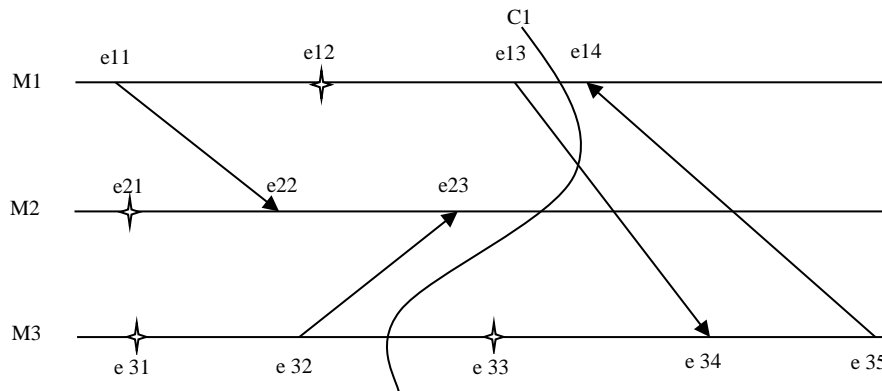


Série d'exercices N° 1 (Etat global d'un système réparti)

Exercice 1 :

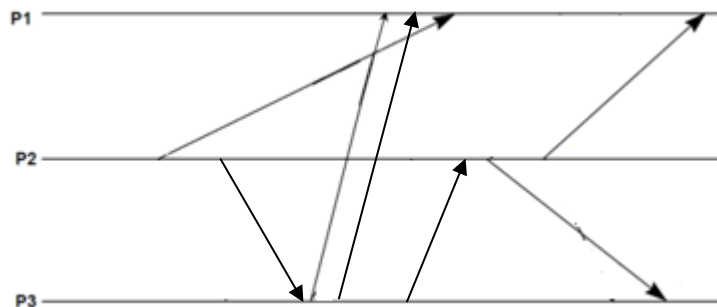
Soit un système distribué composé de trois machines (Schéma 1) M_1 , M_2 et M_3 . On définit deux coupures C_1 et C_2 . C_2 est représentée par l'évènement e_{14} sur le site M_1 , e_{23} sur le site M_2 et e_{34} sur le site M_3 .



1. Vérifier la cohérence de chacune des deux coupures C_1 et C_2 .
2. Dater les évènements en utilisant les horloges scalaires.
3. Etablir l'ordre total du système. Qu'est que vous remarquez ?
4. Etablir l'ordre total de chacune des deux coupures C_1 et C_2 .
5. Utiliser les horloges de Mattern pour refaire la question 1.

Exercice 2 :

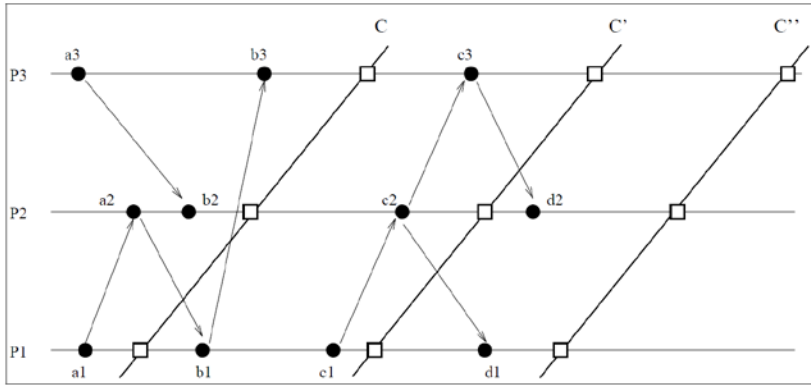
Considérons le schéma d'exécution d'évènements suivant (Schéma 2).



1. Dater les évènements en utilisant l'algorithme de Mattern.
2. Soit C la plus grande chaîne d'ordre partiel d'évènements :
 - Quelle est la taille de C ?
 - Donner l'ensemble (s) des vecteurs qui composent C .
 - Comparer C avec l'ensemble des dates vectorielles des différents évènements du système.
 - Qu'est qu'on peut déduire ?
3. Est-ce que le schéma contient un non-respect du protocole de délivrance causale des messages? Justifier. Que peut-on dire?
4. Dater les évènements en utilisant les horloges matricielles? Comparer avec la datation de la question 1.

Exercice 3 :

On considère le diagramme temporel (figure 1) associé à un algorithme distribué faisant intervenir 3 processus P_1 , P_2 , P_3 et les évènements a_1, b_1, c_1, d_1 sur P_1 , a_2, b_2, c_2, d_2 sur P_2 , a_3, b_3, c_3 sur P_3 .



- 1- Calculer les estampilles de Lamport pour chacun de ces événements.
- 2- Calculer les "horloges vectorielles" (vecteurs à 3 composantes) pour chacun de ces événements, puis commenter les valeurs des vecteurs estampillant les événements c1 et c3.
- 3- Dire comment les événements b3 et c2 estampillés par les horloges de Lamport se comparent (sont-ils comparables selon la relation de causalité)? Puis, en considérant leur estampille vectorielle, dire comment ces événements b3 et c2 se comparent. Recommencer exactement la même question en considérant à présent les événements b3 et d2.
- 4- Décrire en termes d'événements les coupures C, C', C''. Sont-elles cohérentes ?

Exercice 4 :

Dans un système distribué, il existe des cas où un site doit envoyer le même message à tous les autres sites. Ceci peut être réalisé par une succession de communications entre chaque couple de sites concerné (l'émetteur, un récepteur). Le problème intervient si des sites n'ont pas reçu leur message (erreur de transmission, tampon plein ...).

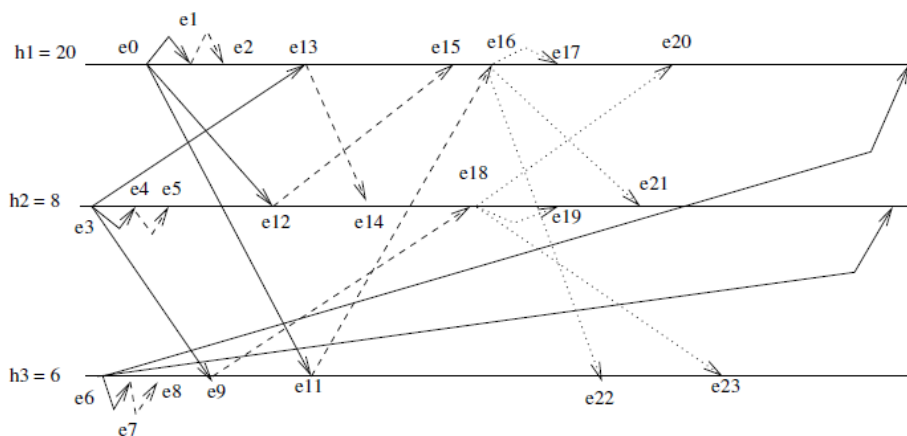
Certains protocoles assurent que tout message émis par l'un des sites est reçu par tous les sites. Le protocole ABCAST (Atomic BroadCAST) est l'un des protocoles de diffusion qui utilise les dates scalaires comme suit :

Un site S_i émet un message M en diffusion en le datant par son horloge locale H_i : $EL_M = H_i$. Puis, il incrémente H_i . S_i maintient un tableau Ack de taille n lui permettant de mémoriser les acquittements reçus pour le message M.

Un site S_j qui reçoit un message M, le met dans une file d'attente locale AM au site pour sauvegarder les messages reçus (Les messages M) et non utilisables. Il renvoie à l'émetteur un accusé de réception R estampillé (EL_R) par la date de réception du message M puis il incrémente H_j .

Un site S_i qui reçoit un accusé de réception R de S_j , met $Ack[j]$ à 1 et mémorise EL_R . Quand l'émetteur a reçu tous les accusés de réception, il valide le message M en envoyant aux destinataires un message V d'estampille EL_V égale au maximum des estampilles des accusés reçus.

Un site S_j qui reçoit un message de validation V d'estampille EL_V correspondant à un message M, il date le message M par l'estampille EL_V , effectue $H_j = \max(EL_V, H_j) + 1$, et met M dans la liste des messages utilisables UM.

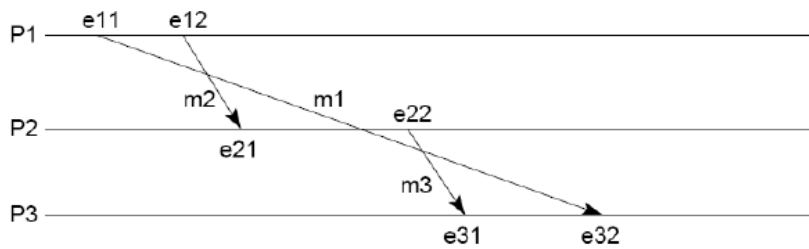


1. Définir les structures de données et les variables utilisées.
2. Donner pour chaque évènement:
 - Le site,
 - L'action (Diffusion, Réception Accusé, Réception de Validation, Emission d'Accusé, ...),
 - La date,
 - Les valeurs des variables,
 - La valeur de l'horloge locale après l'évènement.

Exercice 5 :

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 em son événement d'émission, rm son événement de réception et dm 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).



1. Mettre en évidence le non-respect des dépendances causales en émission pour le chronogramme ci-dessus (ce chronogramme ne correspond pas à celui du système décrit à la page précédente). Placer les événements de délivrement des messages sur le chronogramme en respectant ces dépendances causales.
2. Déterminer 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. Les horloges de Mattern ou de Lamport permettent-elles de détecter le non-respect des dépendances causales en émission et de bien ordonner les délivrances des messages ?
4. Proposer une nouvelle méthode de datation permettant d'assurer l'ordre causal.