du passé et du futur)**
- **chaque site détermine son état local et celui de ses canaux en réception, puis envoie ces états au site "élu"**

♦ **Propriétés**

- **l'algorithme se termine**

- **l'état global enregistré correspond à une coupure cohérente**
- **les états enregistrés des canaux sont corrects pour cette coupure**

SOLUTION DE CHANDY et LAMPORT (1985)

♦ HYPOTHÈSE : CANAUX FIFO

tout message mk envoyé sur un canal FIFO sépare les messages du canal en deux sous-ensembles : $<_{mk}$ (avant mk) et $>_{mk}$ (après mk).

♦ CONTRAINTES DE COHÉRENCE

• Quand Si enregistre son état ELi, toutes les émissions de messages faites par Si avant ELi sont captées dans ELi et les réceptions de ces messages doivent être captées dans les ELj et Cij des sites Sj et celles-là seulement.

♦ MISE EN OEUVRE

• Dès que Si enregistre ELi, il émet un message marqueur mk sur chaque canal Cij. Sj doit enregistrer ELj au plus tard à la réception de mk et Sj doit capter tous les messages $<_{mk}$, soit dans ELj soit dans Cji. Les messages de $>_{mk}$ ne doivent pas être captés car ils viennent du futur de ELi sur Si.

♦ ALGORITHME SUR CHAQUE SITE Si

♦ Début (Si "élu") ou première réception d'un marqueur mk (émis par Sj)

(1) enregistrer ELi

(2) enregistrer ECji comme vide car Si a déjà reçu tous les messages $<_{mk}$ et ceux de $>_{mk}$ ne font pas partie de l'état global.

Au démarrage sur "élu", aucun ECji n'est enregistré.

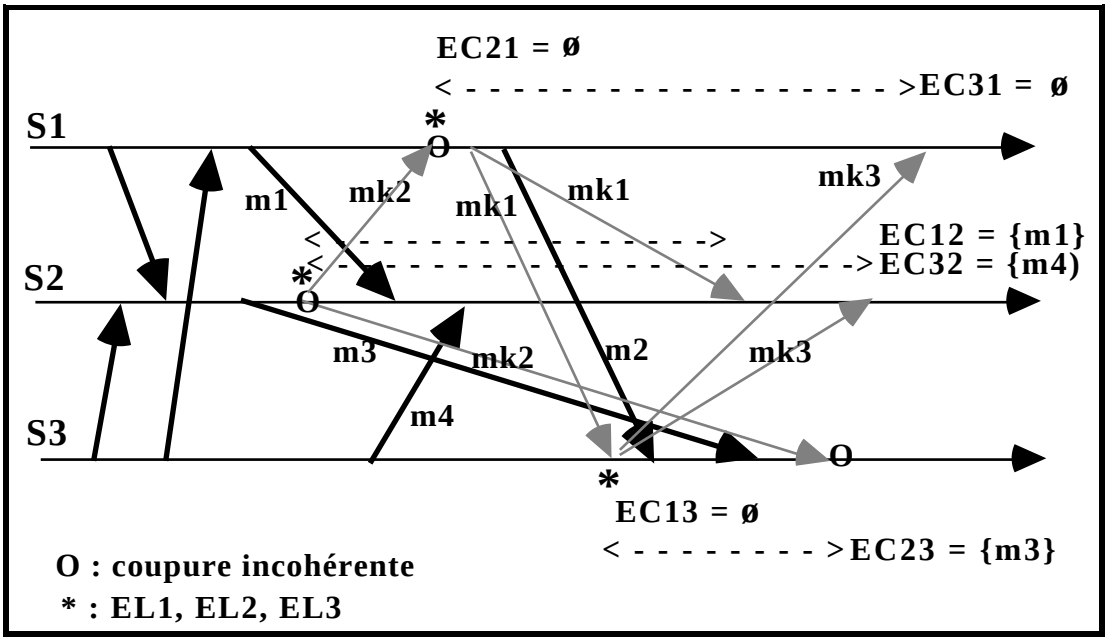
(3) diffuser mk à tous ses voisins sur les canaux Cij

(1-2-3) doit être atomique

♦ **Réceptions suivantes du marqueur mk (émis par Sj)**

enregistrer ECji comme constitué des messages <mk reçus par Si entre l'enregistrement de ELi et l'arrivée de mk (envoyés par Pj avant ELj mais pas reçus par Pi au moment de ELi)

EXEMPLE DE DETERMINATION D'UN ETAT COHERENT

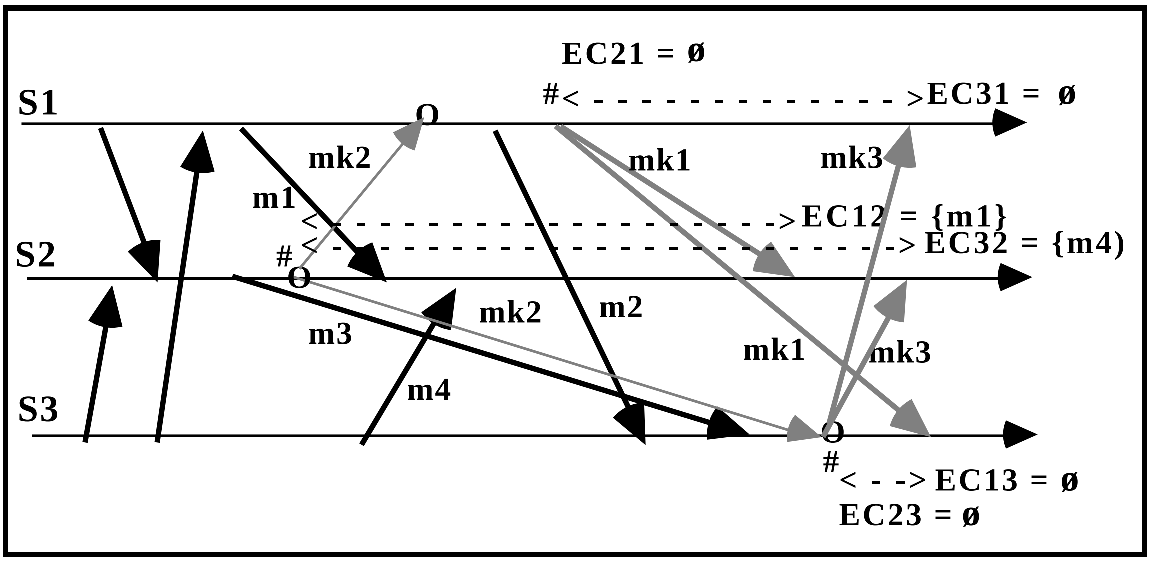


S2 lance la détermination d'état global et diffuse mk2

Les 3 événements émission2(mk2), réception1(mk2) et réception3(mk2) déterminent une coupure incohérente.

Les marqueurs mk1 et mk3 permettent :

- **de forcer la transitivité de la causalité et d'avoir une coupure cohérente.**
- **de noter dans les ECij les messages qui traversent la coupure cohérente.**



autre trace et autre état cohérent

DETERMINER UN ETAT GLOBAL DANS UN SYSTEME REPARTI
éléments de bibliographe

Solution pour un canal FIFO :

K.M. Chandy and L.Lamport, *Distributed Snapshots : Determining Global States of Distributed Systems*. ACM TOCS, Vol 3,1, (1985) pp. 63-75

solutions pour les canaux non FIFO

méthode cumulative (et rapide) de Lai et Yang : T.H.Lai and T.H.Yang, *On Distributed Snapshots*; Inf. Proc. Letters, Vol. 25, (1987), pp. 153-158

méthode non cumulative (mais lente) de Mattern : F.Mattern, *Virtual Time and Global States of Distributed Systems* . Proc. of Int. Workshop on Parallel and Dist. Systems, North Holland, 1988, pp. 215-226

solution utilisant des messages de contrôle : M.Ajuha, *Global Snapshots for Asynchronous Distributed Systems with non FIFO Channels*. Tec. Rep. #CS92-268, U. of Calif., San Diego (1992), 7 p.

solutions fondées sur l'ordre causal

méthode centralisée d'Acharya et Badrinath : A.Acharya and B.R.Badrinath, *Recording Distributed Snapshots Based on Causal Order of Message Delivery*. Inf. Proc. Letters, Vol. 44, (1992), pp.317-321

méthode répartie d'Alagar et Venkatesan : S.Alagar and S.Venkatesan, *An Optimal Algorithm for Distributed Snapshots*

with Causal Message Ordering , Tec. Rep, U. of Texas, Dallas (193), 7 p.

DOCUMENTATION UTILISEE

J.M.Helary, A.Mostefaoui, M.Raynal, *Déterminer un état global dans un système réparti* , RR. 2090, INRIA (1993), 21 p.

ENREGISTREMENT D'UNE TRACE

• **Objectifs :**

• **Reconstruire la séquence des messages échangés pour faire une analyse post mortem ou pour préparer la réexécution d'une situation à analyser.**
Il faut donc conserver la dépendance causale entre les événements dépendants et indiquer les événements causalement indépendants.

• **On date chaque événement e du système avec une méthode de datation causale (l'horloge causale). Soit D(e) la date ainsi fournie.**
Elle permettra de reconstituer la trace si et seulement si :

$$a \rightarrow b \Leftrightarrow D(a) < D(b)$$

$$a \,||\, b \Leftrightarrow \text{non } (D(a) < D(b)) \text{ et non } (D(b) < D(a))$$

C'est la condition forte des horloges car elle inclut : $D(a) < D(b) \Rightarrow a \rightarrow b$

Rappel : condition pour que deux événements a et b soient concurrents, ou encore causalement indépendants :

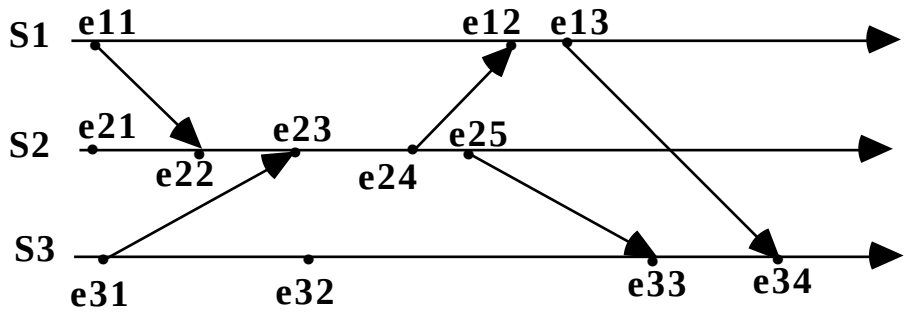
$$a \,||\, b \Leftrightarrow \text{non } (a \rightarrow b) \text{ et non } (b \rightarrow a)$$

UNE METHODE DE DATATION CAUSALE
les historiques

Rappel : passé d'un événement e

• Le passé (ou historique) d'un événement e, c'est par définition :

hist(e) = e $\cup$ ensemble des événements e' tels que e' $\rightarrow$ e



hist(e33) = {e11 e25 e24 e23 e22 e21 e31 e32 e33}

Idée : utiliser le passé de e pour la datation car le passé permet de représenter la dépendance causale :

$$a \rightarrow b \Leftrightarrow a \in hist(b)$$

$$a \,||\, b \Leftrightarrow (a \notin hist(b)) \text{ et } (b \notin hist(a))$$

Gros inconvénient : la taille de hist(e)

Remède : on observe que, pour définir hist(e), un événement par site suffit.

HORLOGES VECTORIELLES

(Fidge, Mattern 1988)

• **Projection de hist(e) sur Si :**

$$hist_i(e) = \{ a \in hist(e) \mid a \in Si \}$$

• **Propriété :** $e_{i,k} \in hist_i(e) \Rightarrow$ **pour tout** $j < k : e_{i,j} \in hist_i(e)$

Si on indice les événements de $hist_i(e)$ **et si un événement avec un indice k appartient à** $hist_i(e)$**, alors tous les événements d'indice inférieur à k font aussi partie de** $hist_i(e)$**.**

• **Alors un seul entier suffit pour représenter** $hist_i(e)$**, c'est le nombre d'événements de** $hist_i(e)$**.**

Comme $hist(e) = \cup hist_i(e)$**, on représente tout hist(e) avec un vecteur V(e)**

$1 \leq i \leq n : V(e)[i] = k$ **tel que** $e_{i,k} \in hist_i(e)$ **et** $e_{i,k+1} \notin hist_i(e)$

V(e)[i] = nombre d'événements de Si "connus de e"

(i. e. connus sur le site de e immédiatement après l'occurrence de e)
(nombre d'événements de l'historique de e qui sont localisés sur Si)

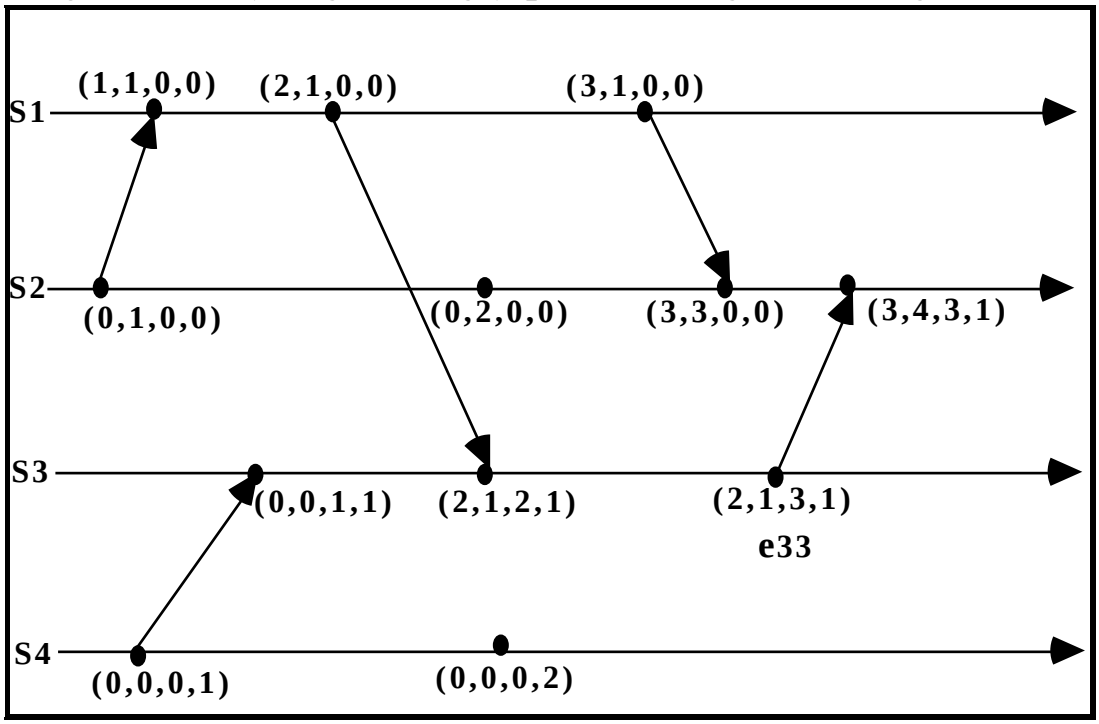
REALISATION DES HORLOGES VECTORIELLES

On associe une horloge vectorielle V_i à chaque site Si

- **Initialement Vi = (0,...,0)**
- **A chaque événement à dater local à Si, on fait $V_i[i] := V_i[i] + 1$**
- **Chaque message m porte une estampille Vm (Vm = Vi de l'émetteur)**
- **A la réception de (m, Vm) par un site Si, on enrichit l'historique connu par Si avec l'historique transporté par m :**

  $V_i[i] := V_i[i] + 1$

  $V_i[j] := \max(V_i[j], Vm[j])$ pour tous $j = 1,...n, j \neq i$



- **Autre façon de connaître l'"heure" de e 33:**

  l'histoire de e comprend

  2 événements sur S1

1 événement sur S2

3 événements sur S3

1 seul événement sur S4

PROPRIETES DES HORLOGES VECTORIELLES

• Relation d'ordre partiel sur les horloges vectorielles :

$V \leq V'$ défini par : quel que soit i, $V[i] \leq V'[i]$

$V < V'$ défini par $V \leq V'$ et $V \neq V'$

$V \,||\, V'$ défini par $\neg (V < V')$ et $\neg (V' < V)$

Les horloges vectorielles représentent <u>exactement</u> la dépendance causale :

pour tout a, b

$$a \rightarrow b \Leftrightarrow V(a) < V(b)$$

$$a \,||\, b \Leftrightarrow V(a) \,||\, V(b)$$

• Les horloges vectorielles sont "denses" :

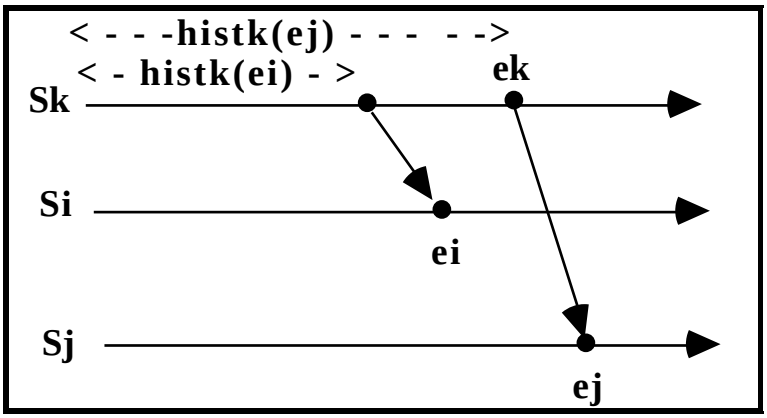
Soit $ei \in Si$, $ej \in Sj$,

si $V(ei)[k] < V(ej)[k]$, pour $k \neq j$, alors il existe ek sur Sk tel que

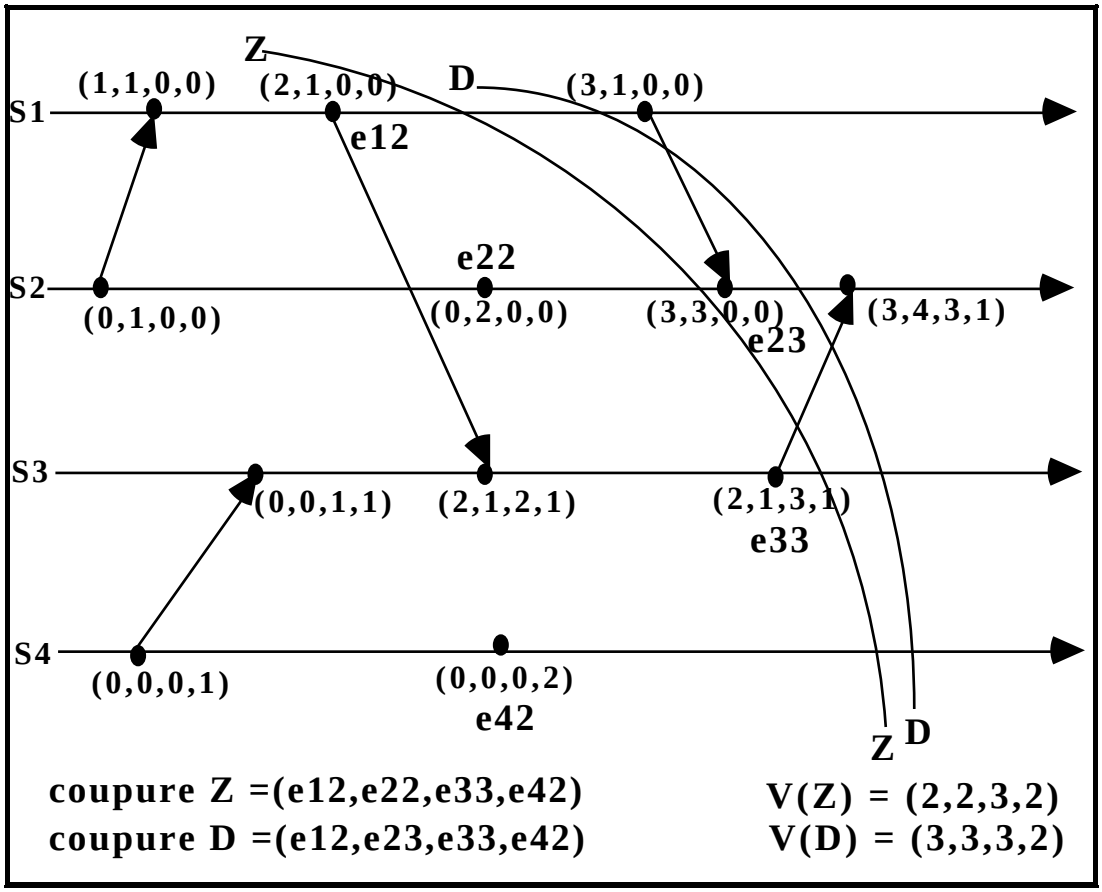
$\neg(ek \rightarrow ei)$ et $(ek \rightarrow ej)$

En effet $V(ei)[k] = |hist_k(ei)|$ = nombre d'événements de hist(ei) sur Sk

$V(ej)[k] = |hist_k(ej)|$ = nombre d'événements de hist(ei) sur Sk


< - - -histk(ej) - - - - ->
< - histk(ei) - >
ek
Sk
Si
ei
Sj
ej

HORLOGES VECTORIELLES ET COUPURES COHERENTES



- **Date d'une coupure C = (c1,c2,...,cn)**

  V(C) = sup (V(c1),...,V(cn))

  soit pour tout i, V(C)[i] = sup (V(c1)[i],...,V(cn)[i])

La coupure est cohérente si et seulement si

V(C) = (V(c1)[1],...,V(cn)[n])

V(Z) = (2, 2, 3, 2) = (V(e12)[1], V(e22)[2], V(e33)[3], V(42)[4])

V(D) = (3, 3, 3, 2)

(V(e12)[1], V(e23)[2], V(e33)[3], V(42)[4]) = (2, 3, 3, 2)

• Z est cohérente, D ne l'est pas

DIFFUSION FIABLE AVEC ORDRE CAUSAL
(Birman, Schiper, Stephenson 1990)

• Utilisation des horloges vectorielles pour garantir la causalité

• Si un message arrive trop tôt pour la causalité, on le fait attendre

• On n'estampille que les émissions

1) avant diffusion de m, le site Si exécute V ¡[i] := V ¡[i] + 1

2) estampille de m par Si : Vm = Vi

3) à la réception sur Sj de (m, Vm) diffusé par Si, on attend que :

a) toutes les diffusions précédentes de Si soient arrivées sur Sj

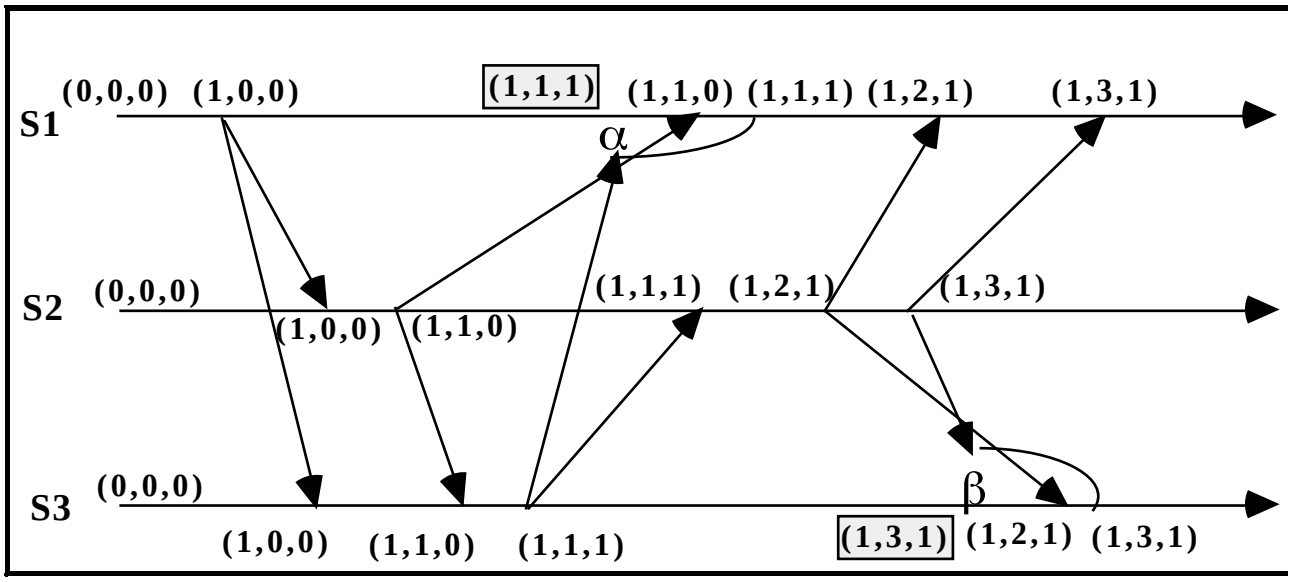
soit Vj[i] = Vm[i] - 1

b) toutes les diffusions antérieures à m et reçues sur Si aient été aussi reçues par Sj

soit pour tous k ≠ i, Vj[k] ≥ Vm[k]

4) après remise de m, on enregistre l'historique connu grâce à m

soit Vj := max (Vj, Vm)



en α : V1 = (1,0,0) et Vm = (1,1,1) soit V1[3] = Vm[3] - 1; V1[2] < Vm[2]

en β : V3 = (1,1,1) et Vm = (1,3,1) soit V3[2] ≠ Vm[2] - 1; V3[1] ≥ Vm[1]

LINEARISATION DE L'OBSERVATION D'UN SYSTEME

• Linéarisation de l'observation d'un système réparti S:

• "vue" en séquence des événements de S par un observateur interne

• "vue" en séquence des événements de S par un observateur externe

• C'est la définition d'un ordre total, soit <<, sur les événements de S

• Linéarisation valide si elle est compatible avec la relation de précédence causale

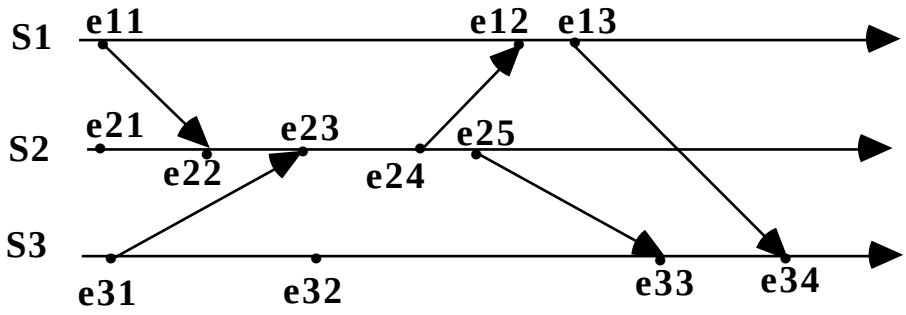
$$e, e' \in S : e \rightarrow e' \Rightarrow e << e'$$

• Exemple :

e11 e21 e31 e22 e23 e32 e24 e25 e12 e33 e13 e34 valide

e11 e21 e31 e22 e32 e23 e24 e25 e12 e33 e13 e34 valide (car e32 ||e23)

e11 e21 e31 e22 e23 e32 e24 e25 e12 e33 e34 e13 invalide (car e13 → e34)



• On date chaque événement e du système avec une méthode de datation totale (l'horloge logique). Soit H(e) la date ainsi fournie qui est valide ssi:

$$e \rightarrow e' \Rightarrow H(e) < H(e')$$

Nota. $H(e) < H(e') \Rightarrow non(e' \rightarrow e)$

C'est la condition faible des horloges car, ou bien $e \rightarrow e'$**, ou bien** $e \,||\, e'$

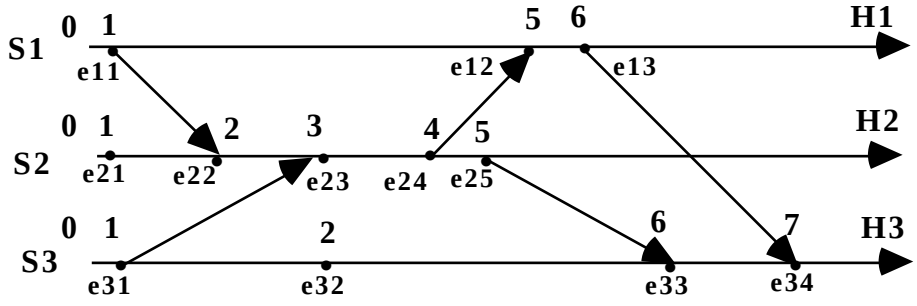
REALISATION DES HORLOGES LOGIQUES

(Lamport 1978)

- **Objectif : réaliser une datation totale des événements**
  - **respectant la dépendance causale**
  - **déterminable par consultation locale**

Un compteur entier Hi, initialisé à 0, est maintenu sur chaque site Si.

- **A chaque événement e, localisé sur Si :**
  - **Hi := Hi + 1**
  - **e est daté par Hi : H(e) := Hi**
- **Si e est l'émission d'un message m, celui-ci est estampillé par la date de son émission sur Si, obtenue en lisant Hi :**
  - **E(m) = H(émission(m)) := Hi**
- **Si e est la réception d'un message m venant de Sj, l'estampille E(m) est utilisée par le compteur Hi (l'horloge logique locale sur Si) pour rattraper le retard sur Hj, s'il en a, avant de dater d'incrémenter Hi pour dater e.**
  - **Hi := max (Hi, E(m)) + 1**



• Réalisation d'un ordre total (<<) : soit a sur Si, b sur Sj

donc H(a) = $H_i(a)$ et H(b) = $H_j(b)$

a << b ⇔ (H(a) < H(b) ou (H(a) = H(b) et i < j)

l'ordre total sur les sites est arbitraire (il doit être le même partout)

EVALUATION DES HORLOGES LOGIQUES

♦ Hypothèses de validité

• le réseau de communication est connexe et fiable

• le temps de transmission des messages est borné supérieurement

• pas de communication cachée qui donneraient un autre ordre total

(via des canaux comme le téléphone entre utilisateurs,

ou comme une heure universelle fournie par satellite).

♦ Avantages

• première datation répartie introduite

• économique : datation par un seul nombre et non par un vecteur

• causalité des messages respectée par remise à l'heure du récepteur

♦ Utilisation importante des estampilles

• ordre total : e11 e21 e31 e22 e32 e23 e24 e12 e25 e13 e33 e34

• exclusion mutuelle répartie

• mise à jour de copies multiples

• diffusion fiable ordonnée totalement

• datation des transactions réparties pour gérer les verrous des BD

• datation des transactions réparties pour la prévention d'interblocage

• gestion cohérente de fichiers répartis

• génération répartie de noms uniques pour la désignation interne

• génération répartie de noms uniques pour l'authentification

♦ Limitation de la datation par estampilles

• la notion de dépendance causale est effacée artificiellement, car

• les estampilles ne sont pas denses : si H(e) < H(e'),

on ne peut pas savoir s'il existe e'' tel que e → e'' ou e'' → e'

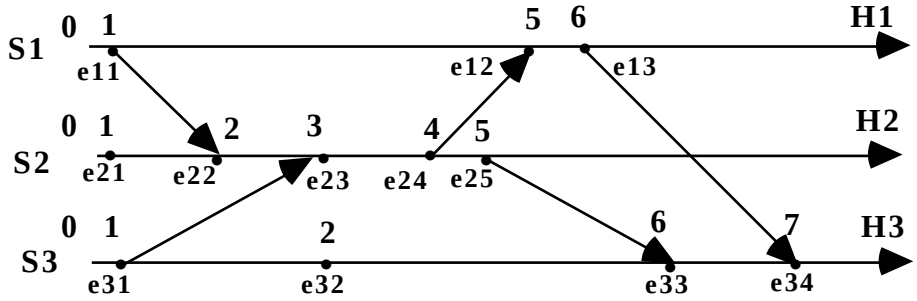
LIMITATION DE LA DATATION PAR HEURE LOGIQUE

$e \rightarrow e' \Rightarrow H(e) < H(e')$

• c'est la condition faible des horloges car

(R) $H(e) \leq H(e') \Rightarrow non(e' \rightarrow e)$ [$(a \Rightarrow b) \Leftrightarrow (non\ b \Rightarrow non\ a)$]

c'est à dire : ou bien $e \rightarrow e'$, ou bien $e \,||\, e'$



a) • $e11 \rightarrow e33$ se traduit en $H1(e11) < H3(e33)$

$e11 << e33$

• $e||e'$ donne n'importe quoi :

$e32||e12$ donne $H3(e32) < H1(e12)$

$e32 << e12$

$e32||e22$ donne $H3(e32) = H2(e22)$

$e32 >> e22$

$e32||e11$ donne $H3(e32) > H1(e11)$

$e32 >> e11$

b) • $H(e22) < H(e33) \Rightarrow non(e33 \rightarrow e22)$ ici $e22 \rightarrow e33$

• $H(e12) < H(e33) \Rightarrow non(e33 \rightarrow e12)$ ici $e32 \,||\, e12$

c) • seule certitude $H(e) = H(e') \Rightarrow e||e'$

$H(e) = H(e') \Leftrightarrow (H(e) \leq H(e'))$ et $(H(e) \geq H(e'))$

$(H(e) \leq H(e'))$ et $(H(e) \geq H(e')) \Rightarrow non(e' \rightarrow e)$ et $non(e \rightarrow é)$

[par (R)] $\Rightarrow e||e'$ CQFD

exemple $H(e13) = H(e33) = 6$. On a bien $e13 \,||\, e33$

• Les estampilles ne sont pas denses : soit e et e' tels que H(e) < H(e'),

on ne peut pas savoir s'il existe e'' tel que e → e'' et/ou e'' → e'

dans l'exemple : H(e22) = 2; H(e32) = 2; H(e33) = 6

H(e22) < H(e33) et il existe e24 tel que e22 → e24 et e24 → e33

H(e32) < H(e33) et il n'existe pas e'' tel que e → e'' et/ou e'' → e'

EXCLUSION MUTUELLE REPARTIE
(Lamport 1978)

♦ **Hypothèses**
- **le nombre N des sites est connu**
- **les canaux sont FIFO**
- **ordre total réalisé par horloge logique**

♦ **Principe : connaissance mutuelle répartie acquise par chaque site"**

♦ **Demande d'entrée d'un site Si en section critique :**
diffusion de (req, H(req), i) à tous les sites, Si compris
entrée en section critique quand Si sait que:
a) tous les sites ont reçu sa demande ou qu'ils en ont émis une aussi b) sa demande est la plus ancienne de toutes

♦ **Réception par Si d'une demande d'entrée en section critique de Sj :**
réponse systématique par (acq, H(acq), Si)

♦ **Libération de la section critique par Si :**
diffusion de (lib, H(lib), Si) à tous les sites, Si compris

♦ **Structure de données sur chaque site : tableau de N messages, 1 par site**
- **initialement sur Si Mij := (lib, 0, Sj) pour tout j**
- **réception de M = (req, H(req), j) ou (lib, H(lib), Sj) => Mij := M**
- **réception de M = (acq, H(acq), Sj) =>**
si Mij ≠ (req, H(req), j) alors Mij := M
si l'acquittement vient après une requête de Sj, on n'oublie pas celle-ci

♦ **Règle de décision pour chaque site Si :**

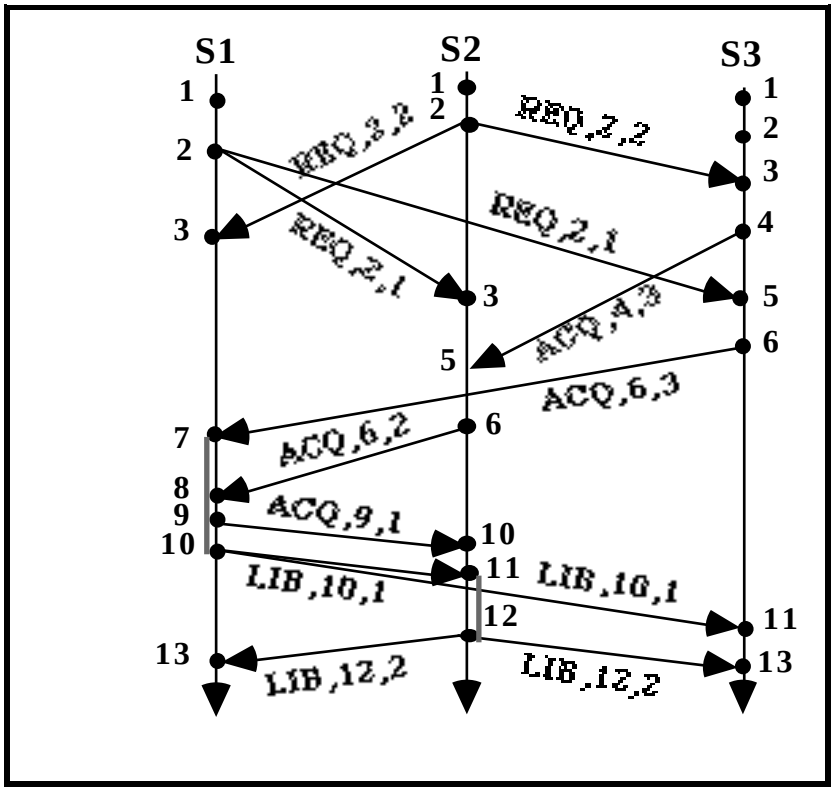
♦♦ **Si entre en section critique quand sa demande Mii << Mij pour tout j**
En effet, comme les canaux sont FIFO,

il n'y a plus de requête plus ancienne en chemin dans un canal

♦ Propriétés :

- **Si est seul en section critique.**
- **Il n'y a pas de coalition car les entrées suivent l'ordre total**
- **Mais il faut 3(n - 1) messages**

EXCLUSION MUTUELLE REPARTIE
(Lamport 1978)
EXEMPLE



	H1		H2		H3	
M11	LIB,0,1	0	M21 LIB,0,1	0	M31 LIB,0,1	0
M12	LIB,0,2	0	M22 LIB,0,2	0	M32 LIB,0,2	0
M13	LIB,0,3	0	M23 LIB,0,3	0	M33 LIB,0,3	0
M11	DEM,2,1	2	M21 DEM,2,1	3	M31 DEM,2,1	5
M12	DEM,2,2	3	M22 DEM,2,2	2	M32 DEM,2,2	3
M13	ACQ,6,3	7	M23 ACQ,4,3	5	M33 LIB,0,3	0

S1 entre en s.c. à H1 = 7

M21	LIB,10,1	11
M22	DEM,2,2	2
M23	ACQ,4,3	5

S2 entre en SC à H2 = 11

EXCLUSION MUTUELLE REPARTIE

(Ricart, Agrawala 1981)

♦ Hypothèses

• le nombre N des sites est connu

• les canaux sont quelconques (hypothèse FIFO non nécessaire)

• ordre total réalisé par horloge logique

♦ Principe : file d'attente répartie par morceau sur certains sites

♦Demande d'entrée d'un site Si en section critique :

diffusion de (req, H(req), i) à tous les sites, Si compris

entrée en section critique quand tous les sites ont donné leur accord

♦ Réception par Si d'une demande d'entrée en section critique de Sj :

• Si n'est ni en section critique ni candidat : accord (acc, H(acc), Si)

• Si est lui-même candidat :

accord (acc, H(acc), Si) si demande de Sj antérieure

sinon mise en attente par Si de la demande de Sj

• Si est en section critique :

mise en attente par Si de la demande de Sj

♦ Libération de la section critique par Si :

accord (acc, H(acc), Si) à toutes les demandes mises en attente par Si

♦ **Propriétés :**

- **Si est seul en section critique.**
- **Il n'y a pas de coalition car les entrées en S.C. suivent l'ordre total**

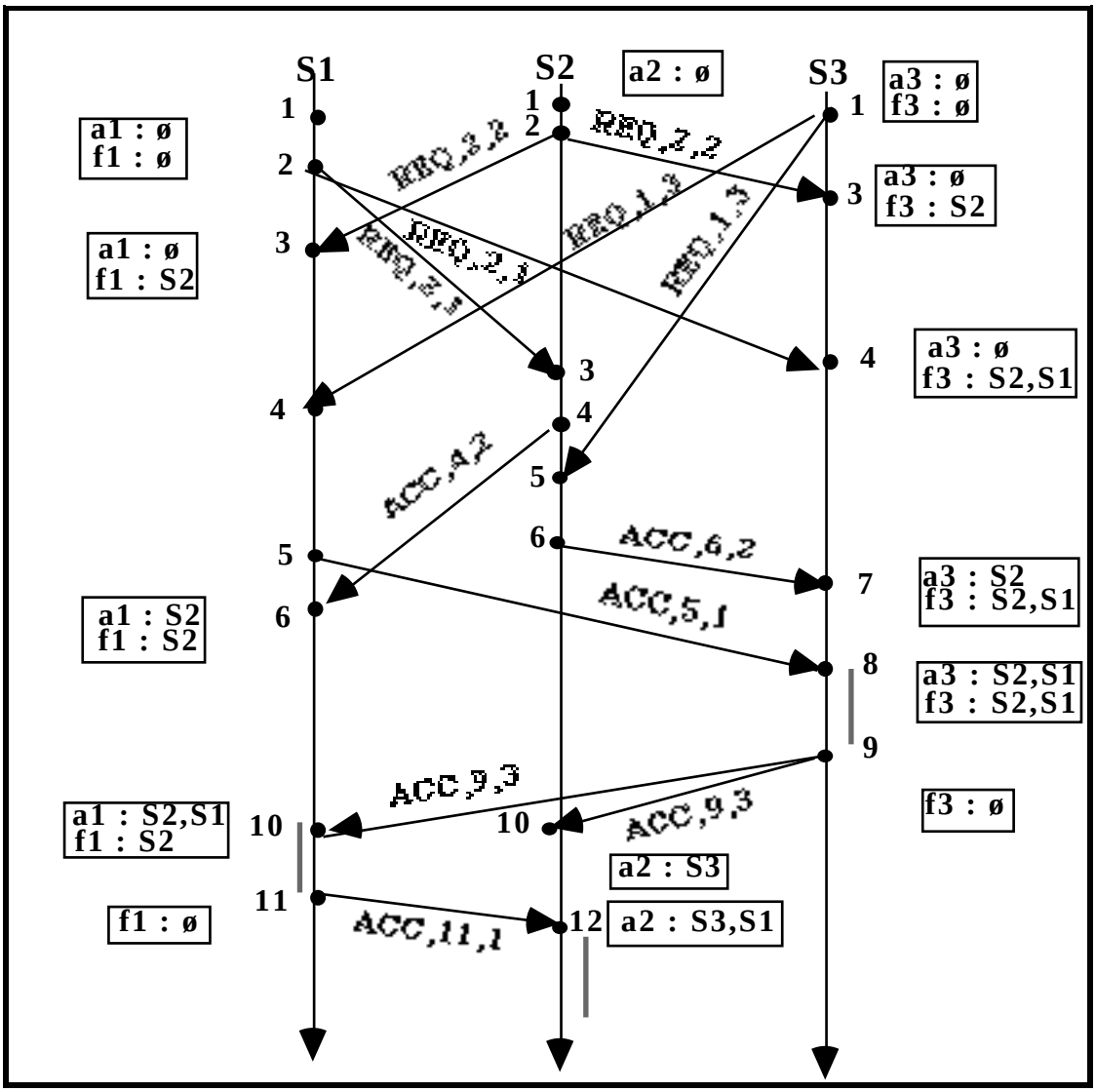
(équité faible car les messages peuvent se doubler)

- **Il faut 2(n - 1) messages**

EXCLUSION MUTUELLE REPARTIE

(Ricart, Agrawala 1981)

EXEMPLE



légende :ai : ensemble des ACC reçus par le site i
fi : ensemble des sites mis en attente par Si