



# Principes et mécanismes de base des systèmes distribués

Master 2 Informatique - UFR S.A.T

Pr. Ousmane THIARE

othiare@ugb.edu.sn  
[[www.ousmanethiare.com](http://www.ousmanethiare.com)]

16 avril 2020

## Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

# Principes et mécanismes de base des systèmes distribués

# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Le terme de système distribué désigne une collection de processus coopérant dans le but de réaliser une application spécifique. Les fonctions réalisées par ces différents processus peuvent être de différentes natures

- soit de calcul



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Le terme de système distribué désigne une collection de processus coopérant dans le but de réaliser une application spécifique. Les fonctions réalisées par ces différents processus peuvent être de différentes natures

- soit de calcul
- soit de base : exclusion mutuelle, ...



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Le terme de système distribué désigne une collection de processus coopérant dans le but de réaliser une application spécifique. Les fonctions réalisées par ces différents processus peuvent être de différentes natures

- soit de calcul
- soit de base : exclusion mutuelle, ...
- soit de contrôle (liées à la distribution) : détection de la terminaison, cohérence, ...



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Ainsi, les systèmes d'exploitation distribués ne calculent pas de résultats (au sens propre du terme) aux termes desquels ils sont terminés mais ils rendent des services  $\Rightarrow$  ils offrent des moyens aux applications réparties et permettent :

- d'assurer la communication et le partage d'informations entre des applications ;



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Ainsi, les systèmes d'exploitation distribués ne calculent pas de résultats (au sens propre du terme) aux termes desquels ils sont terminés mais ils rendent des services  $\Rightarrow$  ils offrent des moyens aux applications réparties et permettent :

- d'assurer la communication et le partage d'informations entre des applications ;
- d'assurer l'exécution parallèle de programmes sur des processeurs différents ;



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- et de partager des ressources physiques et logiques sur un ensemble d'utilisateur.

d'où la nécessité de mécanismes :





# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- et de partager des ressources physiques et logiques sur un ensemble d'utilisateur.

d'où la nécessité de mécanismes :

- et de partager des ressources physiques et logiques sur un ensemble d'utilisateur.



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- et de partager des ressources physiques et logiques sur un ensemble d'utilisateur.

d'où la nécessité de mécanismes :

- et de partager des ressources physiques et logiques sur un ensemble d'utilisateur.
- de désignation, de transferts et de partages d'informations ;



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- et de partager des ressources physiques et logiques sur un ensemble d'utilisateur.

d'où la nécessité de mécanismes :

- et de partager des ressources physiques et logiques sur un ensemble d'utilisateur.
- de désignation, de transferts et de partages d'informations ;
- de contrôle de la cohérence de ces informations ;



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- et de partager des ressources physiques et logiques sur un ensemble d'utilisateur.

d'où la nécessité de mécanismes :

- et de partager des ressources physiques et logiques sur un ensemble d'utilisateur.
- de désignation, de transferts et de partages d'informations ;
- de contrôle de la cohérence de ces informations ;
- de diffusion d'informations ;



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- et de partager des ressources physiques et logiques sur un ensemble d'utilisateur.

d'où la nécessité de mécanismes :

- et de partager des ressources physiques et logiques sur un ensemble d'utilisateur.
- de désignation, de transferts et de partages d'informations ;
- de contrôle de la cohérence de ces informations ;
- de diffusion d'informations ;
- de synchronisation, de placement de processus et de détection de leur terminaison ;



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- de contrôle global du système et d'élection ;



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- de contrôle global du système et d'élection ;
- d'ordonnancement de processus ;



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- de contrôle global du système et d'élection ;
- d'ordonnancement de processus ;
- d'exclusion mutuelle ;





# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- de contrôle global du système et d'élection ;
- d'ordonnancement de processus ;
- d'exclusion mutuelle ;
- de reprise sur panne ;



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- de contrôle global du système et d'élection ;
- d'ordonnancement de processus ;
- d'exclusion mutuelle ;
- de reprise sur panne ;
- de transaction ;



# Introduction

## Rôles d'un système distribué

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- de contrôle global du système et d'élection ;
- d'ordonnancement de processus ;
- d'exclusion mutuelle ;
- de reprise sur panne ;
- de transaction ;
- d'administration...



# Introduction

## Rôles d'un système distribué

### Introduction

#### Éléments de base des systèmes distribués

#### Techniques d'implémentation des protocoles

⇒ plusieurs modèles dont les deux principaux sont le modèle serveur/clients et le modèle à objet.

Il est à noter que l'architecture sous-jacente est une donnée fondamentale. En effet en fonction du fait qu'elle soit fortement couplée ou faiblement couplée la problématique n'est pas la même.

- Dans le premier cas, la communication est faite via une mémoire commune : les problèmes sont plus simples à résoudre. Ainsi, l'exclusion mutuelle peut être simplement résolu par un blocage du bus comme en mono-processeur.



# Introduction

## Rôles d'un système distribué

### Introduction

#### Éléments de base des systèmes distribués

#### Techniques d'implémentation des protocoles

⇒ plusieurs modèles dont les deux principaux sont le modèle serveur/clients et le modèle à objet.

Il est à noter que l'architecture sous-jacente est une donnée fondamentale. En effet en fonction du fait qu'elle soit fortement couplée ou faiblement couplée la problématique n'est pas la même.

- Dans le premier cas, la communication est faite via une mémoire commune : les problèmes sont plus simples à résoudre. Ainsi, l'exclusion mutuelle peut être simplement résolu par un blocage du bus comme en mono-processeur.
- Dans le deuxième cas, les communications se font via des messages avec tous les problèmes liés aux réseaux (perte d'informations , ...)



# Introduction

## Caractéristiques des systèmes distribués

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

**État global** Une des caractéristiques essentielles d'un tel système distribué est l'absence d'état global perceptible à un instant donné par un observateur.

- les processus ne "connaissent" que les événements qu'ils ont générés ;

⇒ l'état des processus à un instant donné n'est pas suffisant.



# Introduction

## Caractéristiques des systèmes distribués

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

**État global** Une des caractéristiques essentielles d'un tel système distribué est l'absence d'état global perceptible à un instant donné par un observateur.

- les processus ne "connaissent" que les événements qu'ils ont générés ;
- les processus ne connaissent que les messages qu'il ont reçus ou envoyés => Ils ne connaissent pas "l'état" du réseau (ce qui y circule)

⇒ l'état des processus à un instant donné n'est pas suffisant.



# Introduction

## Caractéristiques des systèmes distribués

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

**État global** Une des caractéristiques essentielles d'un tel système distribué est l'absence d'état global perceptible à un instant donné par un observateur.

- les processus ne "connaissent" que les événements qu'ils ont générés ;
  - les processus ne connaissent que les messages qu'il ont reçus ou envoyés => Ils ne connaissent pas "l'état" du réseau (ce qui y circule)
  - il n'y a pas de relation naturelle d'ordre strict sur des événements ayant lieu sur des machines différentes.
- ⇒ l'état des processus à un instant donné n'est pas suffisant.





# Introduction

## Caractéristiques des systèmes distribués

### Introduction

### Éléments de base des systèmes distribués

### Techniques d'implémentation des protocoles

## Distribution des données Deux cas :

- distribution inhérente aux problèmes : le concepteur se voit imposé, par exemple, une distribution géographique (disques de Bases de données sur plusieurs sites, ...)

d'où

Bien évidemment, les deux peuvent se faire en même temps.



# Introduction

## Caractéristiques des systèmes distribués

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

### Distribution des données Deux cas :

- distribution inhérente aux problèmes : le concepteur se voit imposé, par exemple, une distribution géographique (disques de Bases de données sur plusieurs sites, ...)
- distribution inhérente à la mise en oeuvre : les utilisateurs veulent pouvoir travailler sur des sites distants.

d'où

Bien évidemment, les deux peuvent se faire en même temps.



# Introduction

## Caractéristiques des systèmes distribués

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

### Distribution des données Deux cas :

- distribution inhérente aux problèmes : le concepteur se voit imposé, par exemple, une distribution géographique (disques de Bases de données sur plusieurs sites, ...)
- distribution inhérente à la mise en oeuvre : les utilisateurs veulent pouvoir travailler sur des sites distants.

d'où

- duplication : une donnée peut se trouver dupliquée sur plusieurs sites => problème de cohérence

Bien évidemment, les deux peuvent se faire en même temps.



# Introduction

## Caractéristiques des systèmes distribués

### Introduction

### Éléments de base des systèmes distribués

### Techniques d'implémentation des protocoles

## Distribution des données Deux cas :

- distribution inhérente aux problèmes : le concepteur se voit imposé, par exemple, une distribution géographique (disques de Bases de données sur plusieurs sites, ...)
- distribution inhérente à la mise en oeuvre : les utilisateurs veulent pouvoir travailler sur des sites distants.

d'où

- duplication : une donnée peut se trouver dupliquée sur plusieurs sites => problème de cohérence
- partitionnement : une grande BD peut être couplée en plusieurs sous-BD sur des sites différents  $\implies$  problème de désignation (comment retrouver une donnée de la base : il faut d'abord savoir sur quel site elle se trouve).

Bien évidemment, les deux peuvent se faire en même temps.



# Introduction

## Caractéristiques des systèmes distribués

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

**Contrôle** On dira qu'il y a *contrôle distribué* lorsqu'il n'y a pas de relation hiérarchique statique entre les processus : il n'y a pas de processus qui joue de rôle particulier a priori. Ainsi, il n'y a pas de maître qui assure en permanence le contrôle global.

Pourquoi ?

- tolérance aux fautes faible voire nulle : si ce processus s'arrête  $\implies$  fin du système ;

Il se peut que pour certaines fonctions du système, il y ait nécessité d'un maître : on utilisera alors un mécanisme d'élection.



# Introduction

## Caractéristiques des systèmes distribués

### Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

**Contrôle** On dira qu'il y a *contrôle distribué* lorsqu'il n'y a pas de relation hiérarchique statique entre les processus : il n'y a pas de processus qui joue de rôle particulier a priori. Ainsi, il n'y a pas de maître qui assure en permanence le contrôle global.

Pourquoi ?

- tolérance aux fautes faible voire nulle : si ce processus s'arrête  $\implies$  fin du système ;
- goulot d'étranglement.

Il se peut que pour certaines fonctions du système, il y ait nécessité d'un maître : on utilisera alors un mécanisme d'élection.



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Un protocole définit le comportement d'un processus vis-à-vis d'autres processus : on va s'intéresser principalement (et presque uniquement) aux réactions du processus à la réalisation d'événements : réception d'un message, réalisation d'une condition, émission d'un message, etc.

Par exemple, on peut définir (succinctement) la réservation d'un imprimante par le protocole suivant :

Pour un processus  $P_i$

- Etat initial : Libre



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Un protocole définit le comportement d'un processus vis-à-vis d'autres processus : on va s'intéresser principalement (et presque uniquement) aux réactions du processus à la réalisation d'événements : réception d'un message, réalisation d'une condition, émission d'un message, etc.

Par exemple, on peut définir (succinctement) la réservation d'un imprimante par le protocole suivant :

Pour un processus  $P_i$

- Etat initial : Libre
- Pour utiliser l'imprimante, passer à l'état Demande





# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Un protocole définit le comportement d'un processus vis-à-vis d'autres processus : on va s'intéresser principalement (et presque uniquement) aux réactions du processus à la réalisation d'événements : réception d'un message, réalisation d'une condition, émission d'un message, etc.

Par exemple, on peut définir (succinctement) la réservation d'un imprimante par le protocole suivant :

Pour un processus  $P_i$

- Etat initial : Libre
- Pour utiliser l'imprimante, passer à l'état Demande
  - Demande :



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Un protocole définit le comportement d'un processus vis-à-vis d'autres processus : on va s'intéresser principalement (et presque uniquement) aux réactions du processus à la réalisation d'événements : réception d'un message, réalisation d'une condition, émission d'un message, etc.

Par exemple, on peut définir (succinctement) la réservation d'un imprimante par le protocole suivant :

Pour un processus  $P_i$

- Etat initial : Libre
- Pour utiliser l'imprimante, passer à l'état Demande
  - Demande :
  - émettre la demande vers tous les autres processus (\*\*\*)



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Un protocole définit le comportement d'un processus vis-à-vis d'autres processus : on va s'intéresser principalement (et presque uniquement) aux réactions du processus à la réalisation d'événements : réception d'un message, réalisation d'une condition, émission d'un message, etc.

Par exemple, on peut définir (succinctement) la réservation d'un imprimante par le protocole suivant :

Pour un processus  $P_i$

- Etat initial : Libre
- Pour utiliser l'imprimante, passer à l'état Demande
  - Demande :
  - émettre la demande vers tous les autres processus (\*\*\*)
  - mettre un compteur à zéro (\*)



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Un protocole définit le comportement d'un processus vis-à-vis d'autres processus : on va s'intéresser principalement (et presque uniquement) aux réactions du processus à la réalisation d'événements : réception d'un message, réalisation d'une condition, émission d'un message, etc.

Par exemple, on peut définir (succinctement) la réservation d'un imprimante par le protocole suivant :

Pour un processus  $P_i$

- Etat initial : Libre
- Pour utiliser l'imprimante, passer à l'état Demande
  - Demande :
  - émettre la demande vers tous les autres processus (\*\*\*)
  - mettre un compteur à zéro (\*)
  - passer à l'état Attente



# Éléments de base des systèmes distribués

## Protocole

Pour un processus  $P_i$

- Attente :

Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

Pour un processus  $P_i$

- Attente :
  - A la réception d'un message "OK" (\*\*)



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

Pour un processus  $P_i$

- Attente :
  - A la réception d'un message "OK" (\*\*)
  - incrémenter compteur



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Pour un processus  $P_i$

- Attente :
  - A la réception d'un message "OK" (\*\*)
  - incrémenter compteur
  - si compteur==N, passer à l'état Utilisation





# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Pour un processus  $P_i$

- Attente :
  - A la réception d'un message "OK" (\*\*)
  - incrémenter compteur
  - si compteur==N, passer à l'état Utilisation
- Utilisation :



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Pour un processus  $P_i$

- Attente :
  - A la réception d'un message "OK" (\*\*)
  - incrémenter compteur
  - si compteur==N, passer à l'état Utilisation
- Utilisation :
  - lancer impression (\*)



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Pour un processus  $P_i$

- Attente :
  - A la réception d'un message "OK" (\*\*)
  - incrémenter compteur
  - si compteur==N, passer à l'état Utilisation
- Utilisation :
  - lancer impression (\*)
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ . (\*)



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Pour un processus  $P_i$

- Attente :
  - A la réception d'un message "OK" (\*\*)
  - incrémenter compteur
  - si compteur==N, passer à l'état Utilisation
- Utilisation :
  - lancer impression (\*)
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ . (\*)
  - A la fin d'impression, passer à l'état Liberation



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Pour un processus  $P_i$

- Attente :
  - A la réception d'un message "OK" (\*\*)
  - incrémenter compteur
  - si compteur==N, passer à l'état Utilisation
- Utilisation :
  - lancer impression (\*)
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ . (\*)
  - A la fin d'impression, passer à l'état Libération
- Libération :



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Pour un processus  $P_i$

- Attente :
  - A la réception d'un message "OK" (\*\*)
  - incrémenter compteur
  - si compteur==N, passer à l'état Utilisation
- Utilisation :
  - lancer impression (\*)
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ . (\*)
  - A la fin d'impression, passer à l'état Libération
- Libération :
  - envoyer "OK" à tous les  $P_j$  tels que  $j$  ait été mémorisé



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Pour un processus  $P_i$

- Attente :
  - A la réception d'un message "OK" (\*\*)
  - incrémenter compteur
  - si compteur==N, passer à l'état Utilisation
- Utilisation :
  - lancer impression (\*)
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ . (\*)
  - A la fin d'impression, passer à l'état Libération
- Libération :
  - envoyer "OK" à tous les  $P_j$  tels que  $j$  ait été mémorisé
  - effacer les  $j$  mémorisés



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Pour un processus  $P_i$

- Attente :
  - A la réception d'un message "OK" (\*\*)
  - incrémenter compteur
  - si compteur==N, passer à l'état Utilisation
- Utilisation :
  - lancer impression (\*)
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ . (\*)
  - A la fin d'impression, passer à l'état Libération
- Libération :
  - envoyer "OK" à tous les  $P_j$  tels que  $j$  ait été mémorisé
  - effacer les  $j$  mémorisés
  - passer à l'état Libre





# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Pour un processus  $P_i$

- Attente :
  - A la réception d'un message "OK" (\*\*)
  - incrémenter compteur
  - si compteur==N, passer à l'état Utilisation
- Utilisation :
  - lancer impression (\*)
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ . (\*)
  - A la fin d'impression, passer à l'état Liberation
- Libération :
  - envoyer "OK" à tous les  $P_j$  tels que  $j$  ait été mémorisé
  - effacer les  $j$  mémorisés
  - passer à l'état Libre
- Libre :



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Pour un processus  $P_i$

- Attente :
  - A la réception d'un message "OK" (\*\*)
  - incrémenter compteur
  - si compteur==N, passer à l'état Utilisation
- Utilisation :
  - lancer impression (\*)
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ . (\*)
  - A la fin d'impression, passer à l'état Liberation
- Libération :
  - envoyer "OK" à tous les  $P_j$  tels que  $j$  ait été mémorisé
  - effacer les  $j$  mémorisés
  - passer à l'état Libre
- Libre :
  - sur réception de "Demande" venant de  $P_j$ , envoyer "OK" à  $P_j$



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- (\*) On ne s'intéresse que très peu à ce qui se passe en local. En effet, on suppose que localement, on dispose de ressources logicielles pour réaliser les opérations. La seule contrainte est que le processus dispose de toutes les données nécessaires aux calculs.



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

- (\*) On ne s'intéresse que très peu à ce qui se passe en local. En effet, on suppose que localement, on dispose de ressources logicielles pour réaliser les opérations. La seule contrainte est que le processus dispose de toutes les données nécessaires aux calculs.
- (\*\*): On considère que l'envoi et la réception de message est réalisé par le système d'exploitation



# Éléments de base des systèmes distribués

## Protocole

Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

- (\*) On ne s'intéresse que très peu à ce qui se passe en local. En effet, on suppose que localement, on dispose de ressources logicielles pour réaliser les opérations. La seule contrainte est que le processus dispose de toutes les données nécessaires aux calculs.
- (\*\*): On considère que l'envoi et la réception de message est réalisé par le système d'exploitation
- (\*\*\*) : nous verrons plus tard ce que cela veut vraiment dire



# Éléments de base des systèmes distribués

## Processus

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Afin de pouvoir participer à plusieurs protocoles simultanément, il est nécessaire que soit "associé" à chaque processus de calcul un ensemble de processus de contrôle. Ainsi, un processus de calcul doit pouvoir être "géré" dans les exclusions mutuelles, doit pouvoir accéder aux données partagées .

Il doit donc répondre à tout moment à des requêtes concernant différents protocoles en cours.

Un site est l'ensemble formé :

- du (ou des) processus de calcul et



# Éléments de base des systèmes distribués

## Processus

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Afin de pouvoir participer à plusieurs protocoles simultanément, il est nécessaire que soit "associé" à chaque processus de calcul un ensemble de processus de contrôle. Ainsi, un processus de calcul doit pouvoir être "géré" dans les exclusions mutuelles, doit pouvoir accéder aux données partagées .

Il doit donc répondre à tout moment à des requêtes concernant différents protocoles en cours.

Un site est l'ensemble formé :

- du (ou des) processus de calcul et
- des processus assurant les protocoles de contrôles internes au système d'exploitation local et ceux spécialement mis en place pour l'application.



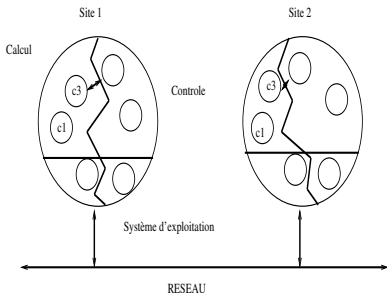
# Éléments de base des systèmes distribués

## Processus

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles



De fait, il nous arrivera de "confondre" site et processus : on parlera indifféremment de  $S_i$  et  $P_i$  en considérant que chaque processus de calcul est inclus dans un site différent.





Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

### Propriétés structurelles

- maillage en anneau : processus ne connaît directement que son suivant (anneau unidirectionnel) ou ces deux voisins immédiats (anneau bi-directionnel).  
 $P_{i \neq N}$  ne connaît que  $P_{i+1}$  et  $P_N$  ne connaît que  $P_1$ .



### Propriétés structurelles

- maillage en anneau : processus ne connaît directement que son suivant (anneau unidirectionnel) ou ces deux voisins immédiats (anneau bi-directionnel).  
 $P_{i \neq N}$  ne connaît que  $P_{i+1}$  et  $P_N$  ne connaît que  $P_1$ .
- maillage en étoile : il existe un processus particulier  $P_{i \neq 1}$  ne connaît que  $P_1$  et  $P_1$  connaît tous les autres sites.



### Propriétés structurelles

- maillage en anneau : processus ne connaît directement que son suivant (anneau unidirectionnel) ou ces deux voisins immédiats (anneau bi-directionnel).  
 $P_{i \neq N}$  ne connaît que  $P_{i+1}$  et  $P_N$  ne connaît que  $P_1$ .
- maillage en étoile : il existe un processus particulier  $P_{i \neq 1}$  ne connaît que  $P_1$  et  $P_1$  connaît tous les autres sites.
- maillage en arbre :  
 $P_1$  ne connaît que ses fils ( $P_1$  est la racine de l'arbre) ;  
Si  $P_{i \neq 1}$  a des fils, alors  $P_{i \neq 1}$  ne connaît que ses fils et son père ;  
Si  $P_{i \neq 1}$  n'a pas de fils alors :  $P_{i \neq 1}$  ne connaît que son père (processus ou terminal).



Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

### Propriétés structurelles

- maillage complet : type Internet  $\implies$  tous les processus se connaissent.

La connaissance de cette topologie est fondamentale lorsqu'on définit des algorithmes distribués. On peut être amené à rajouter une "couche" sur un réseau X pour simuler un maillage Y.

Ainsi, l'anneau peut n'être que virtuel  $\implies$  il ne faut donc pas confondre réseau physique et maillage logique.



Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

### Propriétés comportementales

Elles sont diverses. Les plus fréquemment rencontrées sont :

- **H1** : la transmission sur la voie se fait sans duplication de messages ;



Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

### Propriétés comportementales

Elles sont diverses. Les plus fréquemment rencontrées sont :

- **H1** : la transmission sur la voie se fait sans duplication de messages ;
- **H2** : la transmission se fait sans altération des messages ;



### Propriétés comportementales

Elles sont diverses. Les plus fréquemment rencontrées sont :

- **H1** : la transmission sur la voie se fait sans duplication de messages ;
- **H2** : la transmission se fait sans altération des messages ;
- **H3** : pour tout couple de processus, l'ordre de réception des messages est identique à l'ordre d'émission (pas de déséquencelement) ;



### Propriétés comportementales

Elles sont diverses. Les plus fréquemment rencontrées sont :

- **H1** : la transmission sur la voie se fait sans duplication de messages ;
- **H2** : la transmission se fait sans altération des messages ;
- **H3** : pour tout couple de processus, l'ordre de réception des messages est identique à l'ordre d'émission (pas de déséquence) ;
- **H4** : le délai d'acheminement des messages est fini : tout message envoyé est reçu au bout d'un temps fini ;





### Propriétés comportementales

Elles sont diverses. Les plus fréquemment rencontrées sont :

- **H1** : la transmission sur la voie se fait sans duplication de messages ;
- **H2** : la transmission se fait sans altération des messages ;
- **H3** : pour tout couple de processus, l'ordre de réception des messages est identique à l'ordre d'émission (pas de déséquence) ;
- **H4** : le délai d'acheminement des messages est fini : tout message envoyé est reçu au bout d'un temps fini ;
- **H5** : le délai d'acheminement est borné : si un message n'est pas reçu après X secondes, il est perdu.



# Éléments de base des systèmes distribués

## Liaisons

Introduction

Éléments de base des systèmes distribués

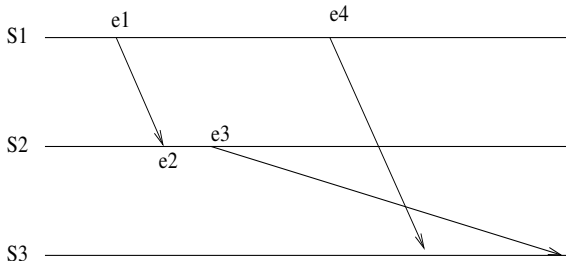
Techniques d'implémentation des protocoles

### Propriétés comportementales

Dans le cas où le protocole de communication garantit les hypothèses **H1**, **H2**, **H3** et **H4**, on dira que les canaux de communication sont **FIFO**.

**Attention** tous les canaux peuvent être **FIFO** et pourtant l'"ordre" de réception des messages sur les différents sites peut être indéterminé.

**Exemple :**



# Éléments de base des systèmes distribués

## Liaisons

Introduction

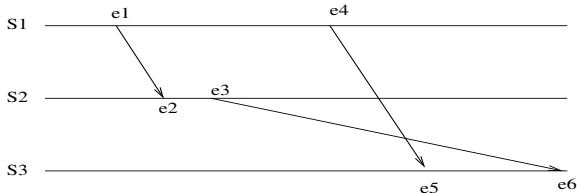
Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

### Propriétés comportementales

**Attention** tous les canaux peuvent être **FIFO** et pourtant l'"ordre" de réception des messages sur les différents sites peut être indéterminé.

**Exemple :**



On ne peut déterminer l'ordre  $e_5 e_6$  :  $e_6$  peut très bien avoir lieu avant  $e_5$ .



# Éléments de base des systèmes distribués

## Liaisons

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

### Propriétés comportementales

**Rappel :** Il n'y a pas d'horloge globale  $\implies$  ce n'est pas parce que sur le dessin un événement à l'air d'avoir lieu avant un autre que c'est vrai. Il faut partir du principe que deux événements indépendants sur deux sites ne peuvent être ordonnés dans le temps grâce à une horloge "physique" globale.

**ATTENTION** un algorithme de système distribué est *toujours* donné (validé) en fonction d'hypothèses sur le comportement des canaux. Ainsi, fréquemment, les algorithmes sont donnés pour des canaux **FIFO**. Les faire tourner sur des canaux non **FIFO**, ne garantit plus leur fonctionnement correcte. Donc, lorsque vous définissez un algorithme distribué, il faudra *toujours* exprimer *clairement* les hypothèses de comportement des canaux (en gros, moins il y aura d'hypothèses, "meilleur" sera



Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

## Trafic

Toutes propriétés identiques et pour une même fonction réalisée, un algorithme distribué est d'autant plus intéressant qu'il est performant (évident). Or le plus souvent, ce qui ralentit le plus, c'est le réseau. Il faut donc, étudier précisément quels sont les coûts réseau (nombre de messages échangés principalement) engendrés par l'algorithme que vous êtes en train de définir.



# Éléments de base des systèmes distribués

## Trafic

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

### Robustesse d'un protocole par rapport à un réseau

Soit  $p$  la probabilité de perte d'un message et  $e$  le nombre d'échanges nécessaires pour réaliser le protocole.

**a)** Calculons le taux d'échecs d'un protocole en fonction de  $p$  et de  $e$ .

Posons  $X$  le nombre de pertes lors d'exécution du protocole.

La probabilité  $P(X = k)$  suit une loi binomiale  $B(k, p)$ , d'où

$$P(X = k) = \binom{k}{e} p^k (1 - p)^{e-k}$$

Calculons la probabilité  $p_t$  pour que le protocole ne "perde" pas de message. Cela correspond au cas où  $X = 0$ .

$$\text{Or } p_t = P(X = 0) = \binom{0}{e} p^0 (1 - p)^{e-0} = (1 - p)^e.$$



# Éléments de base des systèmes distribués

## Trafic

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

### Robustesse d'un protocole par rapport à un réseau

**Exemple** : pour  $p = \frac{1}{1000}$  et  $e = 1000$ , la probabilité qu'aucun message ne se perde est de  $(0.999)^{1000} = 0.368$ . Le protocole se plante environ 2 fois sur 3. Avec  $p = \frac{1}{100}$  et  $e = 1000$ , la probabilité qu'aucun message ne se perde est de  $(0.99)^{1000} = 0.00004$  : le protocole ne marche jamais !!

**b)** Voyons combien de fois il faut relancer le protocole pour qu'il y ait  $x\%$  de chance de se terminer sans perte. Soit  $T$  le nombre de tentatives. Soit  $p_t$  la probabilité qu'aucun message ne se perde lors d'une tentative (par exemple, pour  $p = \frac{1}{1000}$  et  $e = 1000$ ,  $p_t = 0.368$ ). D'où, si on veut que le protocole se termine correctement avec une probabilité d'au moins  $x\%$ , il faut que la probabilité qu'il se plante aux  $T$  tentatives soit inférieure à  $(1 - x)$ .



Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

### Robustesse d'un protocole par rapport à un réseau

Or la probabilité que le protocole se plante à une tentative

est égale à  $(1 - p_t)^T < (1 - x)$  d'où

$\exp(T \cdot \log(p_t)) < (1 - x)$  d'où  $T \cdot \ln(1 - p_t) < \ln(1 - x)$

d'où  $T > \frac{\ln(1-x)}{\ln(p_t)}$ .

**Exemple** : pour  $x = 95\%$  et  $p_t = 0.368$ , on obtient

$-0.195T < -1.301$  d'où  $T > 6.6$ . D'où avec 7 tentatives, le protocole à 95% de chances de réussir (en fait 96%).

Pour une fiabilité à 99%, on a  $\ln(1 - 0.99) = -4.6052$  d'où  $T = 10$ .





Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Trois grandes familles de protocoles :

- calcul diffusant



# Techniques d'implémentation des protocoles

Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

Trois grandes familles de protocoles :

- calcul diffusant
- jeton circulant



# Techniques d'implémentation des protocoles

Introduction

Éléments de  
base des  
systèmes  
distribués

**Techniques  
d'implémenta-  
tion des  
protocoles**

Trois grandes familles de protocoles :

- calcul diffusant
- jeton circulant
- estampillage



# Calcul diffusant

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Les processus peuvent être connectés de manière quelconque. Initialement un seul processus (dit processus racine) peut émettre des messages et ensuite tout autre processus ne peut en émettre que s'il en a reçu lui-même (les demandes de calcul vont se diffuser) On utilise ce type de calcul principalement lorsque le graphe de communication entre les processus à la structure d'arbre. Le principe de calcul est alors le suivant :

- Init : le processus **racine** émet un message de demande de calcul vers chacun de ses successeurs dans l'arbre



# Calcul diffusant

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Les processus peuvent être connectés de manière quelconque. Initialement un seul processus (dit processus racine) peut émettre des messages et ensuite tout autre processus ne peut en émettre que s'il en a reçu lui-même (les demandes de calcul vont se diffuser) On utilise ce type de calcul principalement lorsque le graphe de communication entre les processus à la structure d'arbre. Le principe de calcul est alors le suivant :

- Init : le processus **racine** émet un message de demande de calcul vers chacun de ses successeurs dans l'arbre
- Lorsqu'un processus reçoit un message de demande de calcul, il le ré-expédie vers chacun de ses successeurs



# Calcul diffusant

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Les processus peuvent être connectés de manière quelconque. Initialement un seul processus (dit processus racine) peut émettre des messages et ensuite tout autre processus ne peut en émettre que s'il en a reçu lui-même (les demandes de calcul vont se diffuser) On utilise ce type de calcul principalement lorsque le graphe de communication entre les processus à la structure d'arbre. Le principe de calcul est alors le suivant :

- Init : le processus **racine** émet un message de demande de calcul vers chacun de ses successeurs dans l'arbre
- Lorsqu'un processus reçoit un message de demande de calcul, il le ré-expédie vers chacun de ses successeurs
  - s'il en a



# Calcul diffusant

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Les processus peuvent être connectés de manière quelconque. Initialement un seul processus (dit processus racine) peut émettre des messages et ensuite tout autre processus ne peut en émettre que s'il en a reçu lui-même (les demandes de calcul vont se diffuser) On utilise ce type de calcul principalement lorsque le graphe de communication entre les processus à la structure d'arbre. Le principe de calcul est alors le suivant :

- Init : le processus **racine** émet un message de demande de calcul vers chacun de ses successeurs dans l'arbre
- Lorsqu'un processus reçoit un message de demande de calcul, il le ré-expédie vers chacun de ses successeurs
  - s'il en a
  - sinon il fait son calcul et retourne la réponse vers son père



# Calcul diffusant

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- Lorsqu'un processus reçoit une réponse d'un de ses fils, il la traite ou la mémorise suivant le type de calcul. S'il a reçu une réponse de tous ses fils, il fait son calcul et retourne la réponse vers son père.





# Calcul diffusant

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- Lorsqu'un processus reçoit une réponse d'un de ses fils, il la traite ou la mémorise suivant le type de calcul. S'il a reçu une réponse de tous ses fils, il fait son calcul et retourne la réponse vers son père.
- Lorsque la racine a reçu toutes les réponses, il fait son calcul : le calcul diffusant est terminé.



# Jeton circulant

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

La technique consiste à faire circuler un privilège dans un ensemble de processus connectés en réseau (virtuel ou non).

- Etat initial : Libre, jeton\_present=Faux sauf pour  $P_0$ , util=Faux.



# Jeton circulant

## Introduction

## Éléments de base des systèmes distribués

## Techniques d'implémentation des protocoles

La technique consiste à faire circuler un privilège dans un ensemble de processus connectés en réseau (virtuel ou non).

- Etat initial : Libre, jeton\_present=Faux sauf pour  $P_0$ , util=Faux.
- Pour utiliser l'imprimante, passer à l'état Demande



# Jeton circulant

## Introduction

## Éléments de base des systèmes distribués

## Techniques d'implémentation des protocoles

La technique consiste à faire circuler un privilège dans un ensemble de processus connectés en réseau (virtuel ou non).

- Etat initial : Libre, jeton\_present=Faux sauf pour  $P_0$ , util=Faux.
- Pour utiliser l'imprimante, passer à l'état Demande
- Demande

Sinon



# Jeton circulant

## Introduction

## Éléments de base des systèmes distribués

## Techniques d'implémentation des protocoles

La technique consiste à faire circuler un privilège dans un ensemble de processus connectés en réseau (virtuel ou non).

- Etat initial : Libre,  $\text{jeton\_present} = \text{Faux}$  sauf pour  $P_0$ ,  $\text{util} = \text{Faux}$ .
- Pour utiliser l'imprimante, passer à l'état Demande
- Demande
  - Si  $\text{jeton\_present} = \text{Vrai}$  alors

Sinon



# Jeton circulant

## Introduction

## Éléments de base des systèmes distribués

## Techniques d'implémentation des protocoles

La technique consiste à faire circuler un privilège dans un ensemble de processus connectés en réseau (virtuel ou non).

- Etat initial : Libre,  $\text{jeton\_present} = \text{Faux}$  sauf pour  $P_0$ ,  $\text{util} = \text{Faux}$ .
- Pour utiliser l'imprimante, passer à l'état Demande
- Demande
  - Si  $\text{jeton\_present} = \text{Vrai}$  alors
  - $\text{util} = \text{vrai}$

Sinon



# Jeton circulant

## Introduction

## Éléments de base des systèmes distribués

## Techniques d'implémentation des protocoles

La technique consiste à faire circuler un privilège dans un ensemble de processus connectés en réseau (virtuel ou non).

- Etat initial : Libre, jeton\_present=Faux sauf pour  $P_0$ , util=Faux.
- Pour utiliser l'imprimante, passer à l'état Demande
- Demande
  - Si jeton\_present==Vrai alors
  - util=vrai
  - passer à l'état Utilisation
- Sinon



# Jeton circulant

## Introduction

## Éléments de base des systèmes distribués

## Techniques d'implémentation des protocoles

La technique consiste à faire circuler un privilège dans un ensemble de processus connectés en réseau (virtuel ou non).

- Etat initial : Libre, jeton\_present=Faux sauf pour  $P_0$ , util=Faux.
- Pour utiliser l'imprimante, passer à l'état Demande
- Demande
  - Si jeton\_present==Vrai alors
  - util=vrai
  - passer à l'état Utilisation
- Sinon
  - émettre la demande vers tous les autres processus





# Jeton circulant

## Introduction

## Éléments de base des systèmes distribués

## Techniques d'implémentation des protocoles

La technique consiste à faire circuler un privilège dans un ensemble de processus connectés en réseau (virtuel ou non).

- Etat initial : Libre, jeton\_present=Faux sauf pour  $P_0$ , util=Faux.
- Pour utiliser l'imprimante, passer à l'état Demande
- Demande
  - Si jeton\_present==Vrai alors
  - util=vrai
  - passer à l'état Utilisation

Sinon

- émettre la demande vers tous les autres processus
- passer à l'état Attente



# Jeton circulant

Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

## ■ Attente



# Jeton circulant

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- Attente
  - à la réception du message "Jeton"



# Jeton circulant

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- Attente
  - à la réception du message "Jeton"
  - jeton\_present=Vrai



# Jeton circulant

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

## ■ Attente

- à la réception du message "Jeton"
- `jeton_present=Vrai`
- `util=Vrai`



# Jeton circulant

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- Attente
  - à la réception du message "Jeton"
  - jeton\_present=Vrai
  - util=Vrai
  - passer à l'état Utilisation



# Jeton circulant

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- **Attente**
  - à la réception du message "Jeton"
  - jeton\_present=Vrai
  - util=Vrai
  - passer à l'état Utilisation
- **Utilisation**



# Jeton circulant

## Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- **Attente**
  - à la réception du message "Jeton"
  - jeton\_present=Vrai
  - util=Vrai
  - passer à l'état Utilisation
- **Utilisation**
  - Lancer impression





# Jeton circulant

## Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- **Attente**
  - à la réception du message "Jeton"
  - `jeton_present=Vrai`
  - `util=Vrai`
  - passer à l'état Utilisation
- **Utilisation**
  - Lancer impression
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ .



# Jeton circulant

## Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- **Attente**
  - à la réception du message "Jeton"
  - jeton\_present=Vrai
  - util=Vrai
  - passer à l'état Utilisation
- **Utilisation**
  - Lancer impression
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ .
  - A la fin d'impression, passer à l'état Libération



# Jeton circulant

## Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- **Attente**
  - à la réception du message "Jeton"
  - `jeton_present=Vrai`
  - `util=Vrai`
  - passer à l'état Utilisation
- **Utilisation**
  - Lancer impression
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ .
  - A la fin d'impression, passer à l'état Libération
- **Libération**



# Jeton circulant

## Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- **Attente**
  - à la réception du message "Jeton"
  - jeton\_present=Vrai
  - util=Vrai
  - passer à l'état Utilisation
- **Utilisation**
  - Lancer impression
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ .
  - A la fin d'impression, passer à l'état Libération
- **Libération**
  - util=Faux



# Jeton circulant

## Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- **Attente**
  - à la réception du message "Jeton"
  - jeton\_present=Vrai
  - util=Vrai
  - passer à l'état Utilisation
- **Utilisation**
  - Lancer impression
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ .
  - A la fin d'impression, passer à l'état Libération
- **Libération**
  - util=Faux
  - Si il existe un  $j$  mémorisé alors



# Jeton circulant

## Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- **Attente**
  - à la réception du message "Jeton"
  - jeton\_present=Vrai
  - util=Vrai
  - passer à l'état Utilisation
- **Utilisation**
  - Lancer impression
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ .
  - A la fin d'impression, passer à l'état Libération
- **Libération**
  - util=Faux
  - Si il existe un  $j$  mémorisé alors
  - jeton\_present=Faux



# Jeton circulant

## Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

- **Attente**
  - à la réception du message "Jeton"
  - jeton\_present=Vrai
  - util=Vrai
  - passer à l'état Utilisation
- **Utilisation**
  - Lancer impression
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ .
  - A la fin d'impression, passer à l'état Libération
- **Libération**
  - util=Faux
  - Si il existe un  $j$  mémorisé alors
  - jeton\_present=Faux
  - envoyer "Jeton" à  $P_j$



# Jeton circulant

## Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles



- **Attente**
  - à la réception du message "Jeton"
  - jeton\_present=Vrai
  - util=Vrai
  - passer à l'état Utilisation
- **Utilisation**
  - Lancer impression
  - Sur réception de "Demande" venant de  $P_j$ , mémoriser  $j$ .
  - A la fin d'impression, passer à l'état Libération
- **Libération**
  - util=Faux
  - Si il existe un  $j$  mémorisé alors
  - jeton\_present=Faux
  - envoyer "Jeton" à  $P_j$
  - passer à l'état Libre



# Jeton circulant

Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

## ■ Libre



# Jeton circulant

Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

- Libre
  - Sur réception de "Demande" venant de  $P_j$  :



# Jeton circulant

Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

- Libre
  - Sur réception de "Demande" venant de  $P_j$  :
  - Si `jeton_present==Vrai` alors



Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

- Libre
  - Sur réception de "Demande" venant de  $P_j$  :
  - Si jeton\_present==Vrai alors
  - jeton\_present=Faux



- Libre
  - Sur réception de "Demande" venant de  $P_j$  :
  - Si jeton\_present==Vrai alors
  - jeton\_present=Faux
  - envoyer "Jeton" à  $P_j$



# Estampillage

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Un des problèmes principaux des systèmes distribués est l'absence d'horloge réelle globale alors que l'ordre dans lequel surviennent les événements est primordial dans bien des cas. Ainsi :

pour une entrée dans une section critique, on ne peut, par exemple, prendre comme critère d'autorisation la date de la demande : une machine dont la date à 1 heure d'avance sur les autres machines sera fortement avantagée ;

- de même, la date d'arrivée de réception d'un message peut du coup être inférieure à la date d'émission !



# Estampillage

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Un des problèmes principaux des systèmes distribués est l'absence d'horloge réelle globale alors que l'ordre dans lequel surviennent les événements est primordial dans bien des cas. Ainsi :

pour une entrée dans une section critique, on ne peut, par exemple, prendre comme critère d'autorisation la date de la demande : une machine dont la date à 1 heure d'avance sur les autres machines sera fortement avantagée ;

- de même, la date d'arrivée de réception d'un message peut du coup être inférieure à la date d'émission !
- lorsqu'une donnée est modifiée sur deux sites, laquelle de ces modifications est la plus récente et donc la dernière ?



# Estampillage

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Un des problèmes principaux des systèmes distribués est l'absence d'horloge réelle globale alors que l'ordre dans lequel surviennent les événements est primordial dans bien des cas. Ainsi :

pour une entrée dans une section critique, on ne peut, par exemple, prendre comme critère d'autorisation la date de la demande : une machine dont la date à 1 heure d'avance sur les autres machines sera fortement avantagée ;

- de même, la date d'arrivée de réception d'un message peut du coup être inférieure à la date d'émission !
- lorsqu'une donnée est modifiée sur deux sites, laquelle de ces modifications est la plus récente et donc la dernière ?
- ....





Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

En fait, comme on le verra, dans presque tous les cas, si on peut ordonner les événements liés à un problème, celui-ci est résolu !! Les techniques basées sur l'estampillage vont donc chercher, grâce à des échanges de messages à dater et ordonner ces événements. On est donc amené à définir des **horloges logiques** qui vont permettre de **dater** tous les événements se produisant dans le système et ce selon une relation d'ordre total ou partiel.



# Horloges logiques

Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

Le principe est le suivant :

- si, sur un même site, l'événement  $e_i$  précède (a lieu physiquement avant)  $e_j$  alors la date de  $e_i$  doit être inférieure à celle de  $e_j$  :  $D(e_i) < D(e_j)$  ;



# Horloges logiques

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

Le principe est le suivant :

- si, sur un même site, l'événement  $e_i$  précède (a lieu physiquement avant)  $e_j$  alors la date de  $e_i$  doit être inférieure à celle de  $e_j$  :  $D(e_i) < D(e_j)$  ;
- lors de l'envoi d'un message  $m$ , la date de l'événement  $e_{emission(m)}$  correspondant à l'envoi doit être inférieure à celle de l'événement  $e_{reception(m)}$  correspondant à la réception de ce message  
 $D(e_{emission(m)}) < D(e_{reception(m)})$  ;



# Horloges logiques

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

## Algorithme de Lamport (1978)

Sur chaque site  $S_i$ , on trouve une variable entière  $H_i$  dite horloge locale. La date locale d'un événement  $E$  est notée  $d(E)$ .

⇒ Pour chaque événement  $E$  ne correspondant pas ni à l'envoi, ni à la réception d'un message,  $S_i$  incrémente  $H_i$  et date cet événement par  $d(E) = H_i$

⇒ Lors de l'émission d'un message  $M$  par  $S_i$ ,  $S_i$  incrémente  $H_i$ , estampille le message  $M$  par  $(H_i, i)$  et date l'émission par  $d(E) = H_i$

⇒ Lors de la réception d'un message estampillé  $(H_j, j)$  par  $S_i$ ,  $S_i$  recale son horloge de la manière suivante :

$$\text{si } H_i < H_j \text{ alors } H_i = H_j + 1 \text{ sinon } H_i = H_i + 1$$

Puis  $S_i$  date l'événement  $E$  de réception par  $d(E) = H_i$



## Algorithme de Lamport (1978)

La date globale  $D(E)$  d'un événement  $E$  est alors  $(d(E), i)$  où  $i$  est le numéro du site où a eu lieu l'événement et  $d(E)$  sa date locale sur ce site.

On peut aisément montrer que la relation **précède** (notée  $\rightarrow$ ) sur les dates globales définie par :

$$((d(e_1), i_1) \Rightarrow (d(e_2), i_2))$$



$$\left\{ \begin{array}{l} d(e_1) < d(e_2) \\ \text{ou} \\ (d(e_1) = d(e_2) \wedge (i_1 < i_2)) \end{array} \right.$$



# Horloges logiques

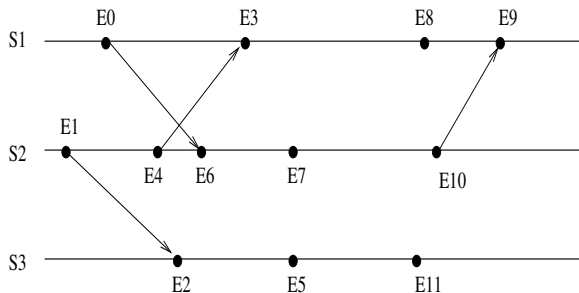
Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

est bien un ordre total strict ( $A \implies B$  est vrai, si la date globale de  $A <$  date globale de  $B$ ).

Le datage des événements dans l'exemple suivant, dans un réseau FIFO :



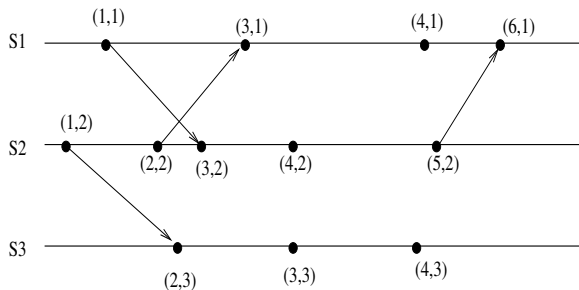
# Horloges logiques

## Introduction

## Éléments de base des systèmes distribués

## Techniques d'implémentation des protocoles

où les flèches représentent des envois de message, nous donne :



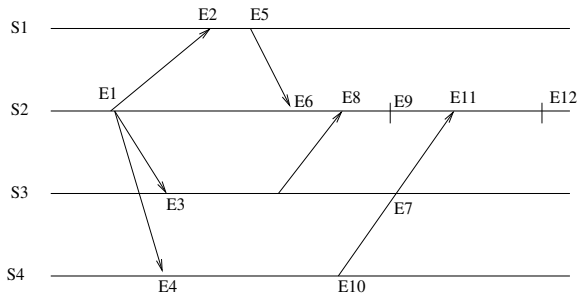
# Horloges logiques

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

**Problème** La relation  $\implies$  ordonne bien les événements mais elle perd la causalité c'est-à-dire l'information qu'un ou plusieurs événements ont **potentiellement** été la cause d'un autre est bien mise en évidence mais par contre le fait que deux événements sont indépendants (ou *concurrents*) :  $e_i || e_j \Leftrightarrow \neg(e_j \rightarrow e_i)$  et  $\neg(e_i \rightarrow e_j)$  est totalement perdu. Ainsi l'exemple suivant :





# Horloges logiques

Introduction

Éléments de  
base des  
systèmes  
distribués

Techniques  
d'implémenta-  
tion des  
protocoles

l'algorithme précédent nous donne :

$$\begin{aligned} D(E_1) &= (1, 2) & D(E_2) &= (2, 1) & D(E_3) &= (2, 3) \\ D(E_4) &= (2, 4) & D(E_5) &= (3, 1) & D(E_6) &= (4, 2) & D(E_7) &= \\ & (3, 3) & D(E_8) &= (5, 2) & D(E_9) &= (6, 2) & D(E_{10}) &= \\ & (3, 4) & D(E_{11}) &= (7, 2) & D(E_{12}) &= (8, 2) \end{aligned}$$

et on voit que :  $E_2 \implies E_3 \implies E_4$  mais rien ne nous dit qu'ils sont en fait indépendants.

De même

$$\left\{ \begin{array}{l} E_2 \implies E_9 \\ E_3 \implies E_9 \\ E_4 \implies E_9 \end{array} \right.$$

d'où le nouvel algorithme dit des **horloges vectorielles**.



# Horloges vectorielles

## Introduction

## Éléments de base des systèmes distribués

## Techniques d'implémentation des protocoles

Sur chaque site  $S_i, i = 1, \dots, n$ , on définit une horloge vectorielle comme un vecteur  $V_i[1 \dots n]$  initialisé à 0. A chaque événement E,  $V_i[i]$  est incrémenté.

- si l'événement E correspond à l'envoi d'un message M par un site  $S_i$ , M est estampillé par  $V_m =$  valeur de l'horloge  $V_i$  au moment de l'envoi

Dans tous les cas, E est, ensuite daté par  $V_i$ .



# Horloges vectorielles

## Introduction

## Éléments de base des systèmes distribués

## Techniques d'implémentation des protocoles

Sur chaque site  $S_i, i = 1, \dots, n$ , on définit une horloge vectorielle comme un vecteur  $V_i[1 \dots n]$  initialisé à 0. A chaque événement E,  $V_i[i]$  est incrémenté.

- si l'événement E correspond à l'envoi d'un message M par un site  $S_i$ , 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 d'un message  $(M, V_m)$  par  $S_i$ , celui-ci exécute :  
 $V_i[k] = \max(V_i[k], V_m[k])$  pour  $k = 1, \dots, n$ .

Dans tous les cas, E est, ensuite daté par  $V_i$ .



# Horloges vectorielles

Dans l'exemple précédent cela nous donne :

E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11
[0,0,0,0]	[1,0,0,0]			[2,1,0,0]						
	[1,0,0,0]			M(2,1,0,0)						
[0,1,0,0]					[0,2,0,0]		[2,3,0,0]	[2,4,2,0]		[2,5,2,0]
M(0,1,0,0)					[2,2,0,0]		[2,3,2,0]			[2,5,2,2]
		[0,0,1,0]				[0,1,2,0]				
		[0,1,1,0]				M(0,1,2,0)				
			[0,0,0,1]						[0,1,0,2]	
			[0,1,0,0]						M(0,1,0,2)	



# Horloges vectorielles et relation de causalité

## Introduction

### Éléments de base des systèmes distribués

### Techniques d'implémentation des protocoles

On remarque que pour un événement  $E$ , sur un site  $S_i$  de date  $V_E$ ,  $V_E[i]$  est le nombre d'événements sur  $S_i$  ayant précédé  $E$  ( $E$  compris) et que  $\sum_{k=0}^n V_E[k]$  est le nombre total d'événements ayant précédé  $E$ . Pour deux horloges vectorielles  $V_1$  et  $V_2$ , on définit la relation d'ordre suivante :

$$\left\{ \begin{array}{l} V_1 \geq V_2 \Leftrightarrow V_1[i] \geq V_2[i] \\ V_1 \leq V_2 \Leftrightarrow V_1[i] \leq V_2[i] \\ V_1 \parallel V_2 \Leftrightarrow \neg(V_1 \geq V_2) \wedge \neg(V_1 \leq V_2) \end{array} \right.$$



# Horloges vectorielles et relation de causalité

Introduction

Éléments de base des systèmes distribués

Techniques d'implémentation des protocoles

La dernière "condition" peut se ré-écrire en  $V_1 \parallel V_2 \Leftrightarrow \exists k_1, k_2, (V[k_1] \leq V[k_2]) \wedge (V[k_1] > V[k_2])$  Pour deux événements  $E_1$  et  $E_2 : E_1 \longrightarrow E_2 \Leftrightarrow V_{E_1} \leq V_{E_2}$ .

Dans l'exemple :

$E_1 \longrightarrow E_2? \dashrightarrow V_{E_1} = [0, 1, 0, 0] \leq V_{E_2} = [1, 1, 0, 0]$  d'où  $E_1$  est bien une cause de  $E_2$ .

$E_2 \longrightarrow E_3? \dashrightarrow V_{E_2} = [1, 1, 0, 0]$  et  $V_{E_3} = [0, 1, 1, 0]$  d'où  $V_{E_2}[2] \leq V_{E_3}[2]$  mais  $V_{E_2}[1] > V_{E_3}[1]$  d'où  $V_{E_2} \parallel V_{E_3}$  d'où  $eE_2$  et  $E_2$  sont indépendants.

