

# Accord et coordination

## Diffusion atomique

Master1 RID

# Accord et coordination

- ❑ Une famille de problèmes en algorithmique distribuée.
- ❑ **Catégories de problèmes de cette famille :**
  - ❑ Accord sur l'accès à une ressource partagée
    - ❑ **Exclusion mutuelle distribuée**
  - ❑ Accord sur l'ordre d'envoi de messages à tous
    - ❑ **Diffusion atomique**
  - ❑ Accord sur un processus jouant un rôle particulier
    - ❑ **Élection d'un maître**
  - ❑ Accord sur une valeur commune
    - ❑ **Consensus**
  - ❑ Accord sur une action à effectuer par tous ou personne
    - ❑ **Transaction**

**Accord sur l'ordre d'envoi de messages à tous**

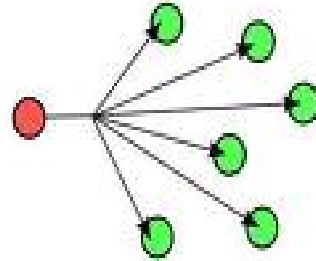
**« Diffusion atomique »**

# Plan

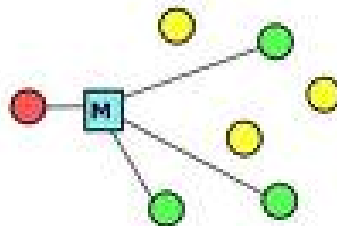
- ❑ **Diffusion**
  - ❑ Basique
  - ❑ Fiable
- ❑ **Ordonnancement**
  - ❑ FIFO
  - ❑ Causal
  - ❑ Total
- ❑ **Diffusion atomique**
- ❑ **Types de diffusion**
- ❑ **Relations entre les diffusions**

# Diffusion

- ❑ Mode d'émission de messages avec multiples récepteurs.
  - ❑ Un message émis par un émetteur est reçu par plusieurs récepteurs.
  - ❑ Tous les processus du groupe dans le cas de la diffusion (**broadcast**).

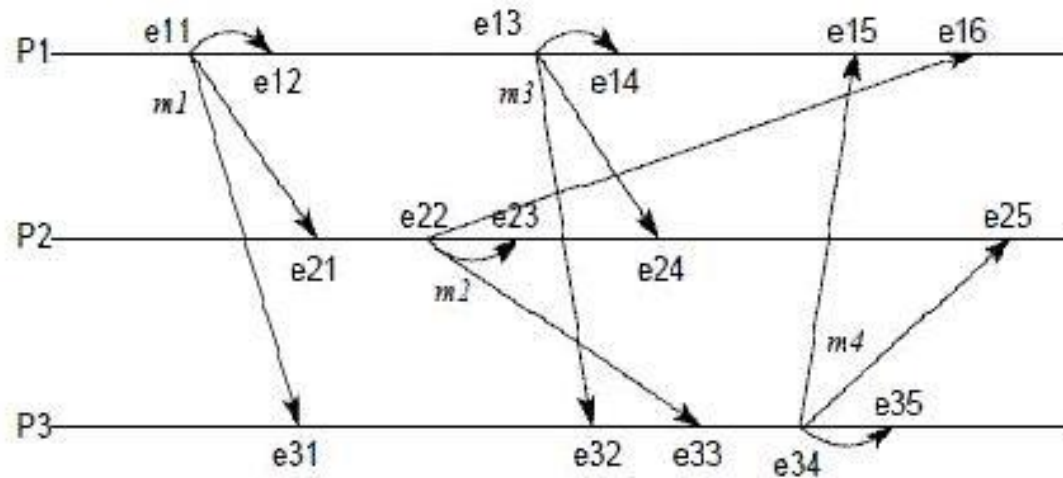


- ❑ Les processus d'un sous-groupe dans le cas du **multicast**.



# Diffusion

- ❑ Ordre et fiabilité de la diffusion des messages.
  - ❑ Par défaut et selon les temps de propagation, les messages arrivent ... quand ils arrivent.
    - ❑ Pas dans le même ordre ni dans un ordre donné pour tous les processus.
  - ❑ Peuvent aussi ne pas arriver chez tous les processus si canaux de communication non fiables ou si l'émetteur plante en cours de diffusion.



**Chronogramme de diffusion de messages**

# Diffusion : support de communication

- ❑ Coté récepteur, on différencie deux actions :
  - ❑ **La réception du message** par le système de communication associé au processus.
  - ❑ **La délivrance du message** : le processus est informé du message.
    - ❑ Un message reçu peut ne pas être délivré selon les cas.
    - ❑ La délivrance d'un message peut être retardé par rapport à sa réception tant que certaines conditions d'ordre ne sont pas respectées.
- ❑ Mise en œuvre peut être faite sur plusieurs types de supports de communication :
  - ❑ **Couche réseau réalisant nativement de la diffusion.**
    - ❑ Typiquement : multicast UDP/IP (mais non fiable).
  - ❑ **Communication point à point « classique ».**

# Protocole de diffusion basique

## ❑ Modèle de communication et de faute :

- ❑ Canaux de communication fiable 1 vers 1 entre tous les processus.

- ❑ Processus peuvent planter (panne franche).

- ❑ Système asynchrone : aucune borne.

- ❑ Sur la durée d'exécution d'une étape du processus.

- ❑ Sur le délai de réception d'un message.

- ❑ Sur la dérive de l'horloge locale d'un processus par rapport au temps réel.



# Protocole de diffusion basique

## □ Protocole de diffusion basique, 2 opérations :

□ **B\_broadcast( $m$ )** exécuté par processus  $p$ .

□ Le processus  $p$  diffuse un message  $m$  à tous les autres processus.

□ **B\_deliver( $m$ )** exécuté sur processus  $p$ .

□ délivrance du message  $m$  à  $p$ .

## □ Opérations associées aux canaux de communication :

□ **receive( $m$ ,  $q$ )** sur processus  $p$ .

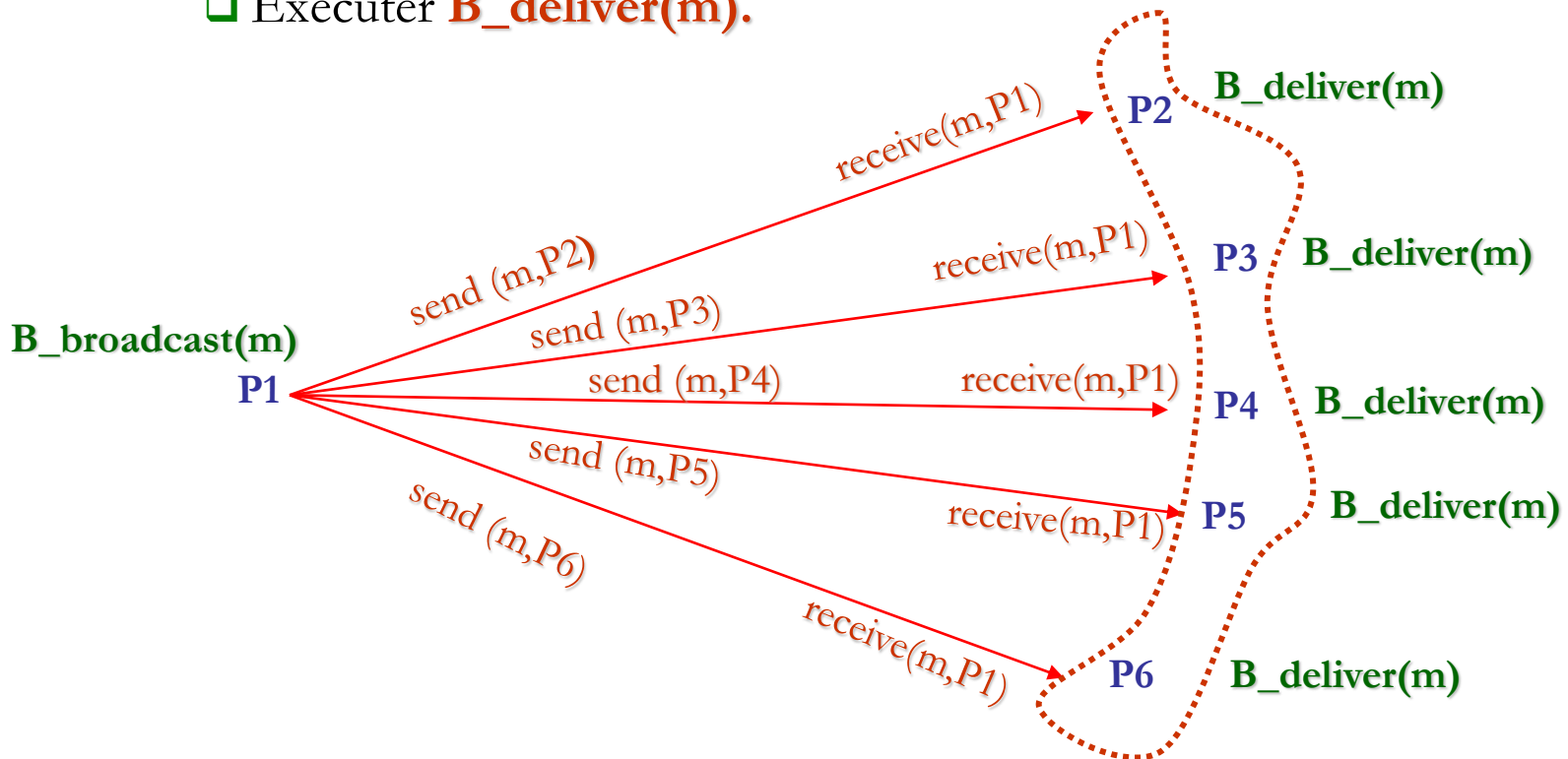
□ Réception du message  $m$  venant de  $q$ .

□ **send( $m$ ,  $q$ )** sur processus  $p$ .

□ Envoi par  $p$  du message  $m$  au processus  $q$ .

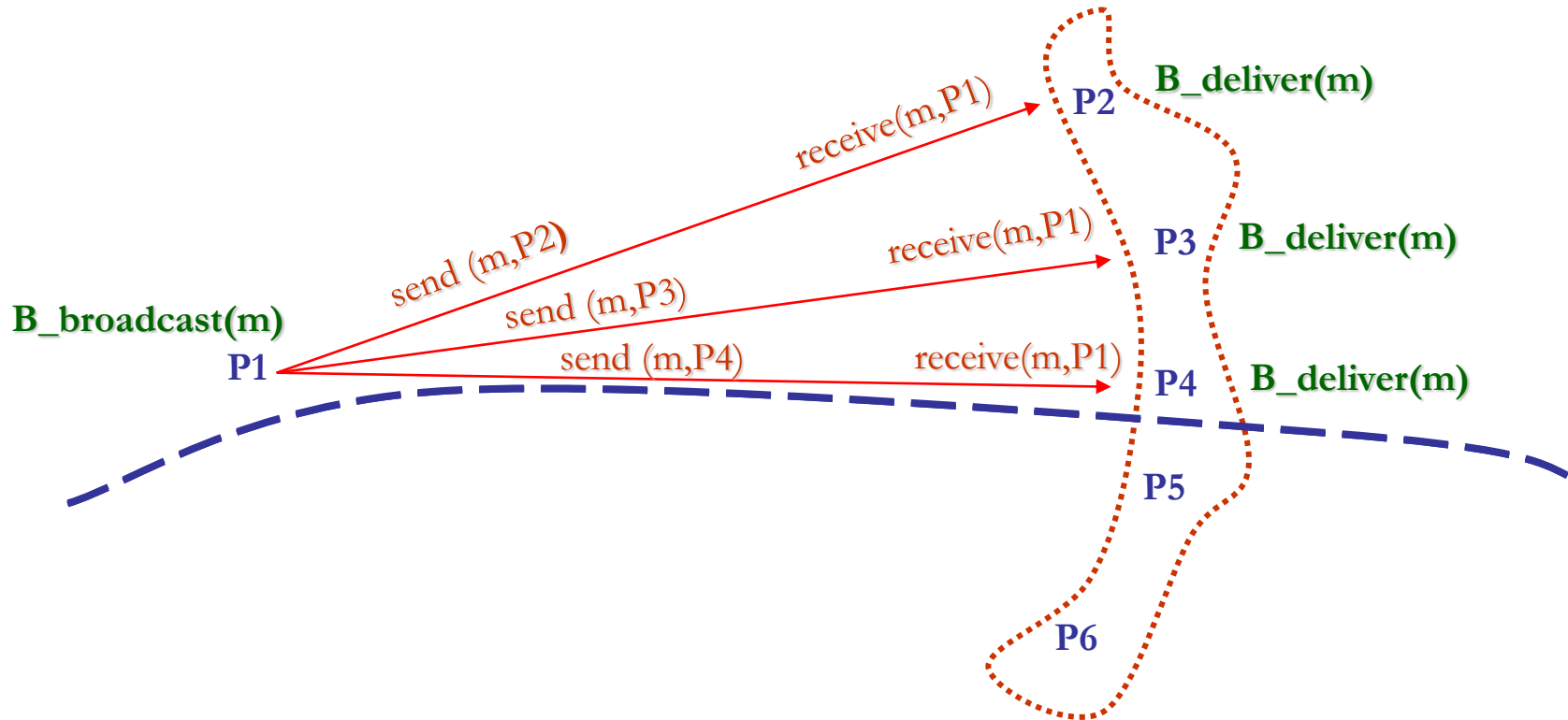
# Diffusion basique

- ❑ Mise en œuvre diffusion basique :
  - ❑ **B\_broadcast(m)** sur  $p$ .
    - ❑ On envoie le message à tout le monde.
      - ❑ Pour chaque processus  $q$  du groupe : exécuter **send (m, q)**.
  - ❑ Lors d'un **receive(m, q)** sur  $p$ .
    - ❑ On vient de recevoir un message  $m$  venant de  $q$ , on le délivre directement.
      - ❑ Exécuter **B\_deliver(m)**.



# Problème de la diffusion basique

- ❑ Problème : pendant la diffusion, P1 tombe en panne.
- ❑ P5 et P6 ne reçoivent pas le message m.



# Diffusion fiable

- ❑ **Diffusion fiable (*reliable broadcast*)** : tous les processus corrects délivrent un message diffusé.
- ❑ **Caractéristiques à assurer :**
  - ❑ **Validité :**
    - ❑ Si un processus correct diffuse  $m$ , alors il finira par être délivré par tous les processus corrects (en un temps fini).
      - ❑ **Un message diffusé est délivré par tout le monde.**

# Diffusion fiable

- ❑ **Diffusion fiable (*reliable broadcast*)** : tous les processus corrects délivrent un message diffusé.
- ❑ **Caractéristiques à assurer :**
  - ❑ **Validité :**
    - ❑ Si un processus correct diffuse  $m$ , alors il finira par être délivré par tous les processus corrects (en un temps fini).
      - ❑ **Un message diffusé est délivré par tout le monde.**
  - ❑ **Accord :**
    - ❑ Si un processus correct délivre  $m$ , alors tous les autres processus corrects délivreront  $m$  également (en un temps fini).
      - ❑ **Tout le monde délivre les mêmes messages.**

# Diffusion fiable

- ❑ **Diffusion fiable (*reliable broadcast*)** : tous les processus corrects délivrent un message diffusé.
- ❑ **Caractéristiques à assurer :**
  - ❑ **Validité :**
    - ❑ Si un processus correct diffuse  $m$ , alors il finira par être délivré par tous les processus corrects (en un temps fini).
    - ❑ **Un message diffusé est délivré par tout le monde.**
  - ❑ **Accord :**
    - ❑ Si un processus correct délivre  $m$ , alors tous les autres processus corrects délivreront  $m$  également (en un temps fini).
    - ❑ **Tout le monde délivre les mêmes messages.**
  - ❑ **Intégrité :**
    - ❑ Un processus correct délivre un message  $m$  au plus une fois, avec  $m$  un message qui a été diffusé par un processus.
    - ❑ **Pas de duplication ni de création de message.**

# Diffusion fiable

- ❑ Le protocole de diffusion basique n'est pas fiable.
  - ❑ Pourtant la communication est fiable.
    - ❑ Pas de perte de messages.
- ❑ Le problème est pour l'exécution de **B\_broadcast(m)**.
  - ❑ Pendant que l'on parcourt la liste des processus pour leur envoyer un par un le message, le processus émetteur peut planter.
    - ❑ Certains processus recevront le message mais pas tous.
    - ❑ **Les propriétés d'accord et de validité ne sont pas respectées.**
- ❑ Définition d'opérations de diffusion fiable :
  - ❑ **R\_broadcast(m) et R\_deliver(m)**.
    - ❑ Variantes fiables des opérations de la diffusion basique (R : reliable).
    - ❑ Utiliseront de la diffusion basique.

# Diffusion fiable

- ❑ Mise en œuvre de la diffusion fiable :

- ❑ **R\_broadcast(m)** sur  $p$ .

- ❑ Envoi le message à tout le monde via de la diffusion basique.

- ❑ Exécuter **B\_broadcast(m)**.

- ❑ Lors d'un **receive(m, q)** sur  $p$ .

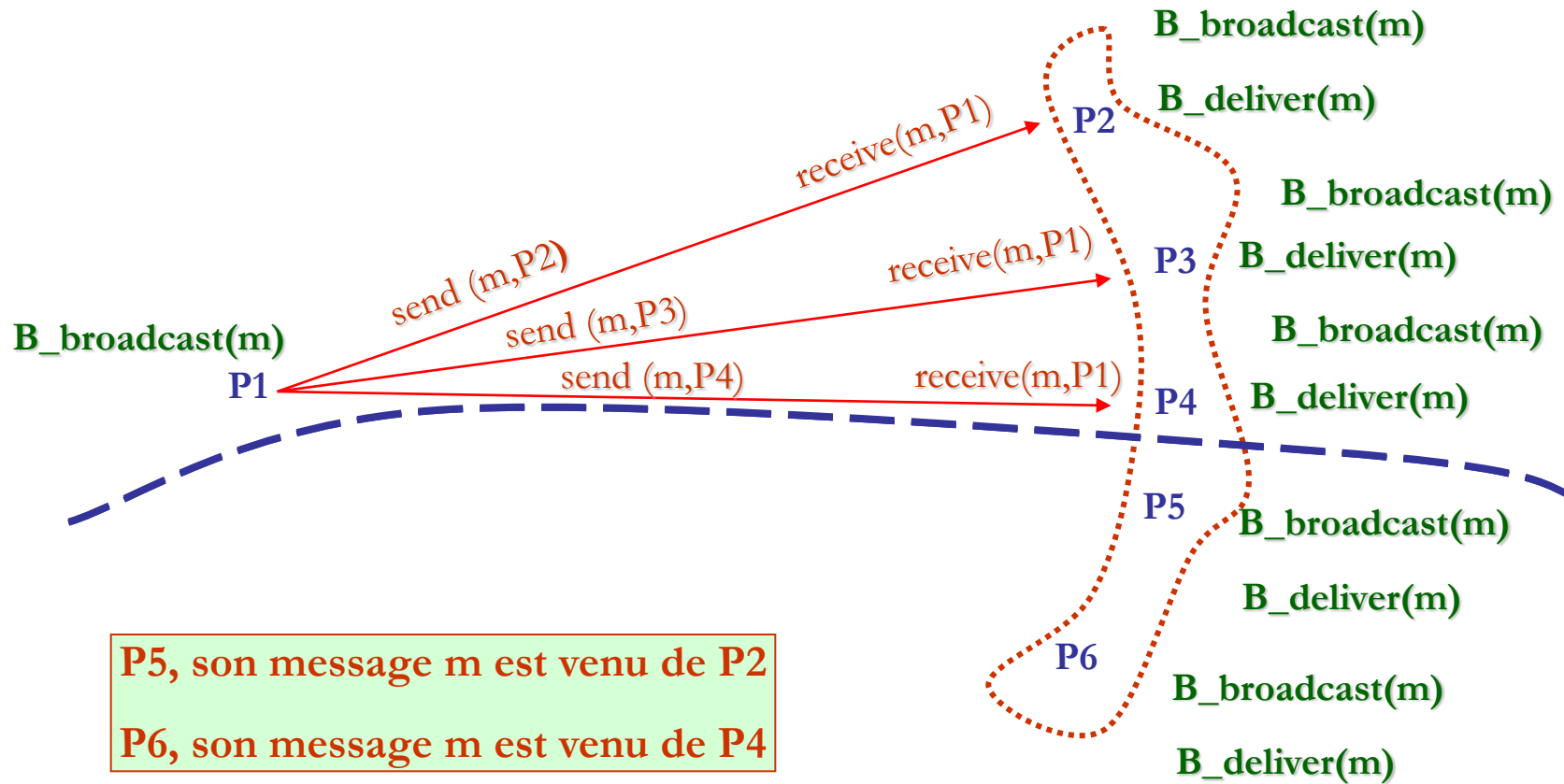
- ❑ Si on avait pas déjà reçu le message  $m$  (**uniquement pour le premier receive**), exécuter :

- ❑ 1. Diffuser le message aux autres processus (sauf pour l'émetteur du message et le processus lui même ) : exécuter **B\_broadcast(m)**.

- ❑ 2. Se délivrer le message : exécuter **R\_deliver(m)**.



# Diffusion fiable



# Diffusion fiable

## ❑ Principes de l'algorithme :

- ❑ A la première réception d'un message, un processus le rediffuse à tous les autres, ce qui permet d'assurer que tout le monde le recevra même si certains plantent pendant un **B\_broadcast(m)**.

## ❑ Inconvénients :

- ❑ Pour  $N$  processus,  $N$  diffusions basiques réalisées, très coûteux en nombre de messages : peu optimal mais fonctionne.
- ❑ Nécessité de vérifier qu'on a pas reçu déjà le message (avec un historique des messages ou des estampilles).

# Diffusion : ordonnancement

- ❑ Diffusion basique ou fiable : ordonnancement.

- ❑ Aucune contrainte ou propriété par défaut sur l'ordre de délivrance des messages sur les processus.

- ❑ Mais des variantes peuvent assurer un certain ordre :

- ❑ FIFO.

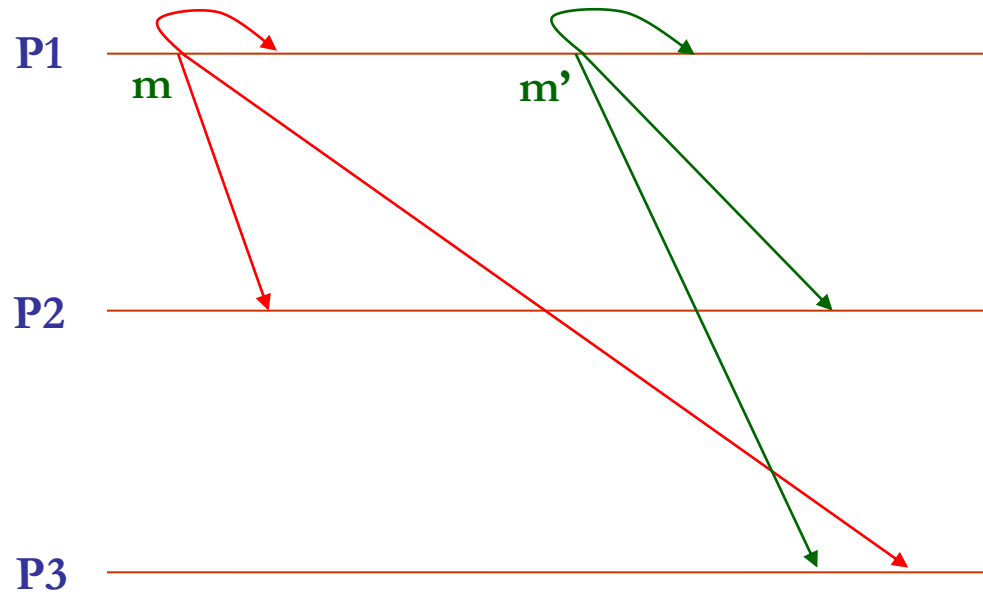
- ❑ Causal.

- ❑ Total.

# Diffusion : ordonnancement

## ❑ **Ordre FIFO :**

- ❑ Si un processus diffuse un message  $m$  puis un message  $m'$ , alors un processus correct qui délivre  $m'$ , délivrera  $m$  avant  $m'$ .

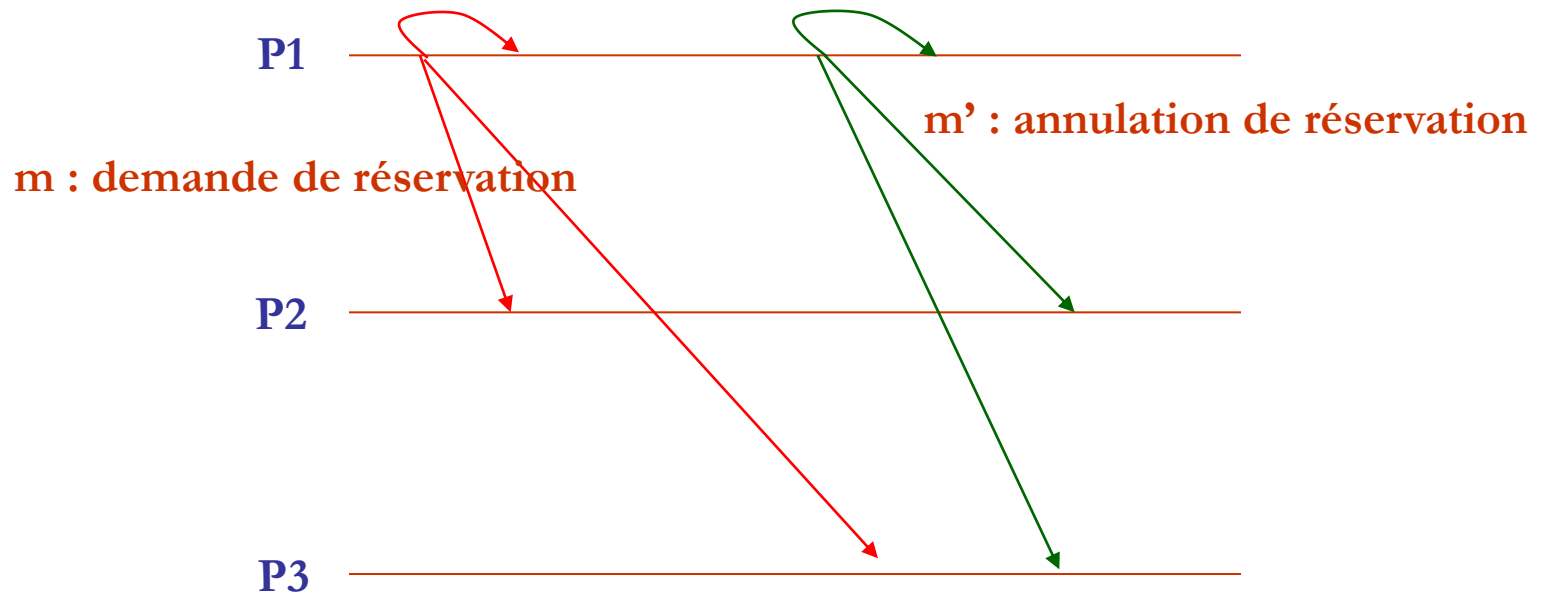


La diffusion est  
faite par un seul  
processus (P1)

Ce n'est pas un ordre FIFO car P3 délivre  $m'$  avant  $m$

# Exemple d'utilisation de la diffusion FIFO

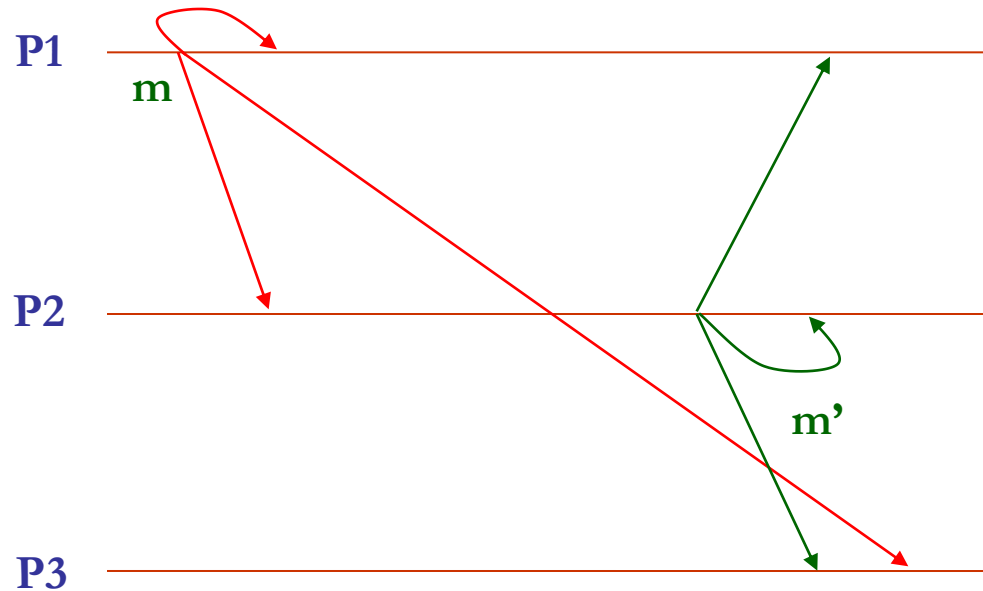
- Dans un système de réservation de places dans un avion, les messages possèdent un contexte (lié à l'émetteur) en dehors duquel le message peut être mal interprété. Par exemple, une annulation de réservation n'a de sens que si le message contenant la demande de réservation est connu. Une telle application demande que **l'algorithme de diffusion respecte l'ordre FIFO**.



# Diffusion : ordonnancement

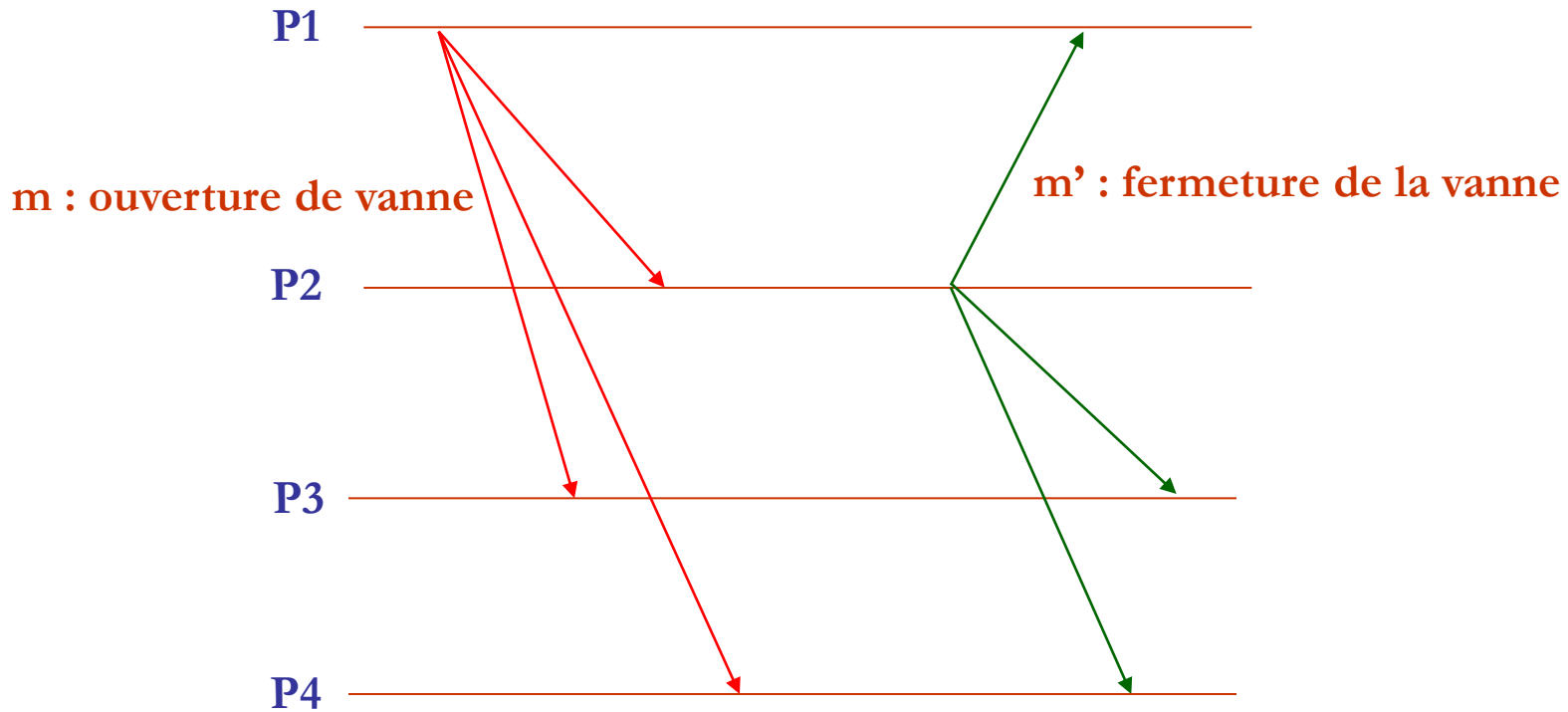
## ❑ **Ordre causal :**

- ❑ Si la diffusion d'un message  $m'$  dépendait causalement de la diffusion d'un message  $m$ , alors un processus correct qui délivre  $m'$  délivrera  $m$  avant  $m'$ .



# Exemple d'utilisation de la diffusion causale

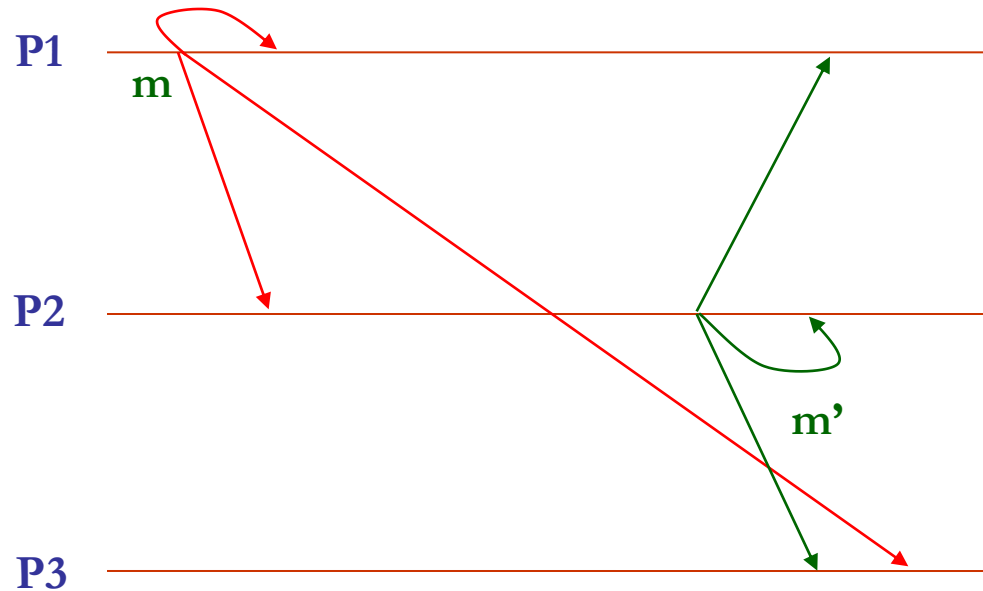
- ❑ P1 envoie à P3 et P4 une commande d'ouverture de vanne, en en rendant compte à P2. Plus tard P2 envoie à P3 et P4 l'ordre de fermeture de la vanne, en en rendant compte à P1.
- ❑ Une telle application demande que **l'algorithme de diffusion respecte l'ordre causale.**



# Diffusion : ordonnancement

## ❑ Ordre total :

- ❑ Si un processus correct délivre  $m$  avant de délivrer  $m'$ , alors tous les autres processus corrects délivreront  $m$  avant  $m'$ .
- ❑ Même ordre de délivrance des messages pour tous les processus corrects.

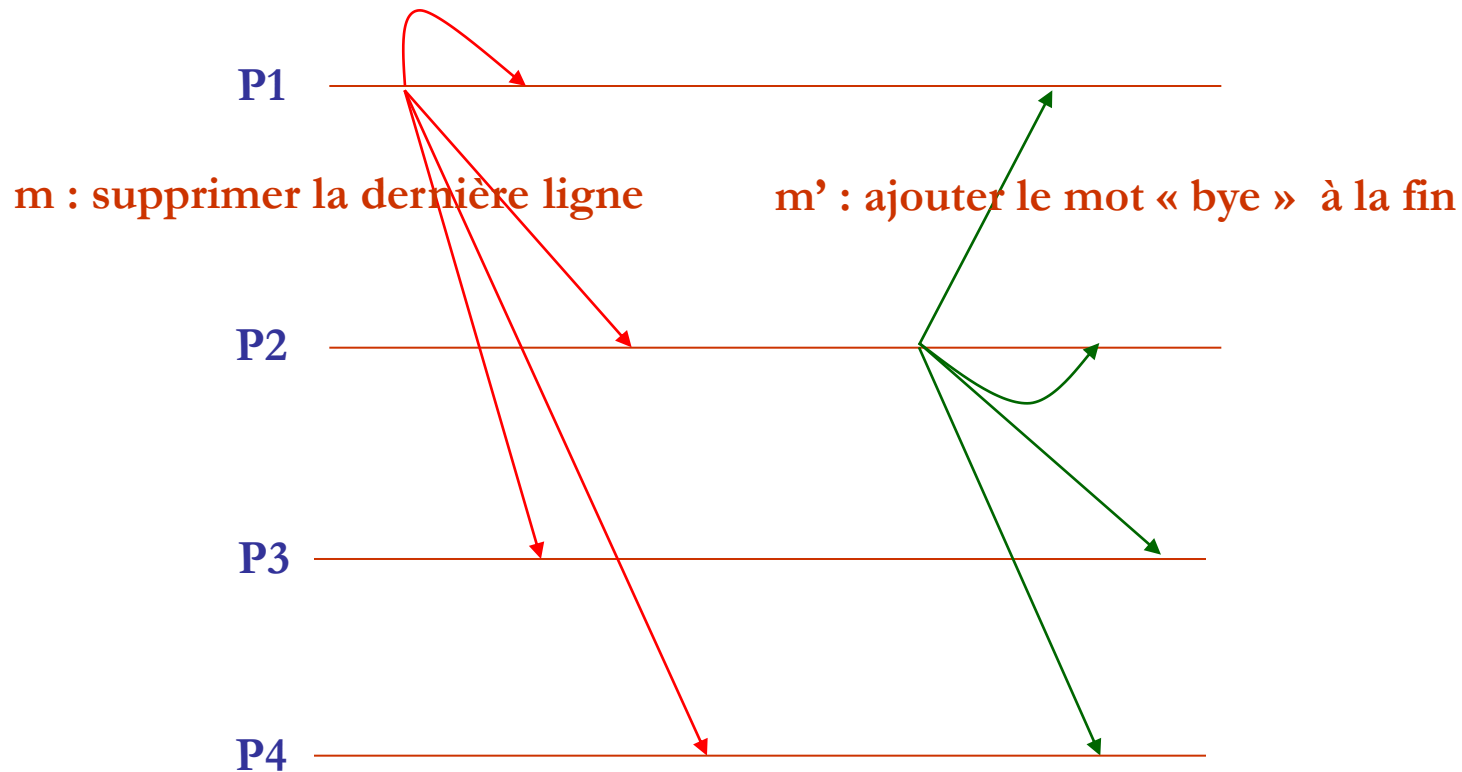


Ce n'est pas un ordre total



# Exemple d'utilisation de la diffusion totale

- ❑ **Copies multiples d'une donnée (ex: fichier texte) :** le fait que toutes les opérations de modifications du fichier en copies multiples soient effectuées dans le même ordre sur toutes les copies suffit à assurer le maintien de la cohérence des copies.



# Diffusion : ordonnancement

## ❑ Relations entre les ordres :

- ❑ L'ordre causal implique l'ordre FIFO.

- ❑ L'ordre FIFO est un ordre causal local : les diffusions lancées localement par le même processus sont en dépendances causales par principe.

- ❑ L'ordre total est indépendant des ordres FIFO ou causal.

## ❑ Combinaisons possibles :

- ❑ Ordre FIFO-total.

- ❑ Ordre causal-total.

# Diffusion : ordonnancement

## ❑ Cas particulier :

❑ Diffusion fiable avec un ordre total : appelé **diffusion atomique**.

❑ Tous les processus reçoivent tous les messages diffusés et tous dans le même ordre.

❑ On ne «mélange» pas les diffusions, une diffusion doit être «terminée» sur tous les processus avant de pouvoir en exécuter une autre.

# Types de diffusion fiable

❑ Au final, 6 combinaisons possibles :

	Pas d'ordre total	Ordre total
Ni FIFO ni causal		
FIFO		
Causal		

# Types de diffusion fiable

❑ Au final, 6 combinaisons possibles :

	Pas d'ordre total	Ordre total
Ni FIFO ni causal	Diffusion fiable	
FIFO		
Causal		

# Types de diffusion fiable

❑ Au final, 6 combinaisons possibles :

	Pas d'ordre total	Ordre total
Ni FIFO ni causal	Diffusion fiable	
FIFO	Diffusion fiable FIFO	
Causal		

# Types de diffusion fiable

❑ Au final, 6 combinaisons possibles :

	Pas d'ordre total	Ordre total
Ni FIFO ni causal	Diffusion fiable	
FIFO	Diffusion fiable FIFO	
Causal	Diffusion fiable causale	

# Types de diffusion fiable

❑ Au final, 6 combinaisons possibles :

	Pas d'ordre total	Ordre total
Ni FIFO ni causal	Diffusion fiable	Diffusion atomique
FIFO	Diffusion fiable FIFO	
Causal	Diffusion fiable causale	



# Types de diffusion fiable

□ Au final, 6 combinaisons possibles :

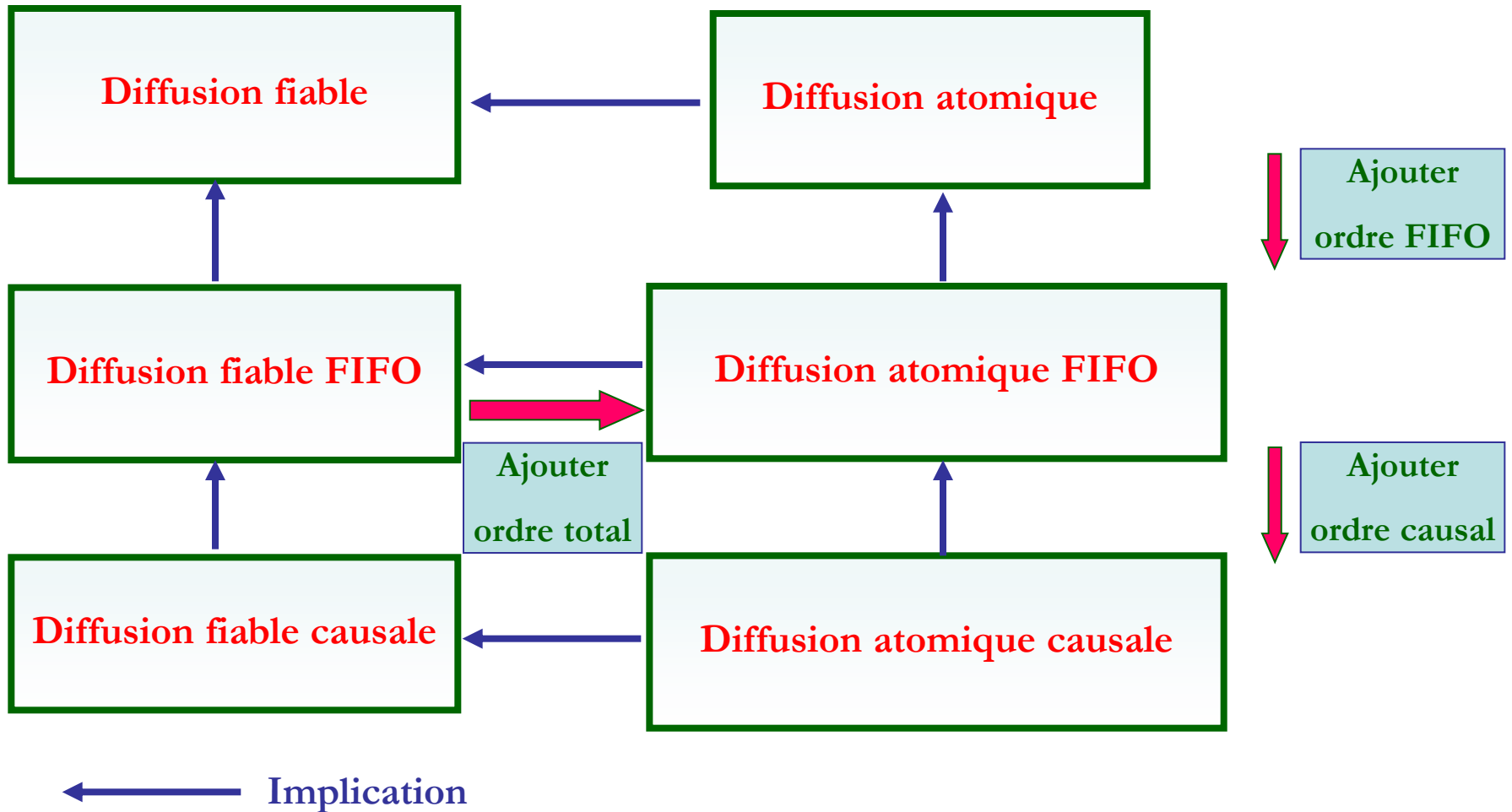
	Pas d'ordre total	Ordre total
Ni FIFO ni causal	Diffusion fiable	Diffusion atomique
FIFO	Diffusion fiable FIFO	Diffusion atomique FIFO
Causal	Diffusion fiable causale	

# Types de diffusion fiable

❑ Au final, 6 combinaisons possibles :

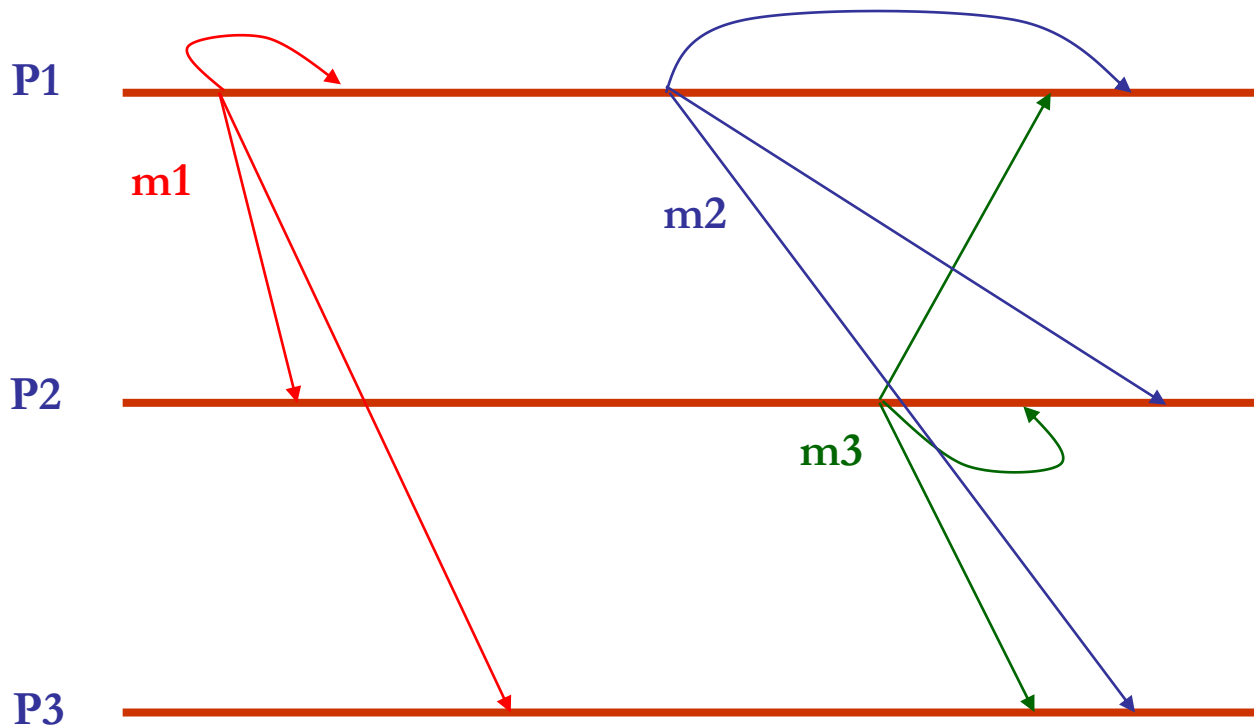
	Pas d'ordre total	Ordre total
Ni FIFO ni causal	Diffusion fiable	Diffusion atomique
FIFO	Diffusion fiable FIFO	Diffusion atomique FIFO
Causal	Diffusion fiable causale	Diffusion atomique causale

# Relations entre les diffusions



# Exercice

❑ Quel le type de cette diffusion ?



❑ Pas de perte de message, donc **diffusion fiable**.

❑ Même ordre de délivrance des messages pour tous les processus corrects : m1, m3, m2 donc **ordre total**.

# Types de diffusion fiable

❑ Au final, 6 combinaisons possibles :

	Pas d'ordre total	Ordre total
Ni FIFO ni causal	Diffusion fiable	Diffusion atomique
FIFO	Diffusion fiable FIFO	Diffusion atomique FIFO
Causal	Diffusion fiable causale	Diffusion atomique causale

# Types de diffusion

□ Au final, 6 combinaisons possibles :

		Ordre total
Ni FIFO ni causal		Diffusion atomique
FIFO		Diffusion atomique FIFO
Causal		Diffusion atomique causale

# Types de diffusion

❑ Au final, 6 combinaisons possibles :

		<b>Ordre total</b>
<b>FIFO</b>		<b>Diffusion atomique FIFO</b>