INFORMATIQUE
Sciences
Université Paris Cité

Algorithmique Répartie

Introduction

M2

Lélia Blin

lelia.blin@irif.fr

Quelle algorithmique connaissez vous?

Algorithmique déjà vu

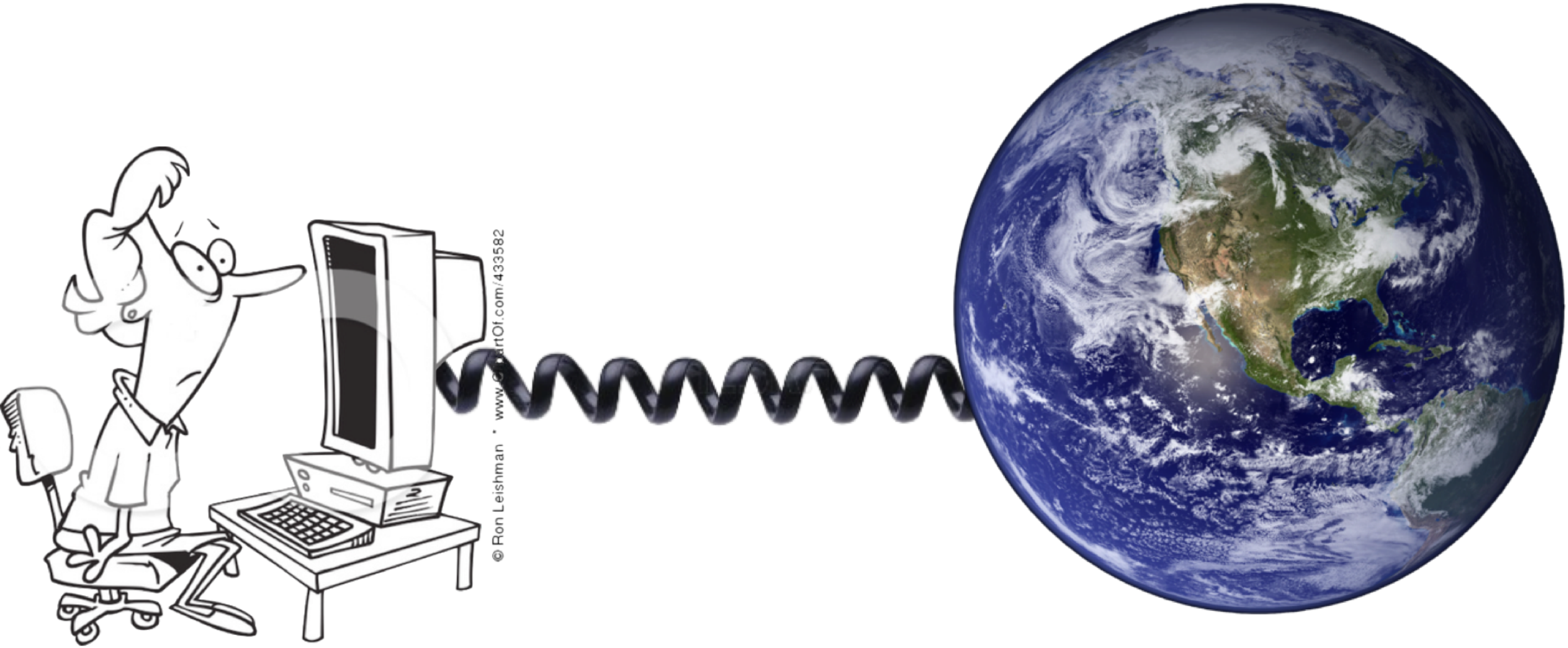
- Sur un ordinateur
  - une entité de calcul
- Une tâche (un calcul)
- Série d'opérations élémentaires
- Les une après les autres
  - séquentielle

Quelle algorithmique connaissez vous?

L'algorithmique séquentielle

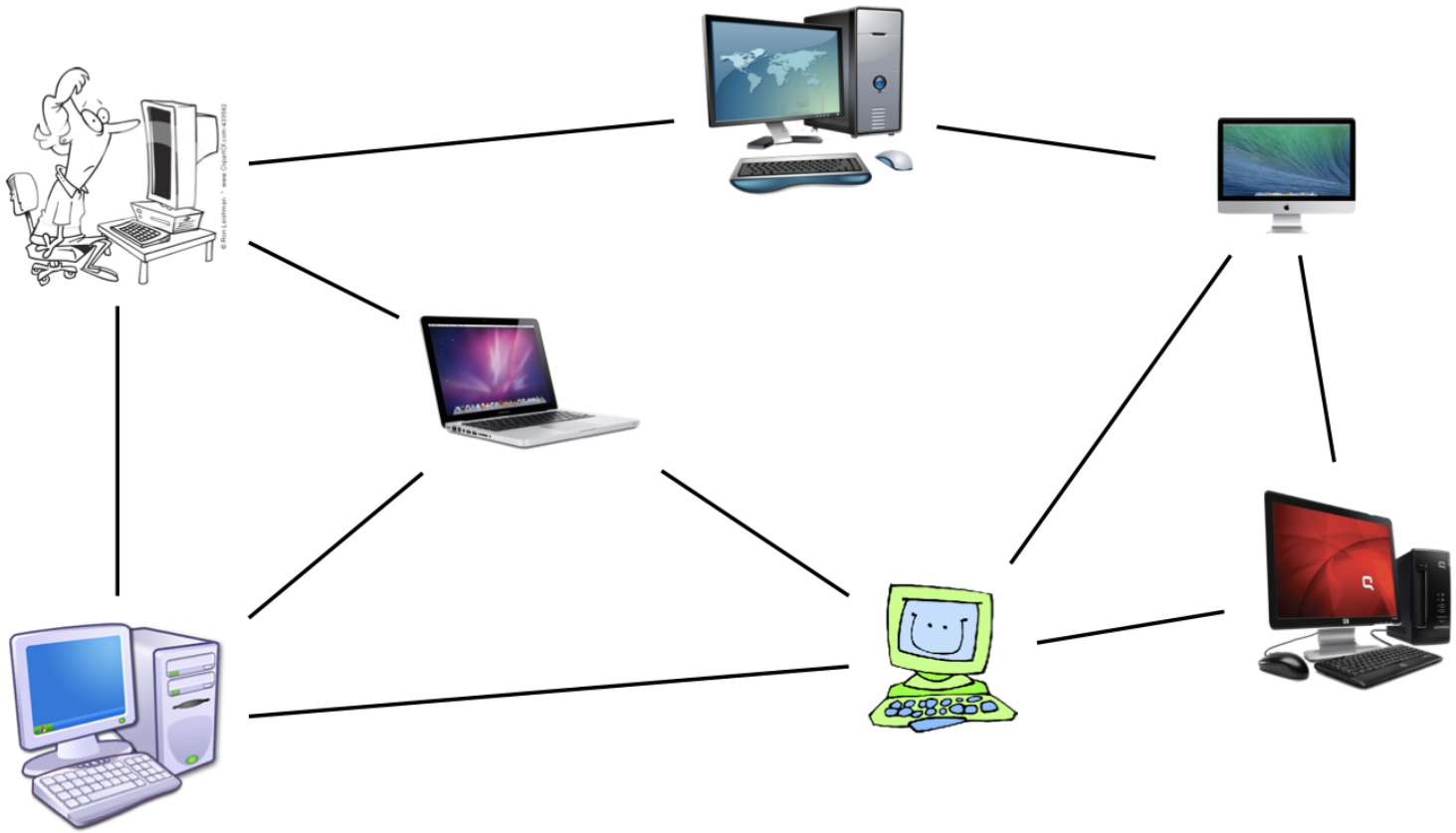

© Ron Leishman * www.ClipartOf.com/433582

Lélia Blin



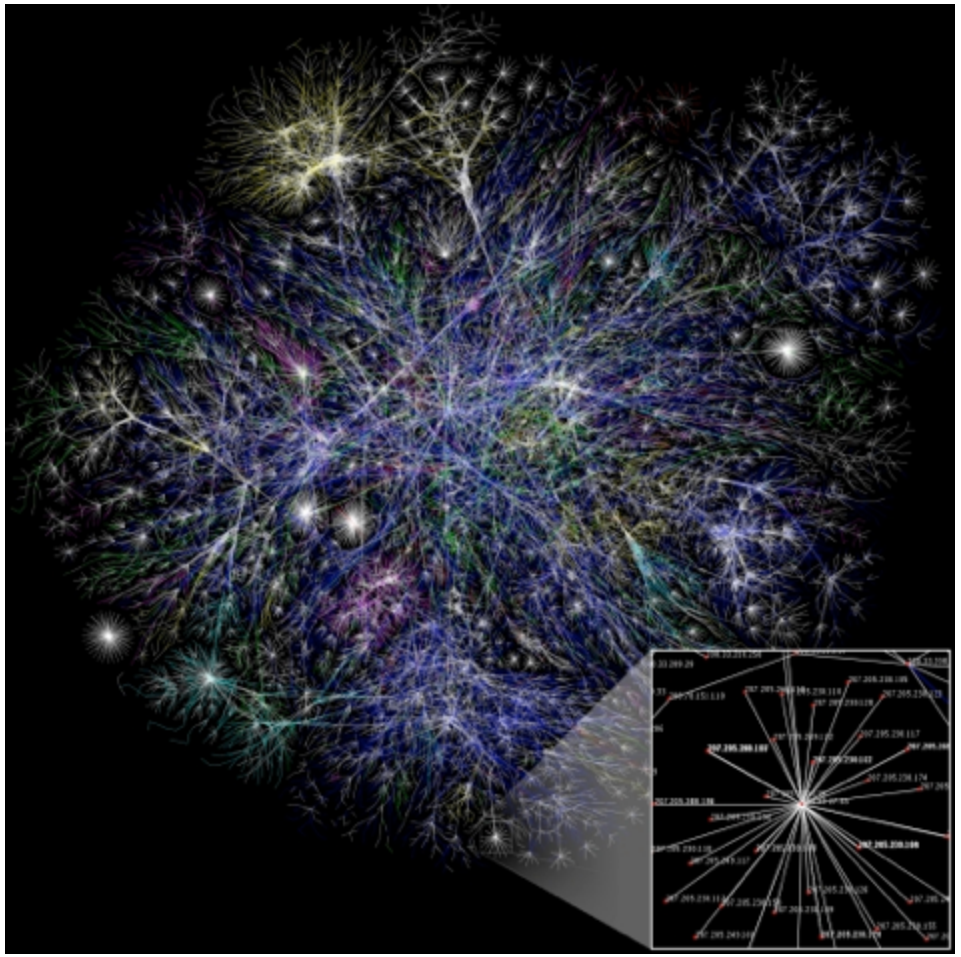
Lélia Blin

Réseau de communication



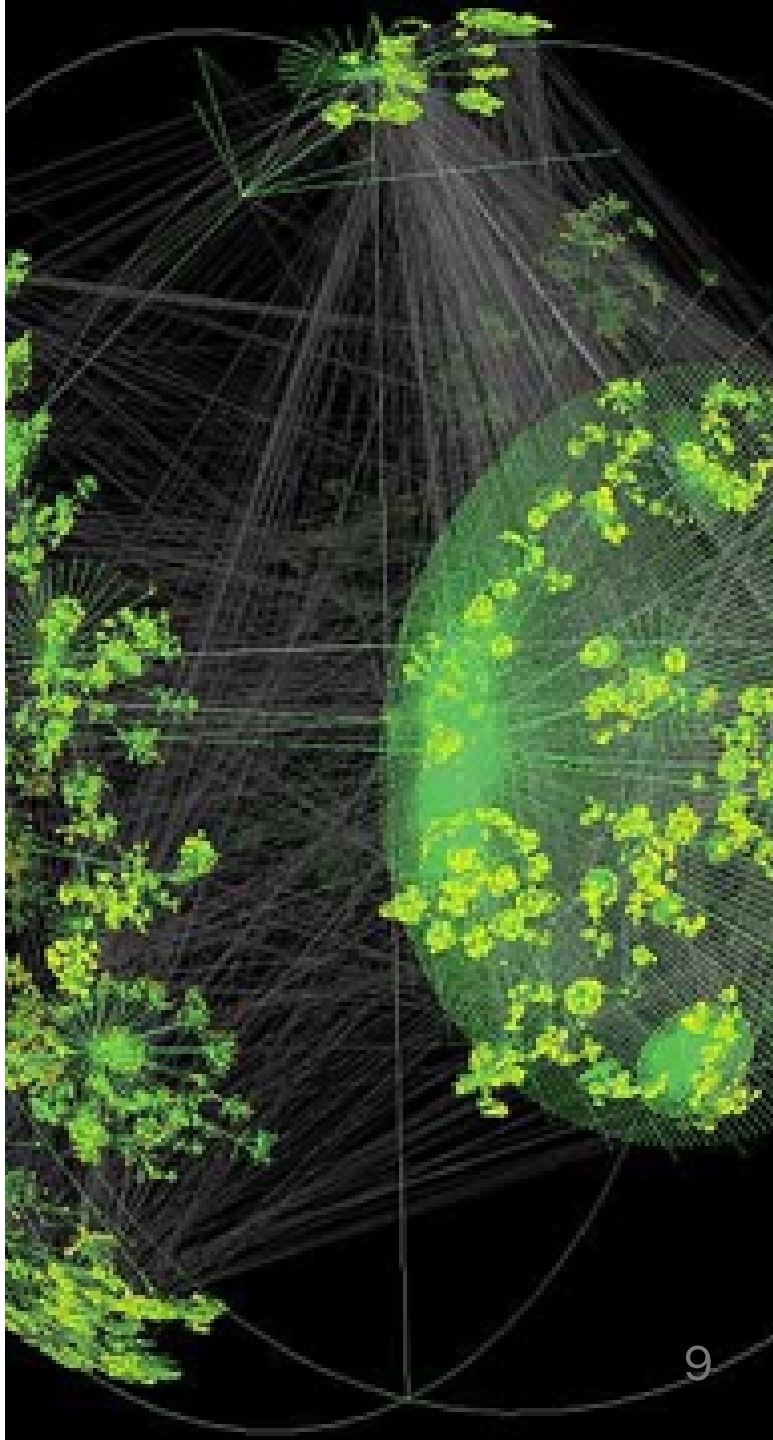
Lélia Blin

Internet



Un réseaux c'est quoi de nos jours ?

- **Très grands réseaux**:
  - Explosion combinatoire du nombre de machines connecté
- **Non homogène**
  - Explosion du type de machines connectées
  - Filaire ou non
  - Statique ou non



Lélia Blin

Non homogène

Des réseaux pourquoi faire?

Echanger de l'information

- Emails, textes, son, images, vidéo, ...

Partager des ressources

- imprimante, mémoire, base de données, services

Tout cela s'organise...

Qui c'est le big boss ?

Pas de big boss

Comment réaliser une tâche commune sur un réseaux à grande échelle?

Question :

Qui est n'est en mars?

Approche centralisé

Qui est n'est en mars?

L’enseignant.e demande à l'ensemble de la salle "Levez la main si vous êtes né en mars"

Approche répartie (distribué)?

Approche répartie

Qui est n'est en mars?

Chaque élève demande à ses voisins qui demandent à leurs voisins

Au delà des réseaux de communications

Algorithmique distribué de nos jours

Système distribué

- L'internet,
- wifi
- cloud
- parallèle,
- les systèmes multicœurs,
- les réseaux mobiles,
- ....

Mais aussi

- colonie de fourmis,
- un cerveau
- la société humaine

> Tous ces systèmes répartis ne se modélisent pas de la manière

Système réparti

C'est un réseau qui est constitué d'entités de calcul (homogène ou non)

- puissance de calcul
- mémoire
- ...

et de moyen de communication

Communications

- Liens de communications
  - filaire
  - wifi
  - radio
  - ....

- Il sont utilisés pour échanger de l'information
- l'information est transporté par des **messages**

Algorithmique répartie

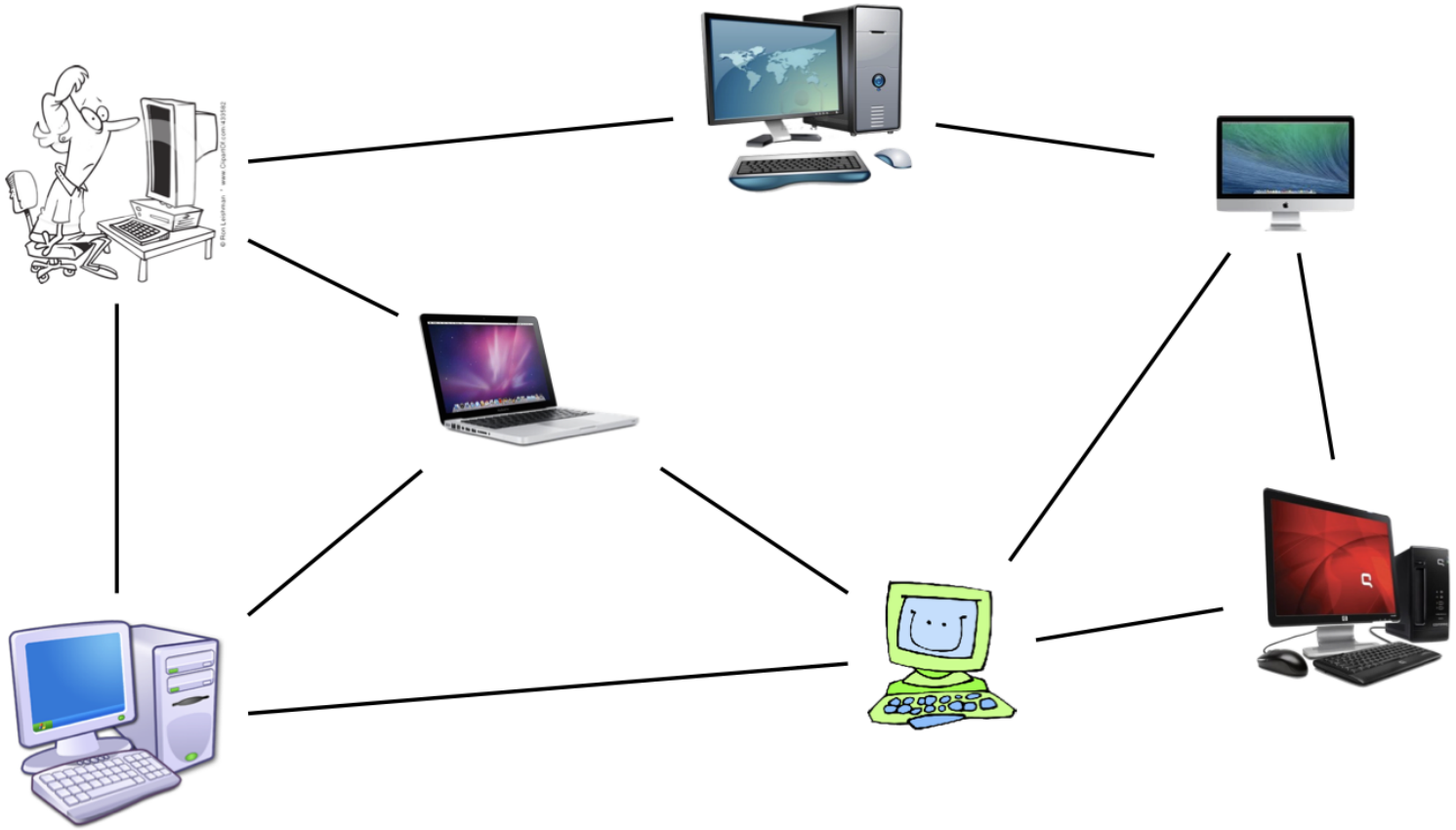
Définition

- Calcul effectué par des entités de calcul autonomes ayant pour but de solutionné un même problème.
- Chaque entité possède une partie de l'information et peut communiquer avec son voisinage.

Les modèles principaux

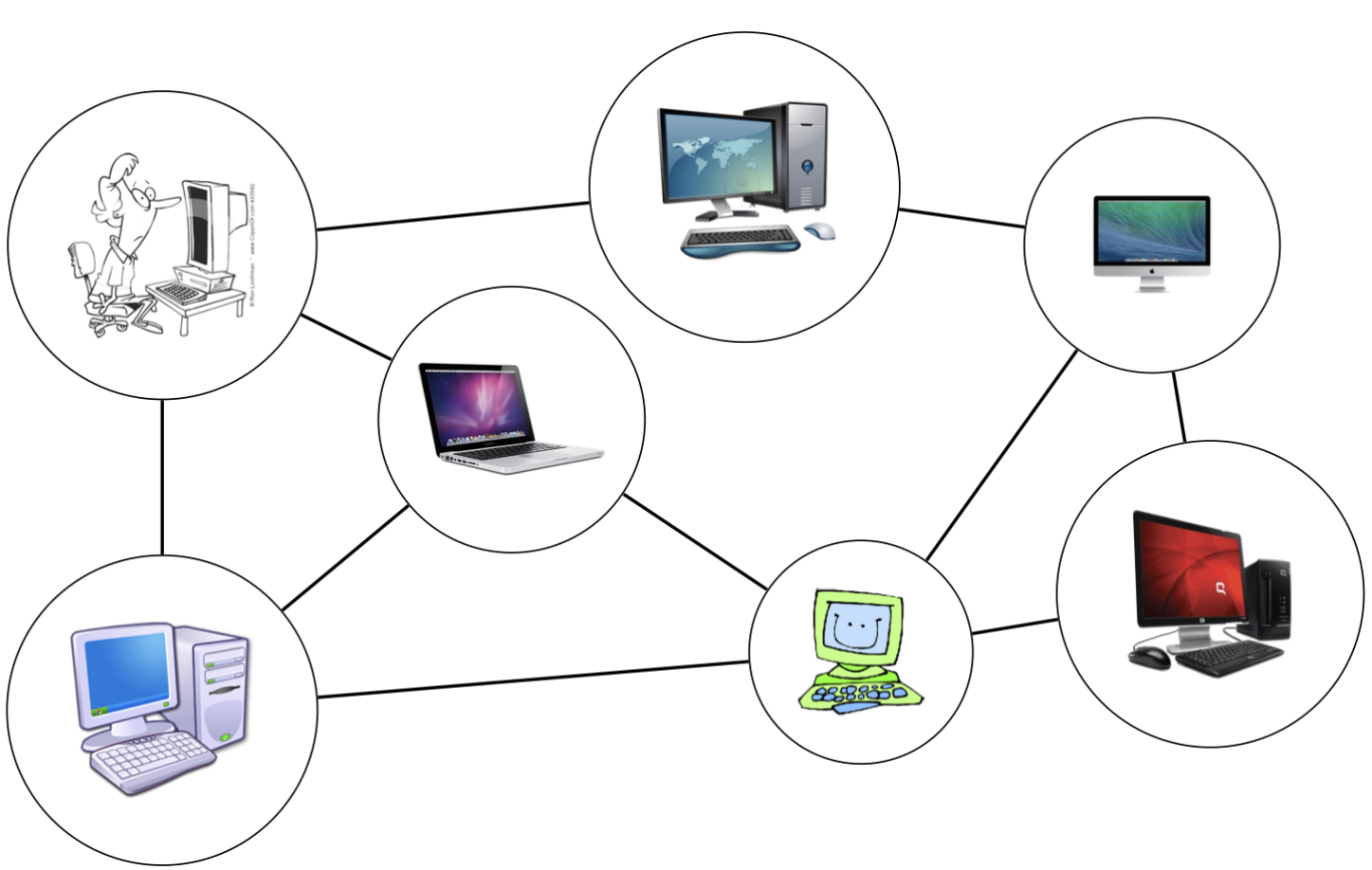
Lélia Blin

Modélisation des réseaux



Lélia Blin

Modélisation des réseaux

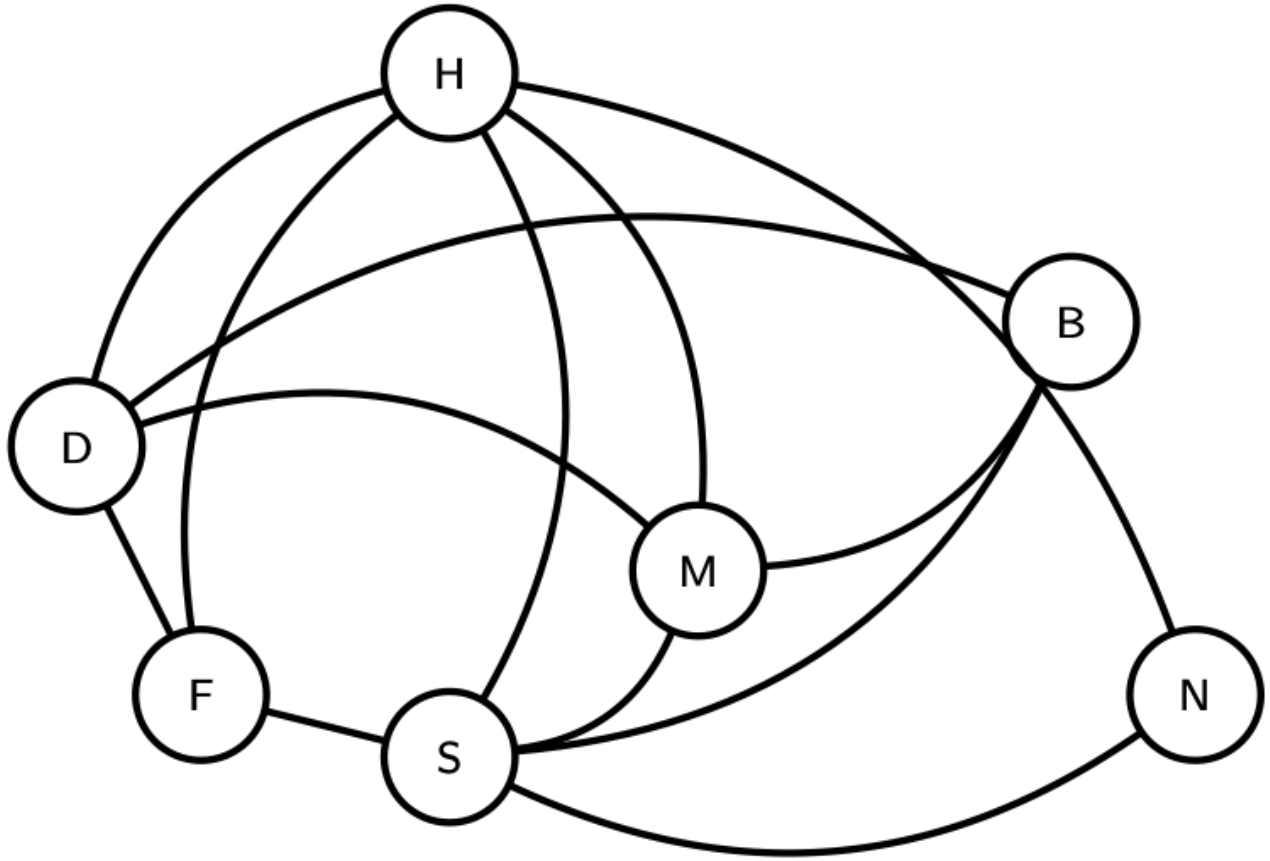


Vocabulaire et définitions

Entité de calcul

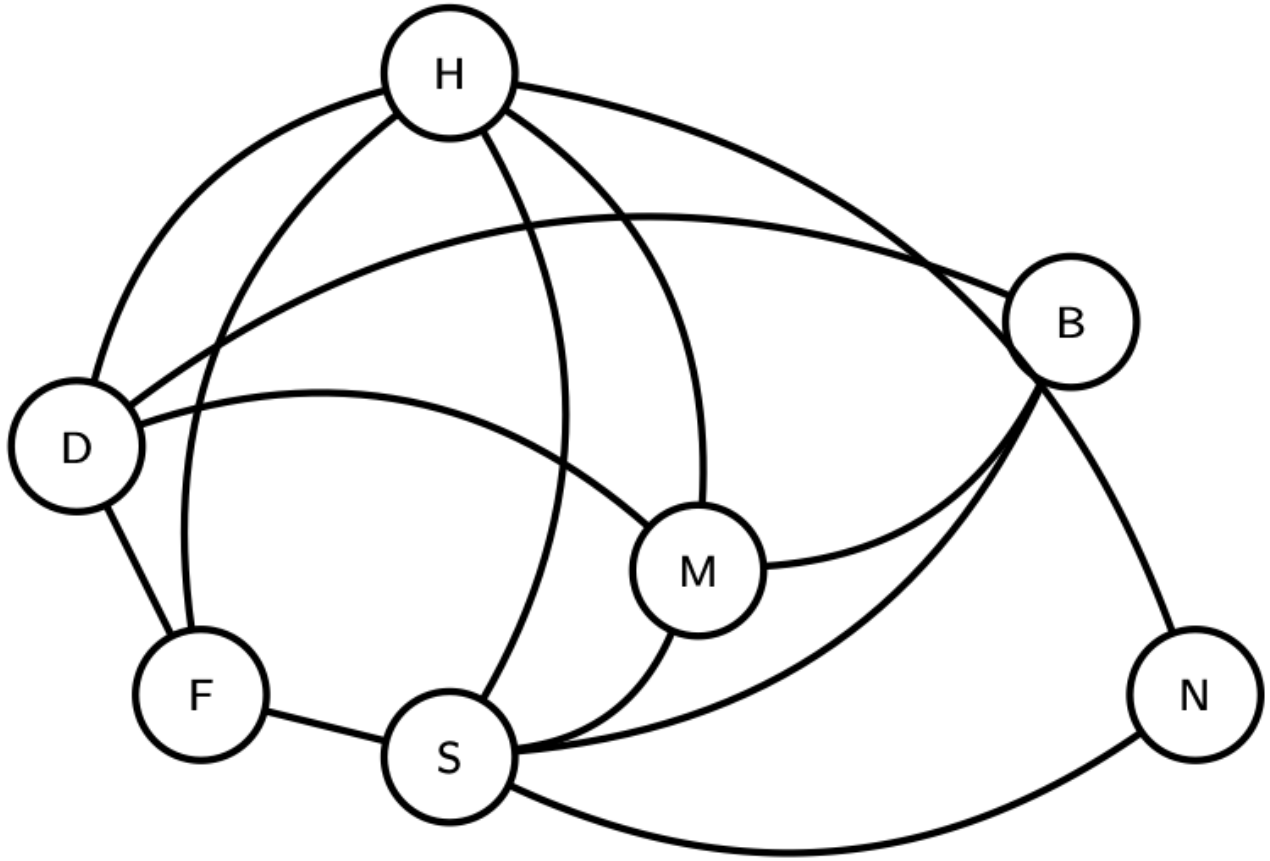
Les entités de calculs peuvent être appelés de différentes façon:

- **Noeuds**
- Processeurs
- Processus



Communications

Les communications se font à l'aide canaux de communications aussi appelé liens de communication, arêtes.



Liens de communications

- Les liens peuvent être unidirectionnel
- Les liens peuvent être bidirectionnel

Ordre des messages

- Le transit des messages à l'intérieur des liens peuvent être FIFO ou pas

Communications FIFO

- Soit deux message m_1 et m_2 envoyer par le noeud A reçu par le noeud B
- Avec m_1 envoyer avant le message m_2
  - m_1 arrivera en B avant m_2

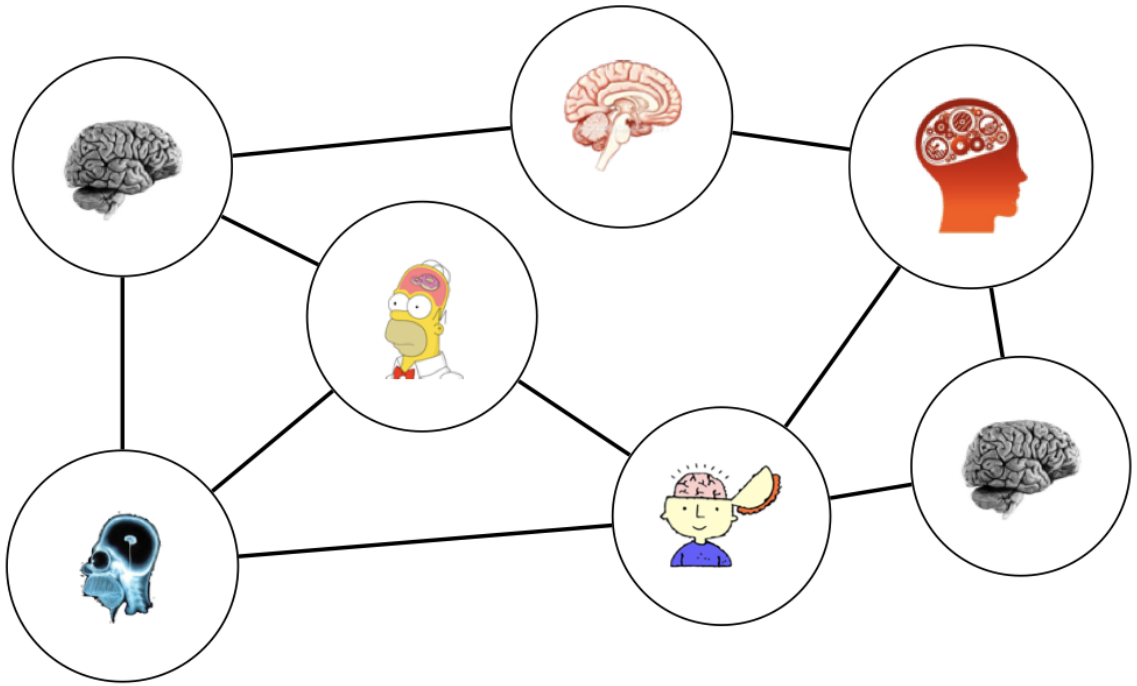
Communications non FIFO

- Soit deux message m_1 et m_2 envoyer par le noeud A reçu par le noeud B
- Avec m_1 envoyer avant le message m_2
  - m_1 arrivera en B avant m_2 ou après m_2

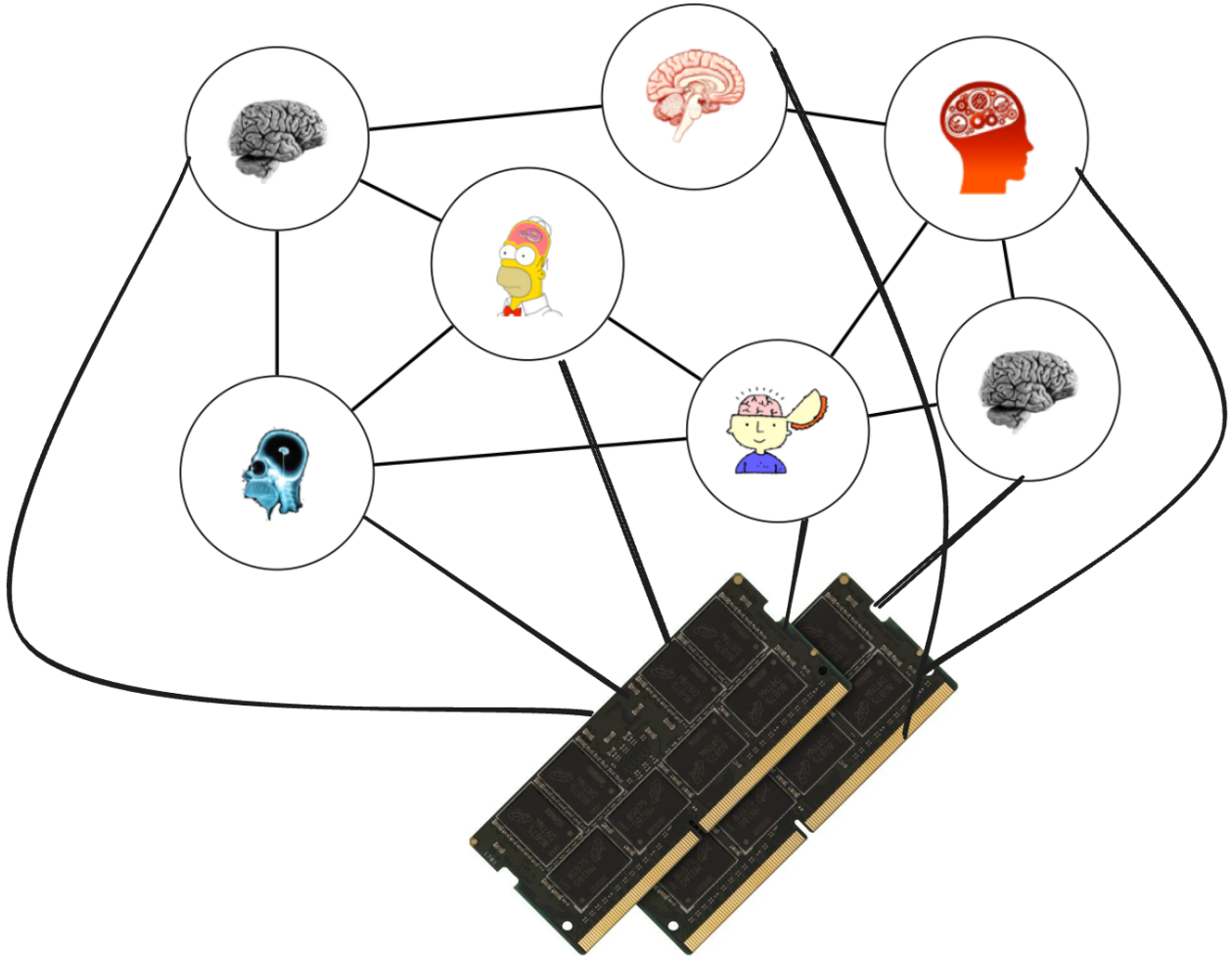
Mémoire des noeuds

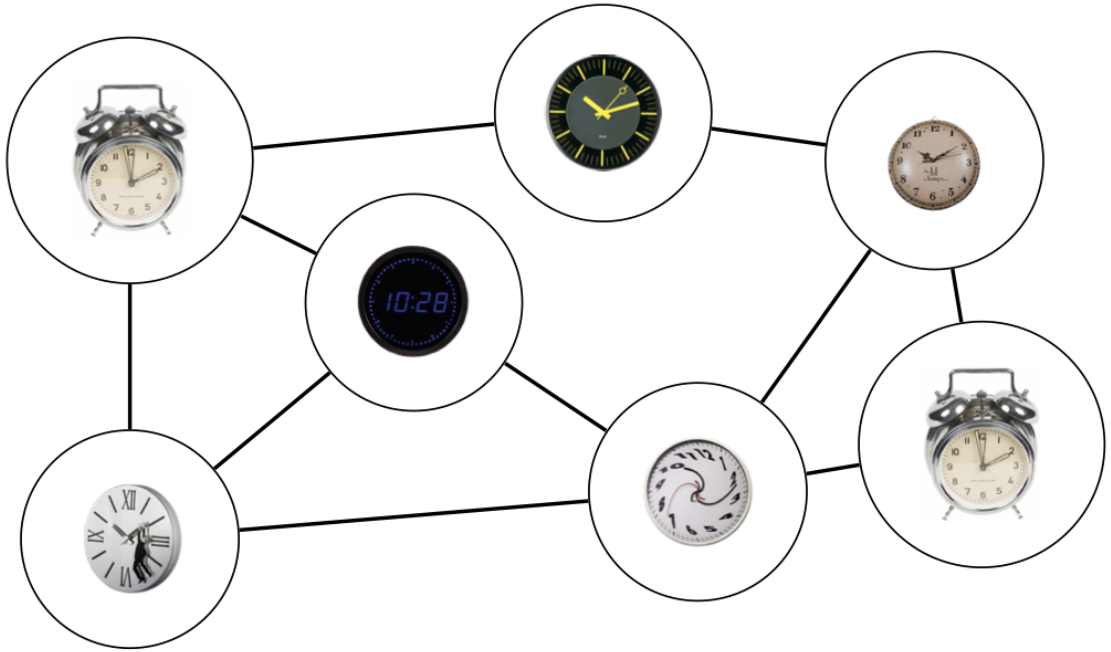
Chaque noeuds a sa propre mémoire:

- chaque mémoire peut-être de taille différente



Mémoire partagée





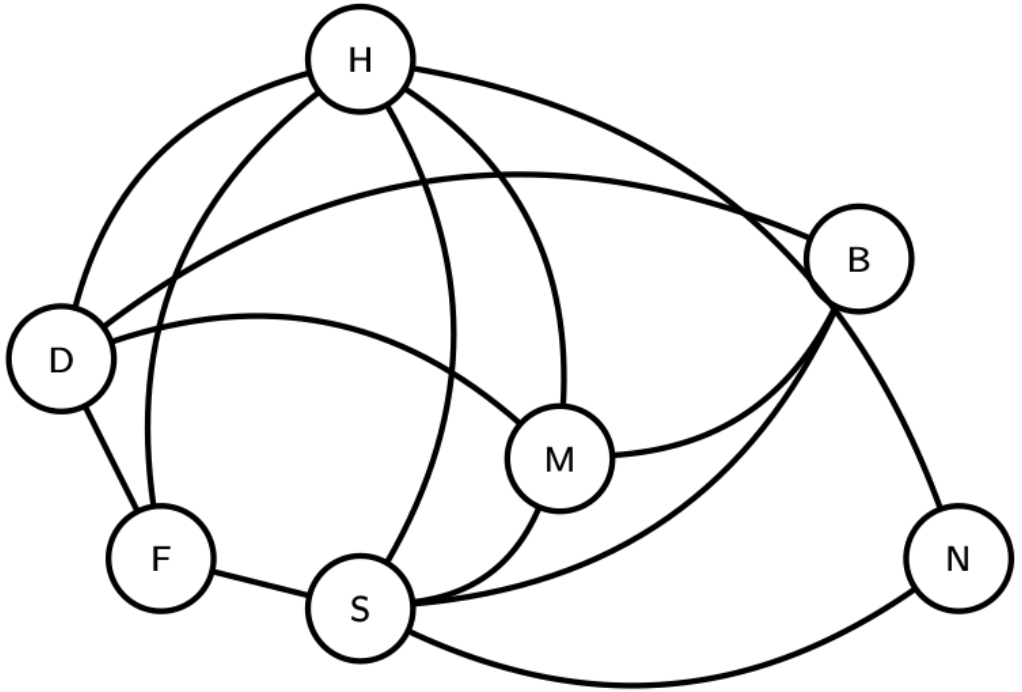
Horloge

Chaque noeud a sa propre horloge

- les temps des horloges peuvent être différents.
- Il n'y a pas forcément d'horloge commune partagée

Lélia Blin

Réseaux synchrones



Réseaux synchrone

Définition informelle

- Les noeuds ont des temps de calculs identiques
- Les temps de circulation des messages sont identiques

Réseaux synchrone

Définition formelle

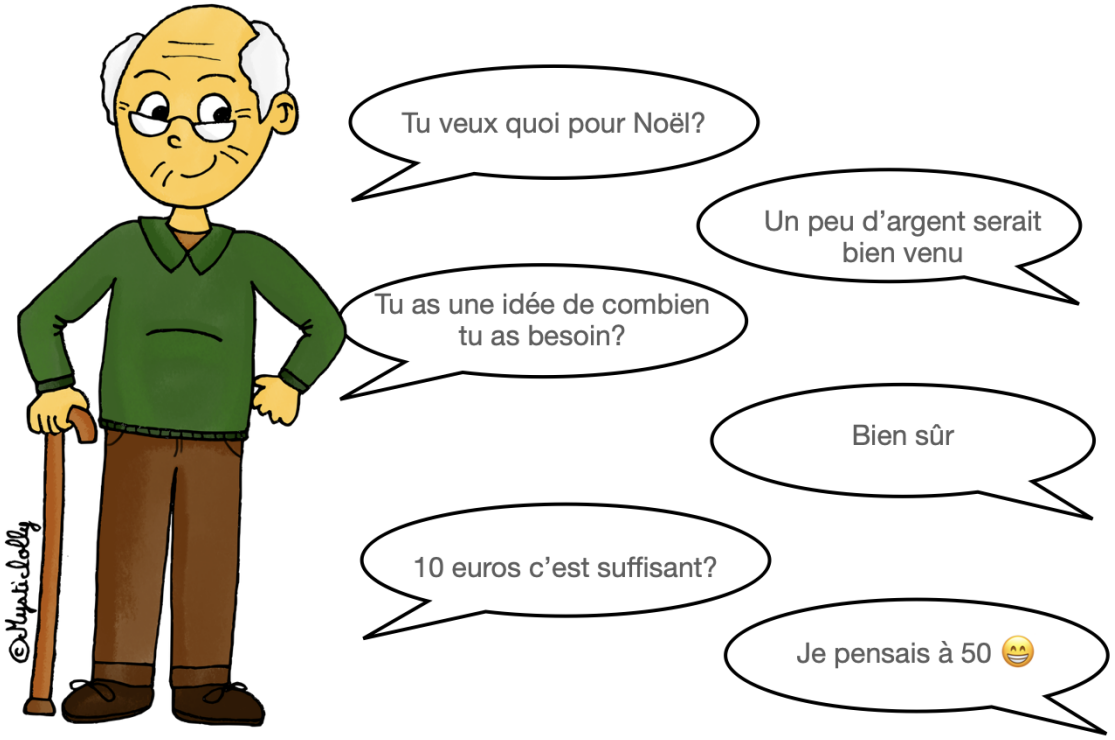
- Les noeuds calculent par rondes synchrones
- Dans une ronde, chaque noeud exécutent les étapes suivantes
  - Effectuer des calculs locals
  - Envoyer des messages à ses voisins
  - Recevoir des messages de ses voisins

Réseaux asynchrones

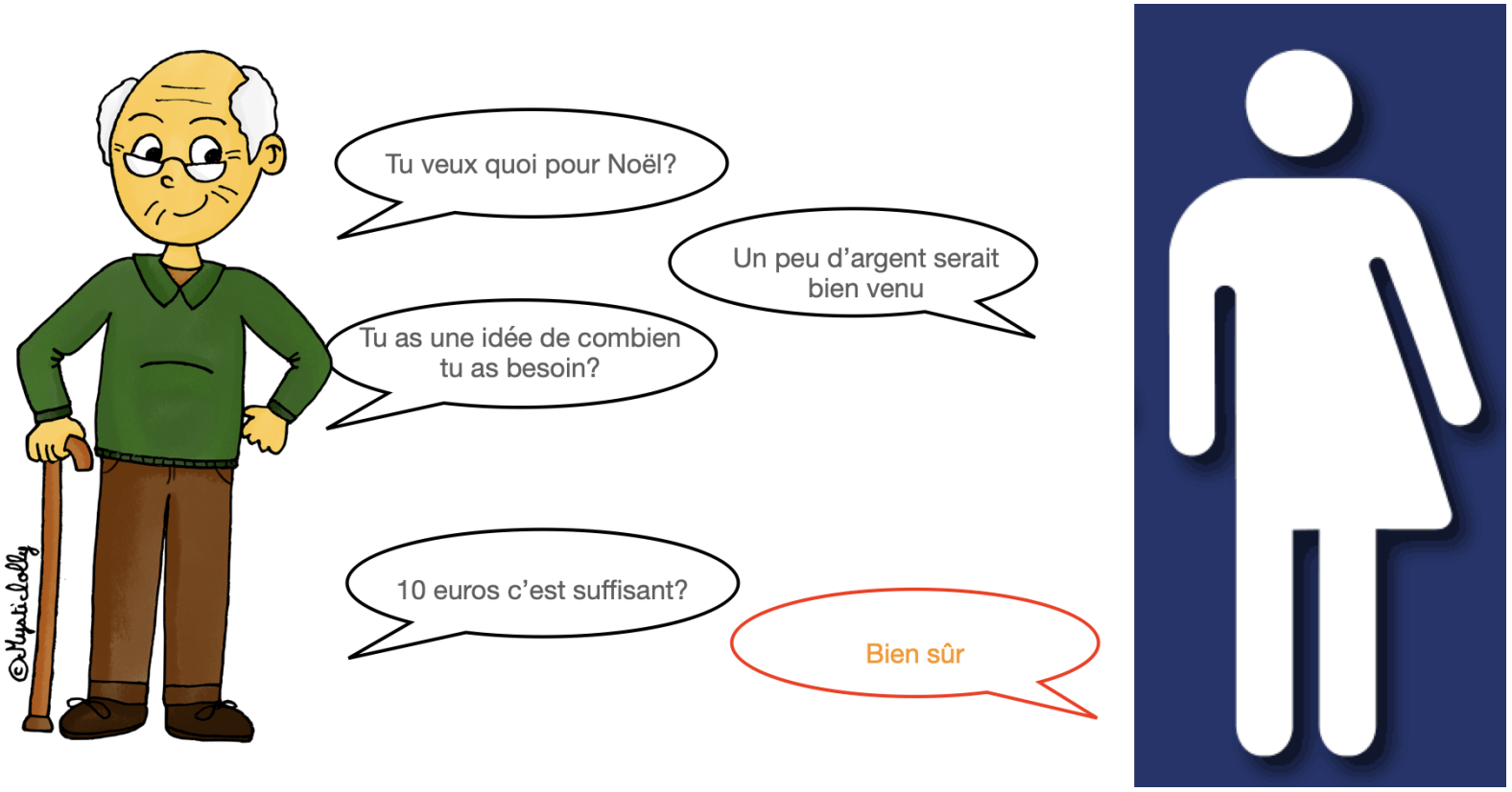
Définition

- Les noeuds ont des temps de calculs différents.
- Le temps de circulation des messages est non borné mais fini.

Asynchrones vs synchrone -- Conversation synchrone

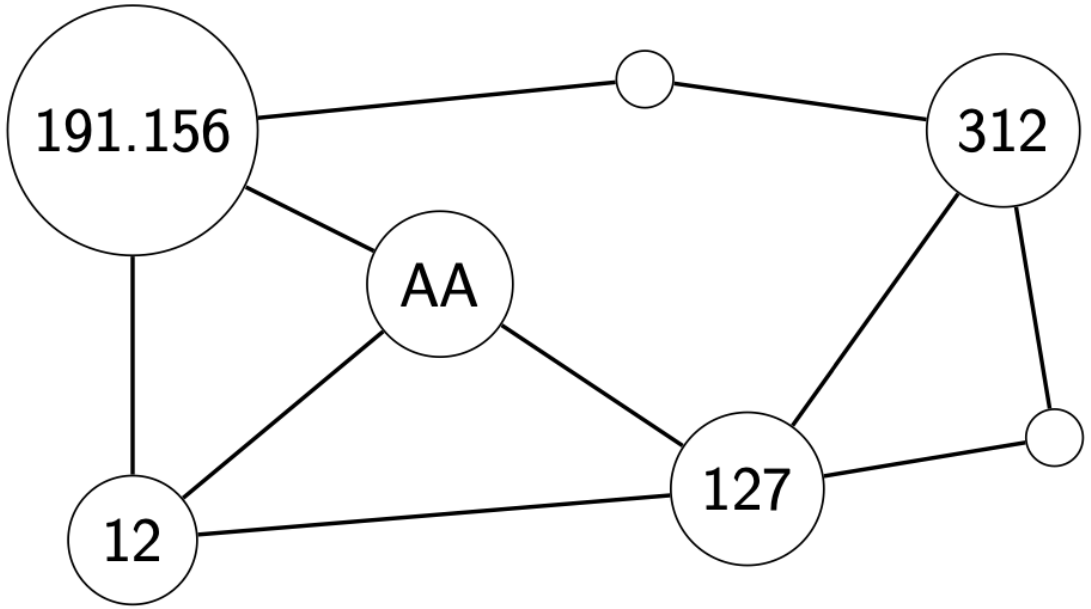


Asynchrones vs synchrone -- Conversation asynchrone



Identifiant

> Chaque noeud possède ou pas un identifiant (unique).



Noeud

Un noeud est une puissance de calcul qui a

- Une mémoire locale
- Un programme local
  - Un ensemble de données et de variables locales
  - Un état local
- Possède ou pas d'identifiant
- Possède peu ou pas de connaissance

Connaissance locale sur les liens de communications

Le noeud v :

1. connait qu'il a d liens de communications
2. peu numéroter ses liens (numéro de port)
3. connait les identifiants de ses voisins.

Connaissance globale

- Le noeud a **aucune connaissance ** du réseau: le plus réaliste.
- Le noeud a la connaissance de la taille du réseau (nombre de noeuds): peu réaliste (grande taille, dynamique)
- Le noeud a la connaissance du diamètre du réseaux
- Le noeud a la connaissance de la topologie du réseaux (grille, anneaux,...)

Algorithmes répartis

- Tous les noeuds ont le même algorithme séquentiel

- Cet algorithme réagit à la réception d'information (messages)
- Cet algorithme envoie de l'information (message)

Variables locales

- Tous les noeuds ont le même algorithme séquentiel
  - Donc tous les noeuds ont des variables qui portent le même nom.

Vision extérieure globale :

- Pour reconnaitre de quelle variable ont parle quand on étudiera les algorithmes on ajoutera l'identifiant du noeud de la variable.

Variables locales vue globale : Exemple

- Algorithme de coloriage des noeuds:
  - chaque noeud a une variable couleur noté c
  - c_v sera la variable couleur du noeud qui a l'identifiant v
  - c_u sera la variable couleur du noeud qui a l'identifiant v

Pseudo-code

- C'est du pseudo code classique (séquentiel) avec
  - des variables
  - des tests
  - des boucles
  - ...

Suivant le modèle le pseudo code peut changer de présentation

Pseudo code pour le passage de messages

Le pseudo code se présente par **block**

Un bloc d'**initialisation**

Un bloc par type de **message** reçu

- Lors de la réception du message $< toto >$ envoyer par le noeud u
- instructions...

Attention: il y aura autant de blocs que de types de messages.

La vraie vie vs la construction des algorithmes

- Quel est le noeud qui commence ?

Qui commence?

Les processus qui veulent faire une tâche effectueront un **réveil spontané**.

Attention :

- On ne peut pas s'aider des noeuds qui commence en les choisissants.
- Il faut qu'il y est au moins un noeuds qui commence.
- Mais tous les noeuds peuvent commencer.

Que font les noeuds qui ne se réveillent pas spontanément?

- Les noeuds qui ne se réveillent pas spontanément seront réveillé à la réception d'un premier message.

Un premier Algorithme réparti

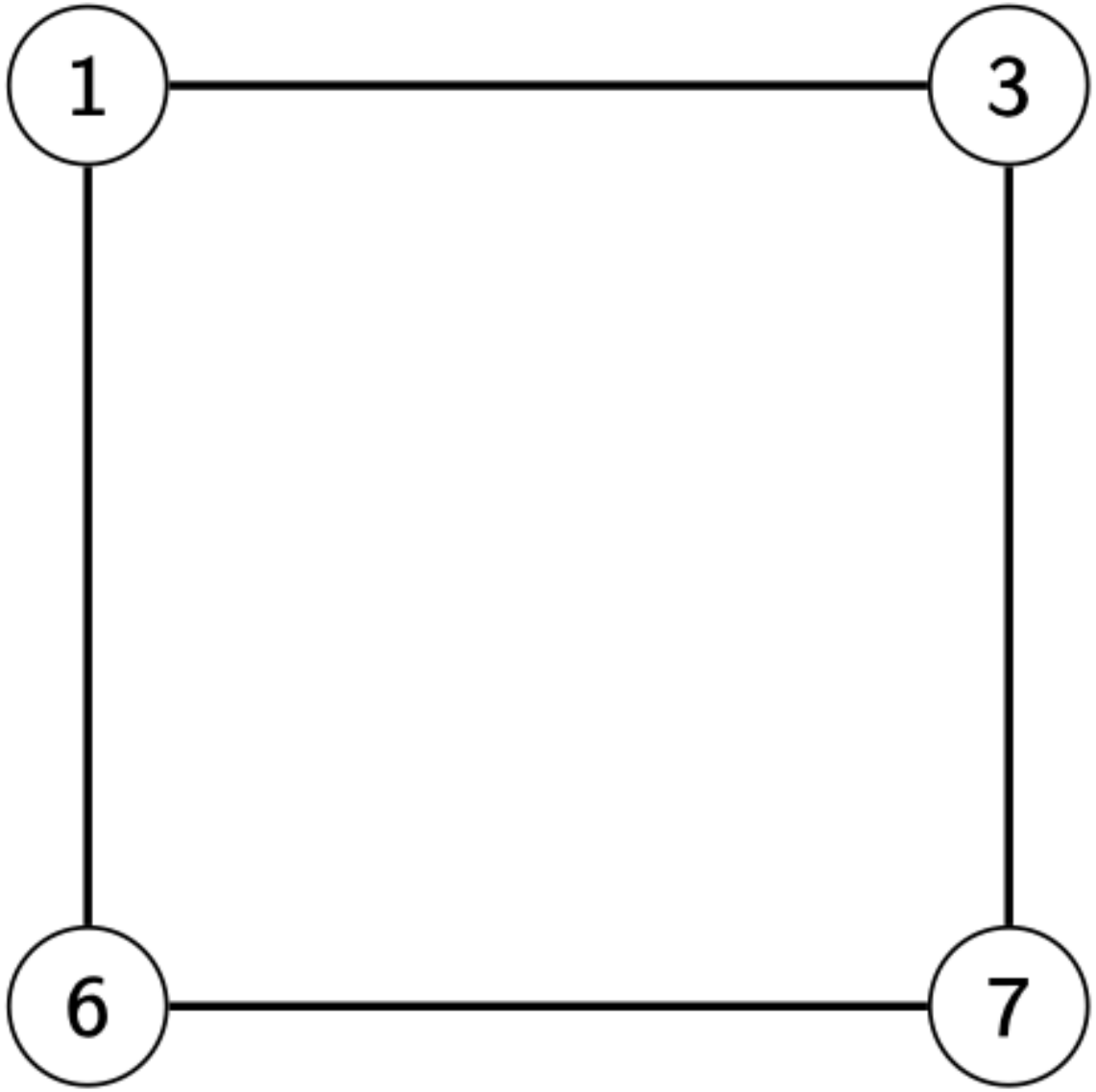
La diffusion

```
Reveil spontané du noeud i:
        val_i=identifiant_i
        Pour tout j∈Voisins_i Envoyer <Valeur,val_i> à j.
```

```
Lors de la réception de <Valeur,val_j> envoyer par le noeud j
         tmp=val_i
         if val_i=null :
                val_i:=idententifiant_i
         if val_i<val_j
                val_i=val_j
         if tmp!= val_i: Pour tout j∈Voisins_i Envoyer <Valeur,val_i> à j.
```

Lélia Blin

Exemple



Lélia Blin

Exemple synchrone

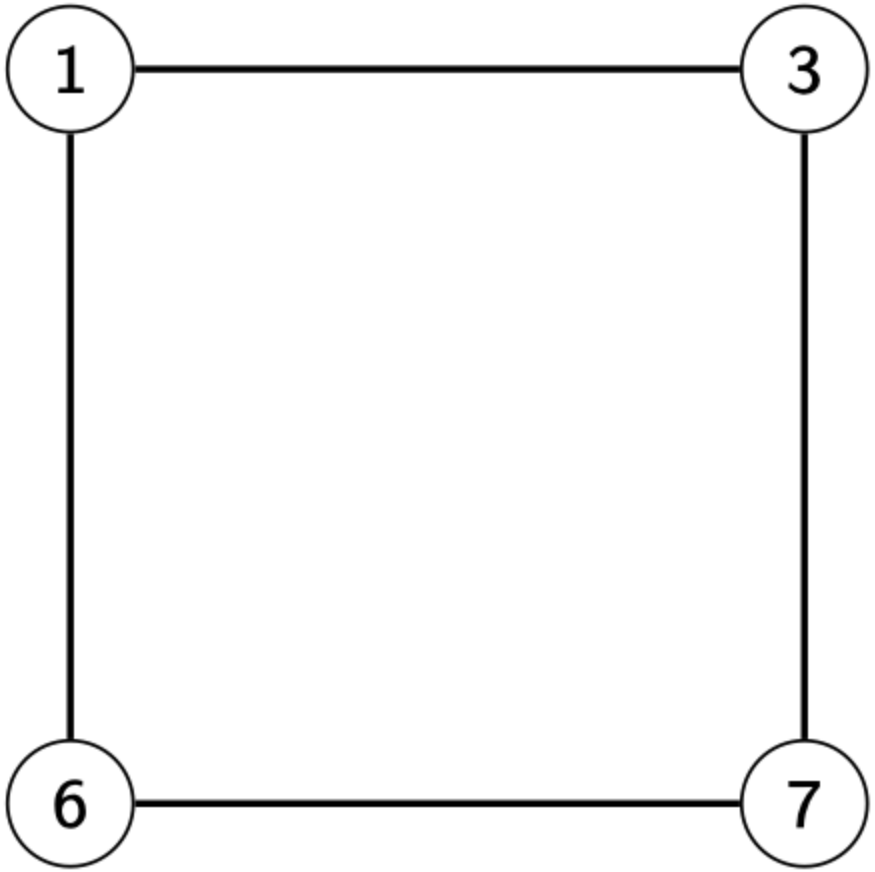
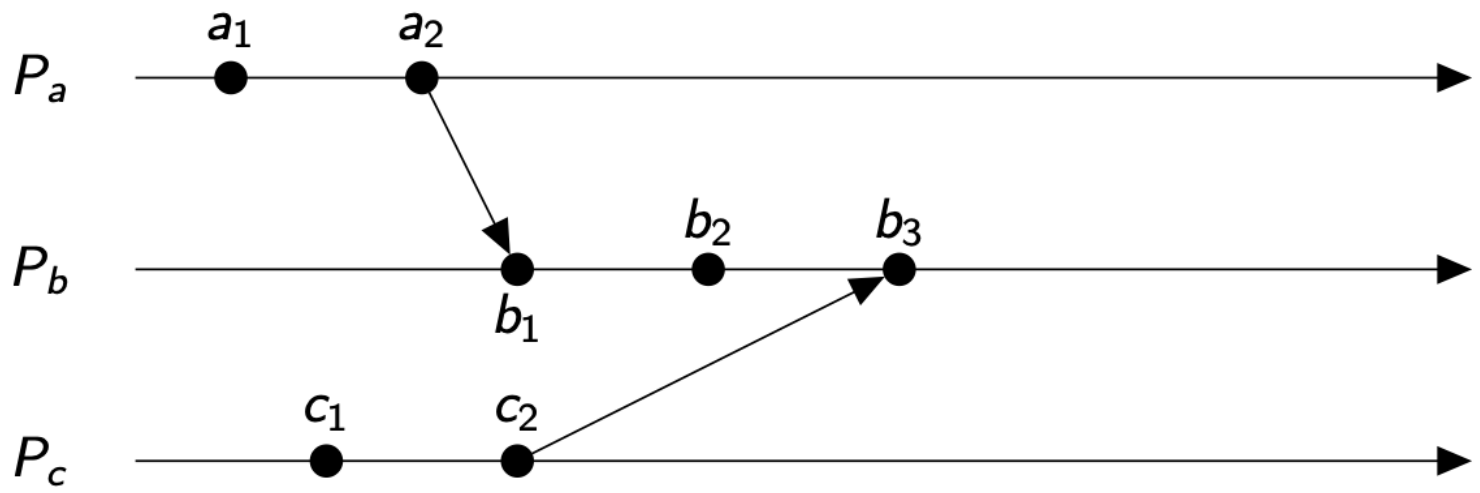


Diagramme de trace

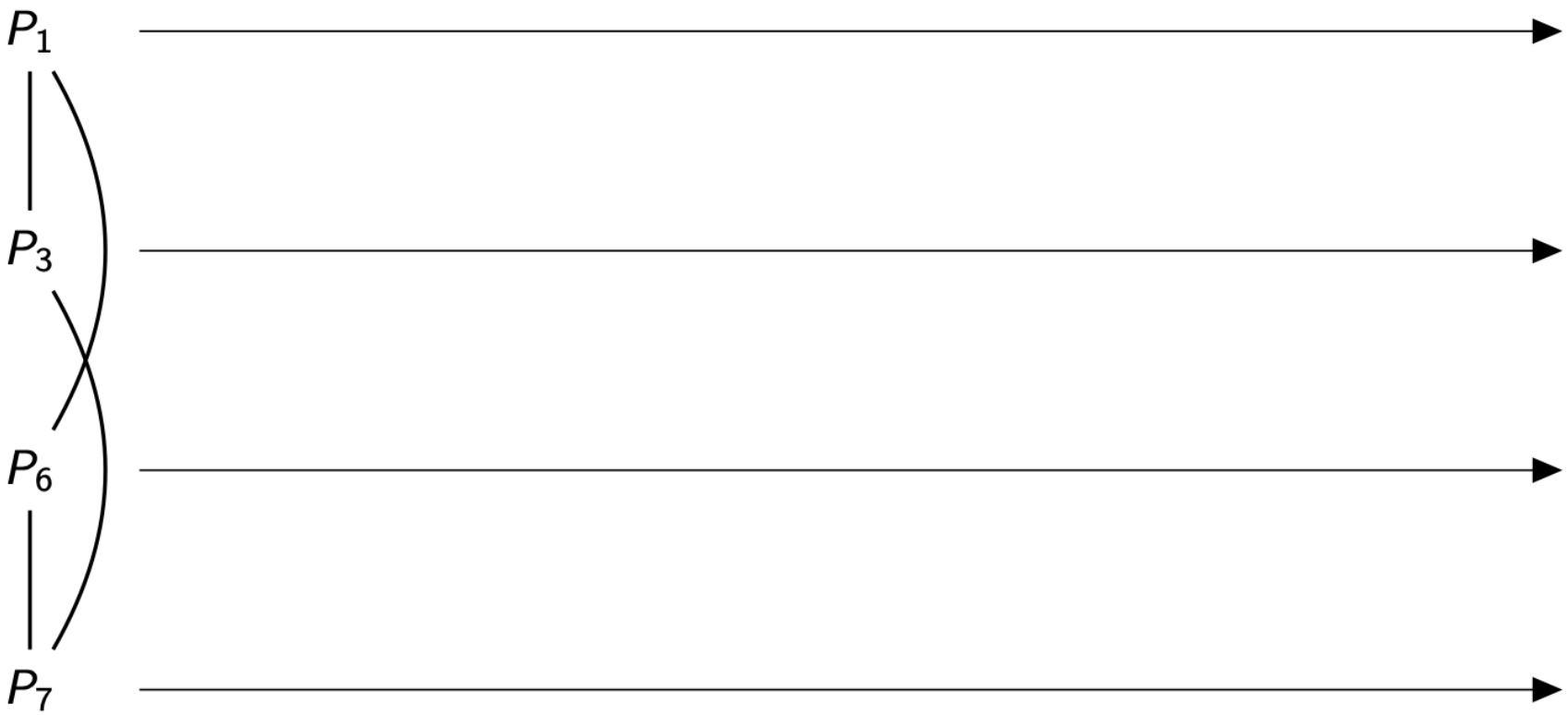


Le temps s'écoule ici vers la droite.

Les points correspondent à des événements internes.

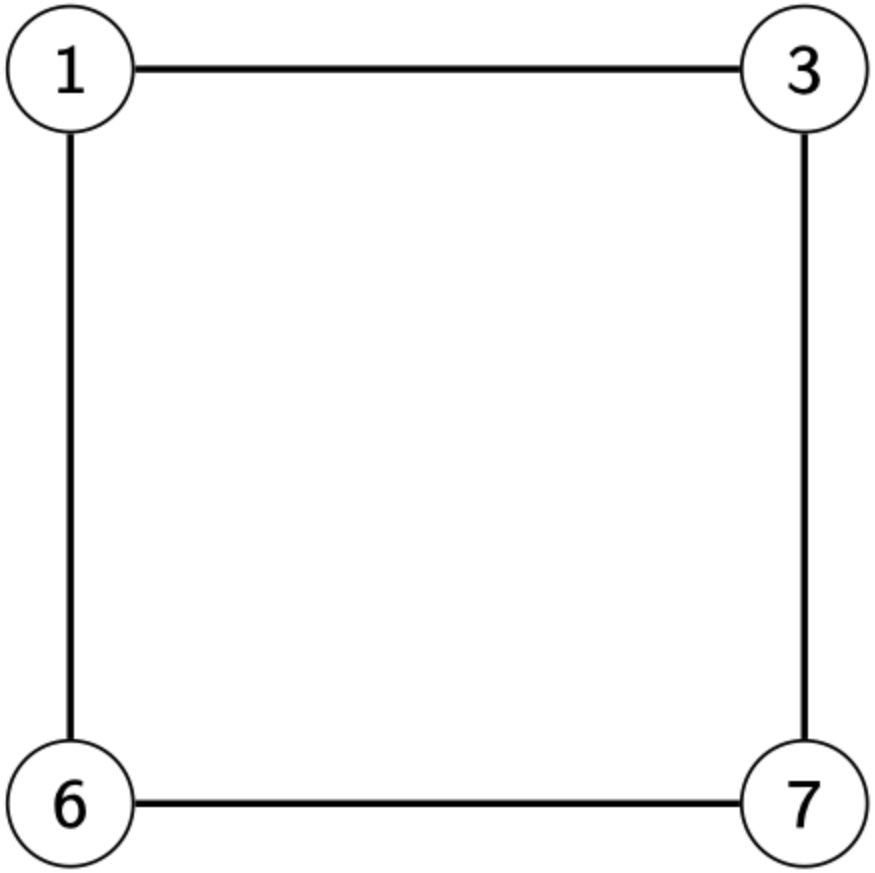
Lélia Blin

Exemple synchrone



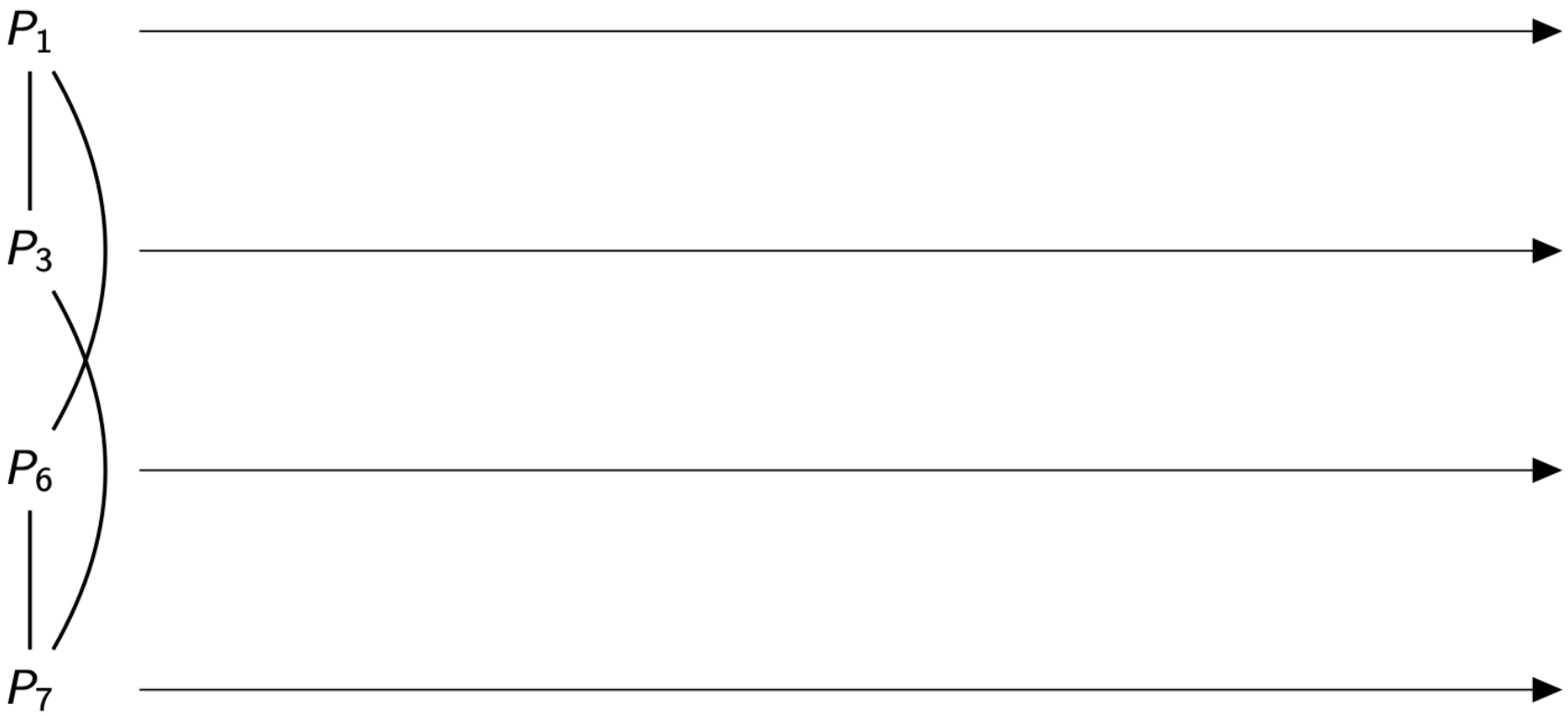
Lélia Blin

Exemple asynchrone



Lélia Blin

Exemple asynchrone



Qualités de cet algorithme?

Qualités de cet algorithme

Avantage(s):

- Simplicité

Inconvenient(s) :

- ...

Qualités de cet algorithme

Avantage(s):

- Simplicité

Inconvenient(s) :

- Les noeuds reçoivent plusieurs fois une même information
- Grand nombre de messages échangés

Comment mesurer la qualité d'un algorithme réparti?

Complexité des algorithmes séquentiels

Notions

- Nombre d'opérations élémentaires
- Espace mémoire occupé

Vocabulaire

- Complexité temporelle
- Complexité spatiale

Comment mesurer la qualité d'un algorithme réparti?

Complexité des algorithmes répartis

> Le nombre de messages échangés.

Complexité des algorithmes répartis

- Le nombre de messages échangés.
- Est-ce suffisant?

Taille des messages

Taille d'un messsage

Nombre d'information que contient le message (en binaire)

Exemples :

- Un mot clé: taille constante $O(1)$bits
- Un identifiant:
  - Si il y a n noeuds dans le réseaux, et que l'on note de 1 à n les identifiants des noeuds
  - Il faut $\log_2 n$ bits pour coder l'identifiant n
  - Taille $O(\log_2 n)$ bits

Comparaison d'algorithmes répartis

Soit A un algorithme échangeant:

- $O(n)$ messages
- de taille $O(n \log_2 n)$ bits

Soit B un algorithme échangeant:

- $O(n^2)$ messages
- de taille $O(\log_2 n)$ bits

Quel est le meilleur algorithme?

Comparaison d'algorithmes répartis

Quel est le meilleur algorithme?

A échange au total:

- $O(n)$ messages $\times$ taille $O(n \log_2 n)$ bits
- $\rightarrow O(n^2 \log_2 n)$ bits

B échange au total:

- $O(n^2)$ messages $\times$ de taille $O(\log_2 n)$ bits
- $\rightarrow O(n^2 \log_2 n)$ bits

A et B sont équivalent en terme d'information échangées.

Taille des messages

IPV6 (Wikipédia)

- Sur une interface déterminée, une trame a une taille maximale appelée Maximum Transmission Unit (MTU)
- Lorsque la longueur du paquet (datagramme) est supérieure, l'information est fragmentée.
- La taille maximum supporté par IPV6 la taille maximale d'un paquet est porté à 4 Go

Taille des messages

Conclusion

Les messages de grosses tailles seront divisés en messages de plus petites tailles

Taille des messages

En algorithmique répartie

La taille "raisonnable" communément admise est $O(\log_2 n)$ bits, où n est le nombre de noeud du réseau.

Questions fondamentales liés aux systèmes distribués

Coordination

- Comment coordonner un système distribué de manière à ce qu'il exécute une tâche de manière efficace ?
- Quelle est le coût d'une telle démarche ?

Localité

- Les réseaux ne cessent de croître.
- Heureusement, il n'est pas toujours nécessaire de disposer d'informations globales pour résoudre une tâche, il suffit souvent que les nœuds communiquent avec leurs voisins.
- Dans ce cours, nous verrons si une solution locale est possible.

Parallélisme

- A quelle vitesse pouvez-vous résoudre une tâche si vous augmentez votre puissance de calcul, par exemple en augmentant le nombre de nœuds qui peuvent partager la charge de travail ?
- Quel est le degré de parallélisme possible pour un problème donné ?

Rupture de symétrie

- Il est parfois nécessaire de sélectionner certains nœuds pour organiser le calcul ou la communication.
- Pour ce faire, on utilise une technique appelée rupture de symétrie.

Incertitude

Si nous devons nous mettre d'accord sur un seul terme pour décrire ce cours, c'est probablement celui d'"incertitude". Comme l'ensemble du système est distribué, les nœuds ne peuvent pas savoir ce que font les autres nœuds à ce moment précis, et les nœuds doivent résoudre les tâches à accomplir malgré l'absence de connaissances globales.

Tolérance aux pannes

- L'un des principaux avantages d'un système distribué est que, même en cas de défaillance, le système dans son ensemble peut survivre.

Bibliographie

- **Gérard Tel**, *Introduction to distributed algorithm*, Cambridge University Press
- **Nancy Lynch**, *Distributed Algorithms* Morgan Kaufman Publishers
- **Michel Raynal**, *Distributed Algorithms for Message-Passing Systems* Springer
- **Hagit Attila, Jennifer Welch**, *Distributed Computing: Fundamentals, Simulations, and Advanced Topics* Wiley-Blackweel
- **Shlomi Dolev**, *Self-Stabilization*, MIT Press
- **Karine Altisen, Stephane Devisme, Franck Petit, Swan Dubois**, Introduction to Distributed Self-Stabilizing Algorithms**, Morgan & Claypool Publisher