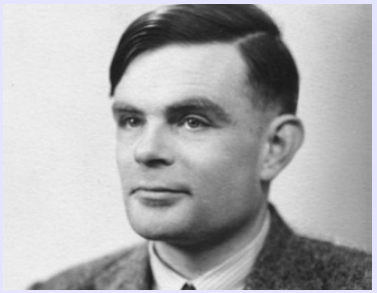
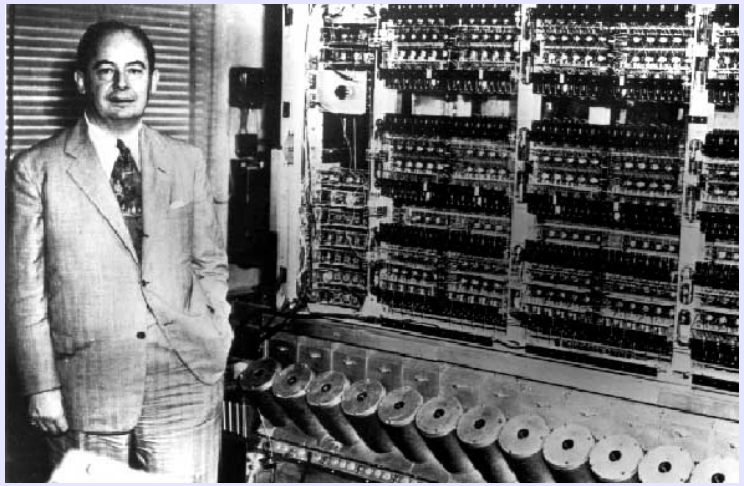




Information, Calcul et Communication

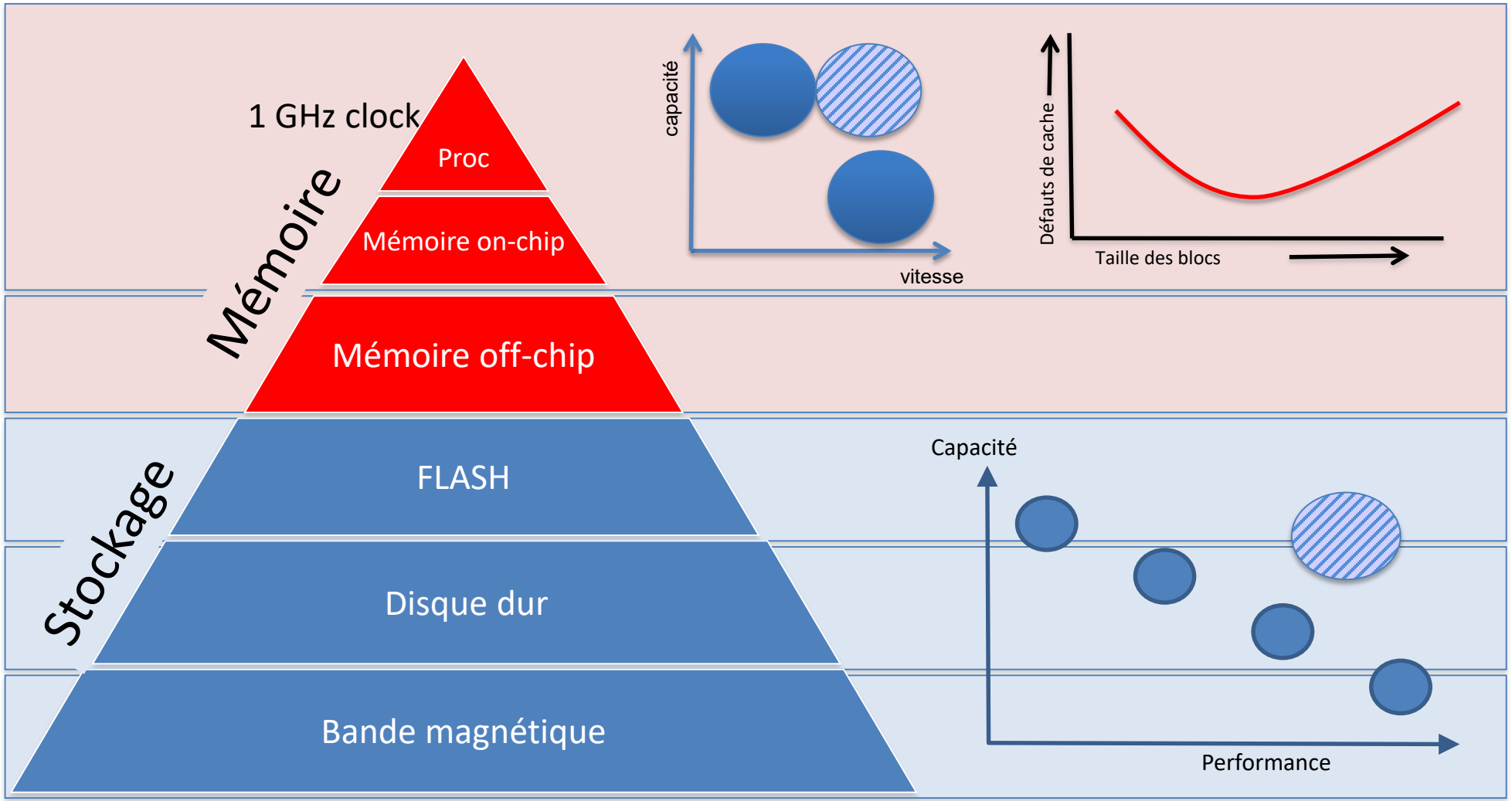
Leçon III.3.1: Stockage et organisation des données

Faculté Informatique et Communications

A. Ailamaki, P. Janson, W. Zwaenepoel

- ▶1. Technologies principales de stockage
- ▶*2.* Organisation des données

1. Technologies de stockage & *Memory Wall*



La Hiérarchie de Stockage et le *Memory Wall*

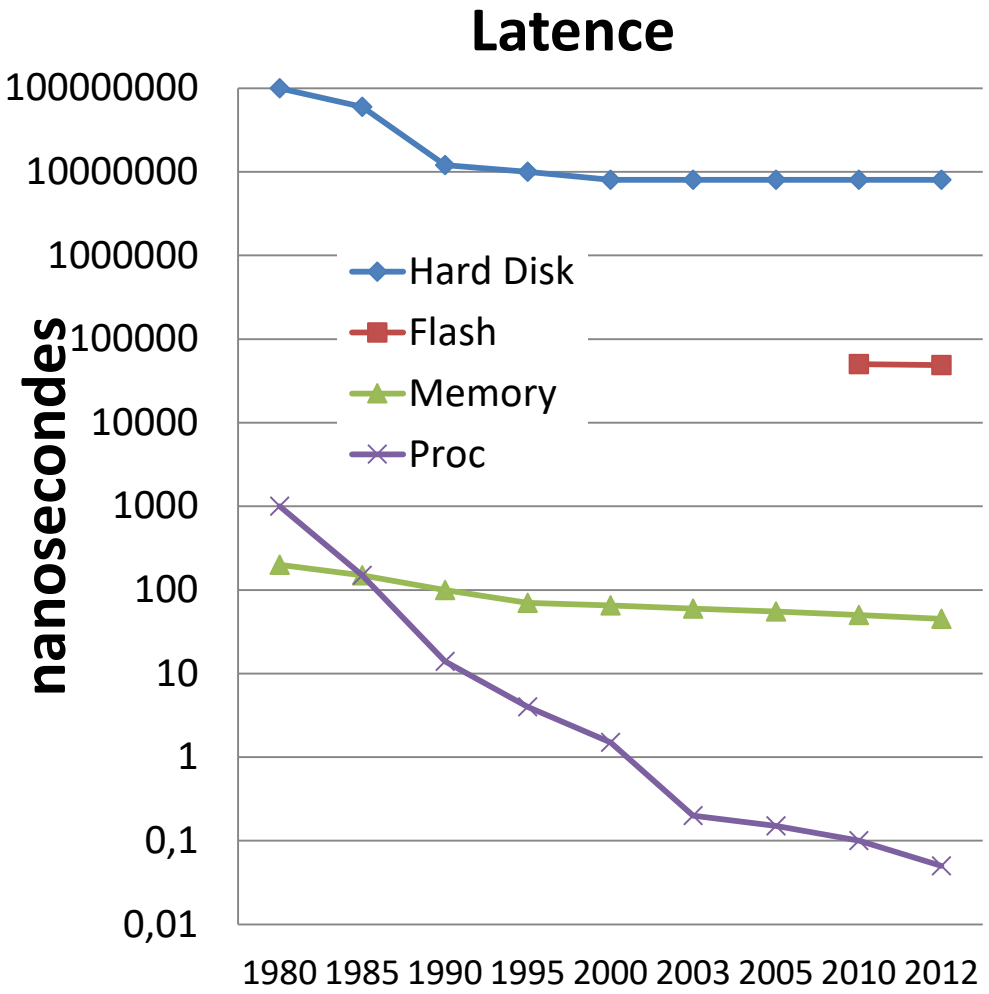
► Les performances des ***mémoires*** s'améliorent surtout en terme de ***capacité*** alors que celles des ***processeurs*** s'améliorent en terme de ***vitesse***.

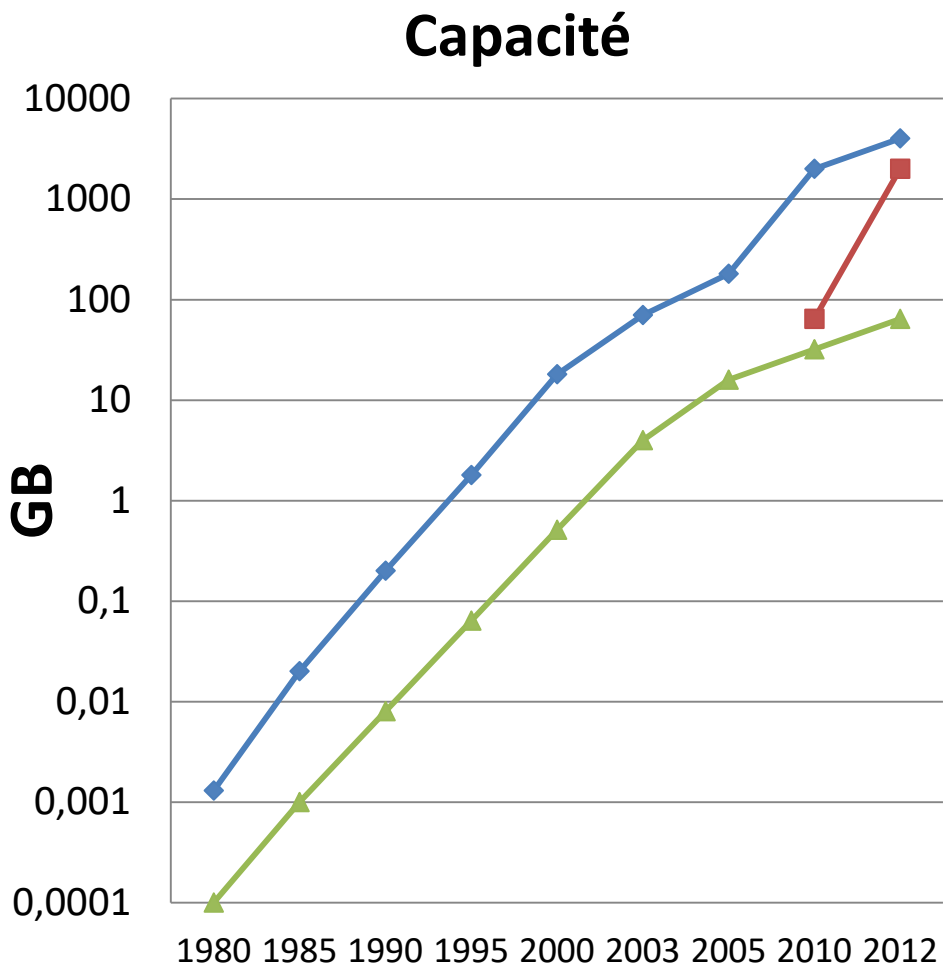
- Or, la véritable vitesse d'un processeur est celle du traitement des données qu'il accède en mémoire. L'accès mémoire devient un goulot d'étranglement qui a conduit à l'expression ...

- **Memory Wall** : la latence de la mémoire détermine de plus en plus la vitesse des calculs à cause du temps d'accès aux données.

- La technologie Flash est une réponse récente (10 ans) à ce défi

Tendances Capacité vs. Latence





Pourquoi nous avons besoin du *Stockage*?

► La mémoire principale ne suffit pas?

► NON! Parce que:

- Elle coûte beaucoup trop cher (coût dans tableau suivant)
- Pas assez de capacité!
- La mémoire principale est ***volatile***. Nous voulons la sauvegarde des données, même en cas de panne de courant.

► D'où: ~ 60% du coût d'un système de production est dans les disques.

Technologie: latence, debit, coût et accès

	Latence	Débit	Coût ($/Go)	Capacité	Rétention	Accès
RAM	1 - 100 ns	Go/s	10	Mo - Go	NON	Aléatoire
Flash	µs	Go/s	0.5	Go - To	Oui	Aléatoire
Disques HDD	ms	100s Mo/s	0.05	> To	Oui	Aléatoire *avec délai*
Bandes magnétiques	Encore plus lent !	100s Mo/s	Encore moins cher !	Encore plus grand !	Oui	Séquentiel

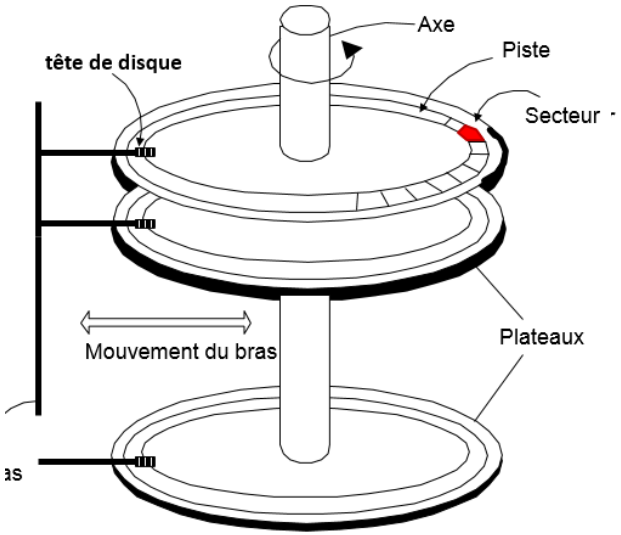
Flash

Accès aléatoire comme RAM
mais par pages comme HDD

électrons piégés dans semi-conducteur

HDD

Latence de rotation +
positionnement du bras



Bandes

Accès strictement séquentiel
=> latence de déroulement

← *charge magnétique*

Accès séquentiel et Accès aléatoire

► Accès séquentiel

(bande magnétique):

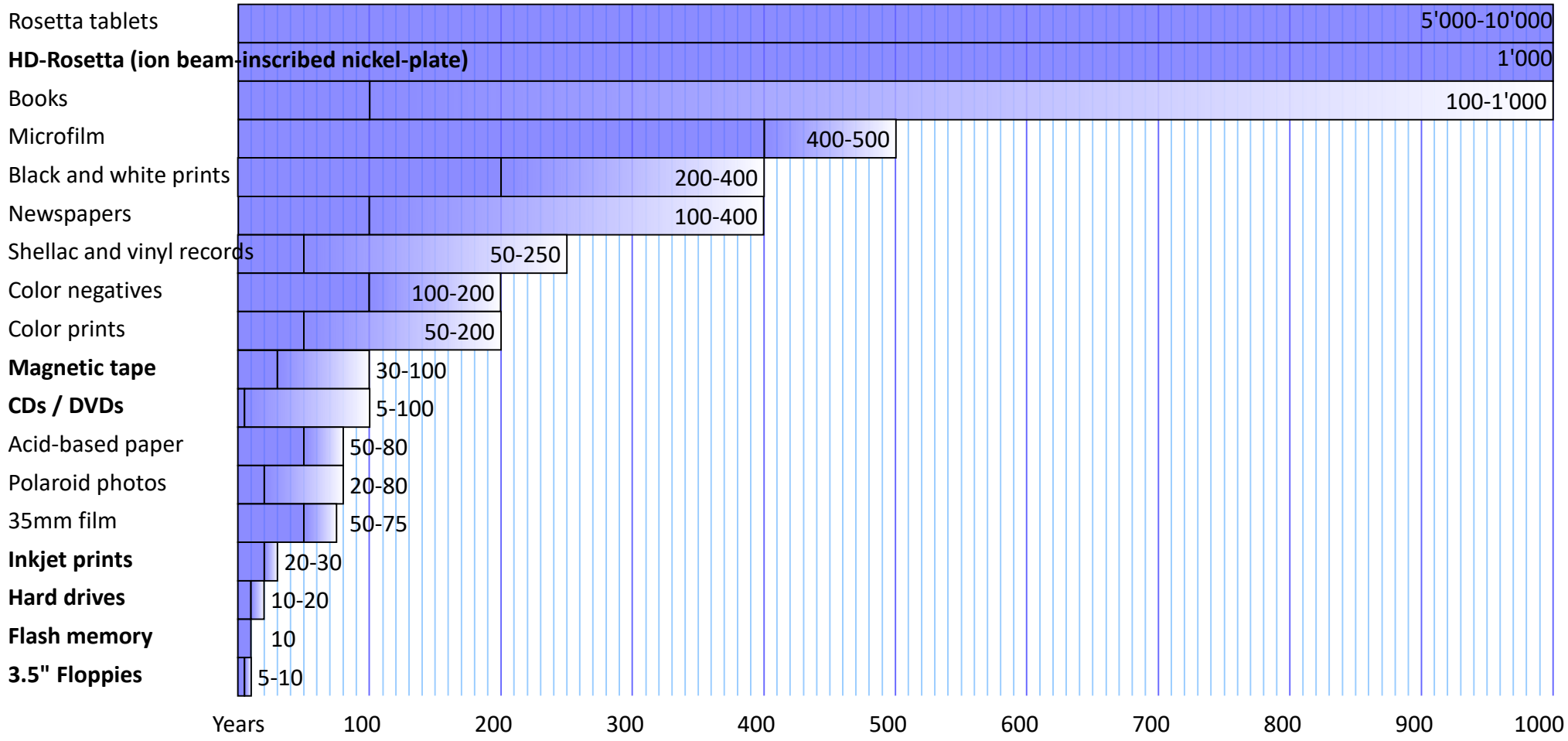
- Il faut parcourir tout le support d'information avant d'arriver à une donnée stockée
- coût linéaire avec la taille du support

► Accès Aléatoire

(HD, Flash, mémoire)

- on peut directement accéder à une donnée à partir de son adresse.
- coût constant d'accès

Rappel: longévité des supports d'information



Source: Wired June 2002, p 062

Technologie de Stockage: Bandes

►Bandes magnétiques (similaires aux.. cassettes!)

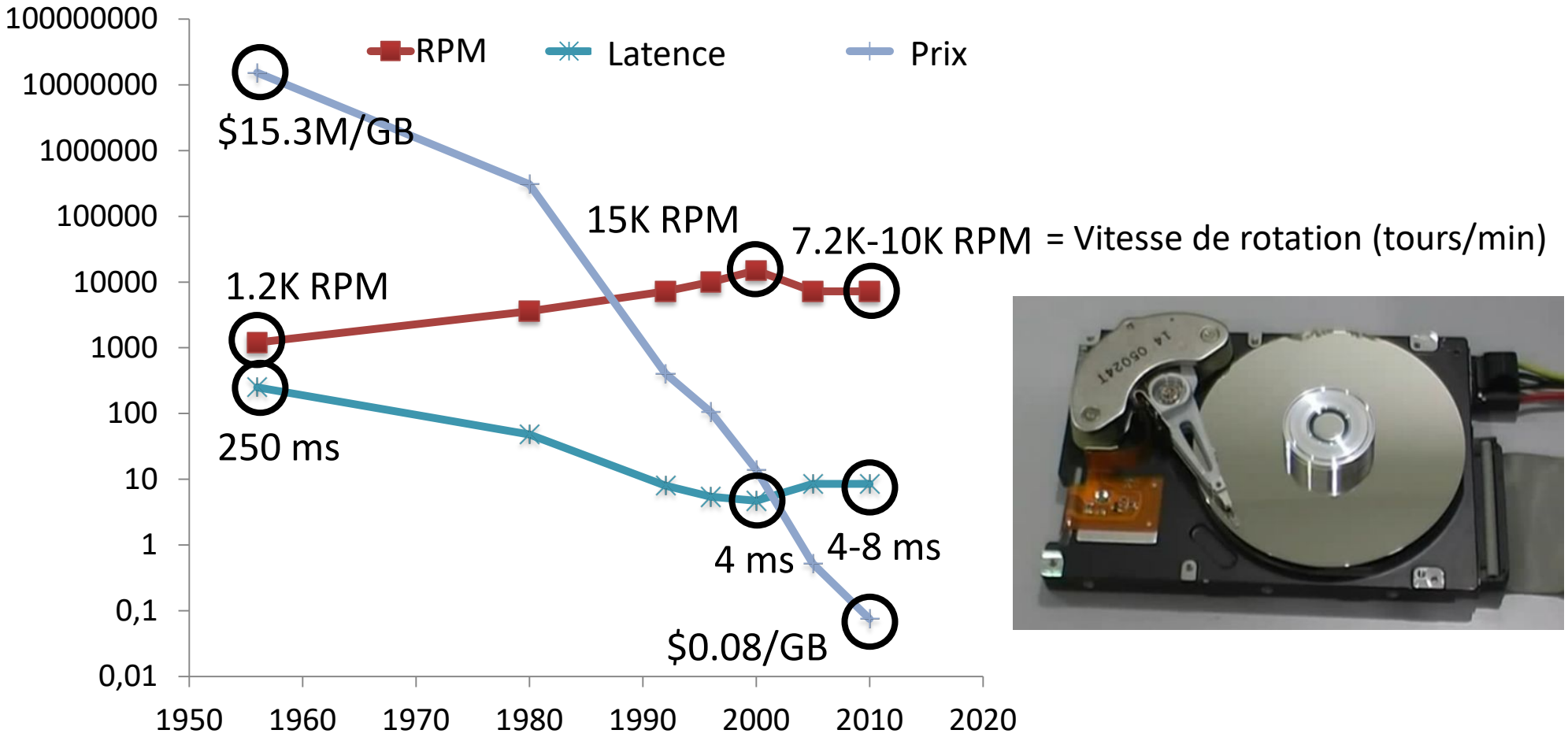
►Peuvent être écrites et lues mais seulement de manière *séquentielle*!

►Utilisées seulement pour l'archivage des données = backup

►Temps d'accès de l'ordre de secondes

Temps d'accès beaucoup plus grand si on veut se déplacer à une position arbitraire.

Technologie de Stockage: Disques (tendances)



Les disques deviennent de moins en moins chers!
MAIS pas plus rapides!

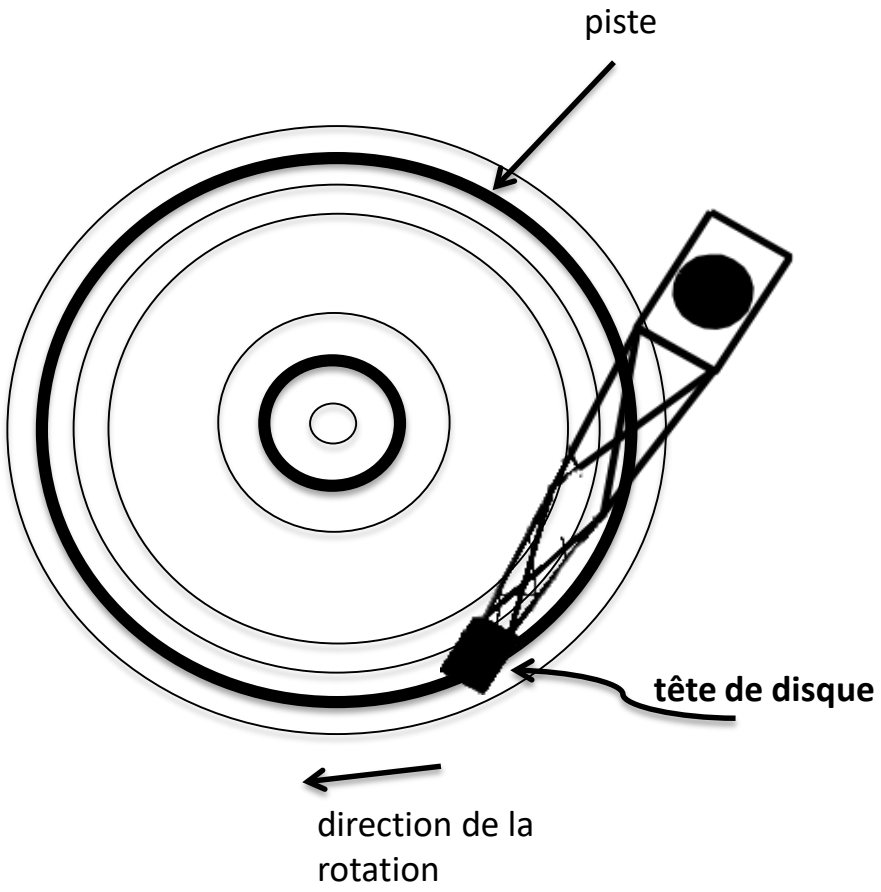
Technologie de Stockage: Disques

►Principal avantage des disques aux bandes: *accès aléatoire* vs. *séquentiel*

►Les données sont stockées et récupérées en unités qui s'appellent blocs de disque ou pages (typiquement 4KB ou 8KB)

►Le temps de récupérer une page du disque varie et dépend du placement de cette page sur le disque, ce qui n'est pas le cas avec la RAM.

- par conséquent, le placement relatif des pages sur le disque a un impact important sur la performance d'un logiciel de gestion de base de données.



Les écritures/lectures aléatoires sont 10 fois plus lentes que les écritures/lectures séquentielles

Accès à un bloc du disque

►Le temps d'accès (lecture/écriture) à un ***bloc du disque*** (équivalent à une ***page de mémoire***) est la somme de:

- *temps de recherche* (déplacer les bras pour positionner la tête du disque sur la piste)
- *délai rotationnel* (attendre la rotation du bloc sous la tête)
- *temps de transfert* (déplacer des données vers / à partir de la surface du disque)

Optimiser la performance des disques

- **cache** du disque dans une zone de la mémoire centrale
  - principe similaire à celui déjà vu entre mémoire on-chip et off-chip
  - quelques différences détaillées plus loin (méta-données)

- **Ordonnancement** du disque pour réduire le temps de recherche
  - Principe: 1) ranger les requètes d'accès dans une file d'attente
    2) mise en œuvre d'une politique d'ordonnancement

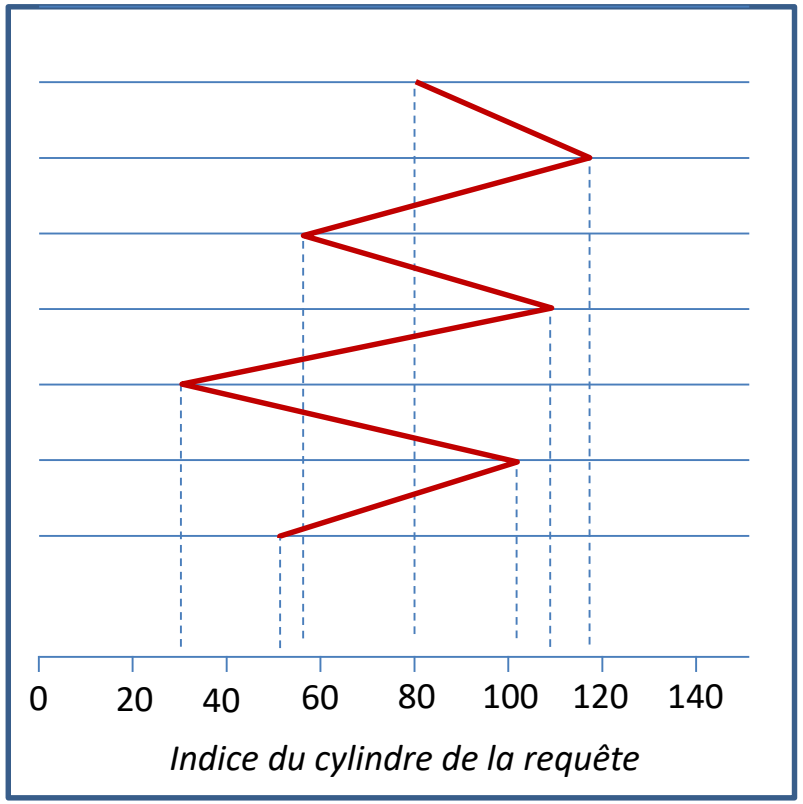
  - Exemple 1 : ***First Come First Served*** (la plus simple, peu performant)

    -> on retire toujours de la file d'attente la requête en tête de file

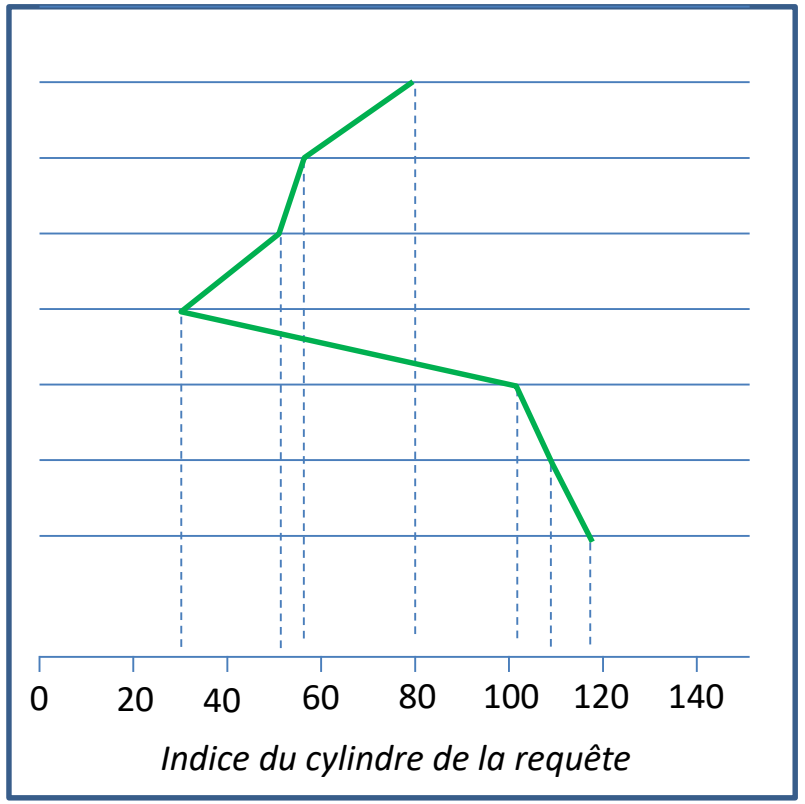
  - Exemple 2 : ***Shortest Seek Time First*** (SSTF). Exploite l'information du cylindre sur lequel se trouve le bloc pour minimiser la distance entre accès successifs

    -> on retire de la file d'attente la requête pour le clyindre le plus proche de celui sur laquelle la tête se trouve

Optimiser la performance des disques(2)

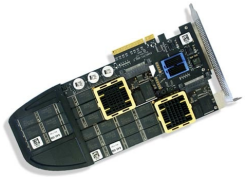


Politique First Come First Served
Nombreux va-et-vient entre cylindres



Politique Shortest Seek Time First
La distance intercylindre est minimisée

Technologie de Stockage: Flash

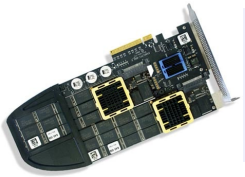


- Puces Flash utilisées >20 years

- Flash a évolué
  - Clés USB
  - Stockage dans des appareils mobiles
  - Flash disques pour consommateurs et entreprises (SSD)

- Flash est utilisé comme…
  - Stockage principal
  - Accélérateur/facilitateur (cache spécialisé, appareil de logging)

Technologie de Stockage: Flash



- Stockage secondaire *ou* couche du cache.

- Les données sont organisées en *pages* (typiquement de 512 octets comme sur les disques) et les pages sont organisées en *blocs (ex: = 64 pages).*
- Avantage principal par rapport aux disques: les *lectures aléatoires* sont aussi rapides que les lectures *séquentielles*.
- *Comme la RAM*, le temps pour récupérer une page n'est pas lié à l'emplacement de cette page sur la mémoire flash.

- **MAIS:** écritures (plus) lentes.
- ***Pour ré-écrire une page, il faut effacer/re-écrire le bloc entier***
  - politique *d'égalisation du nombre d'effacement* car, contrairement à un disque, le nombre d'effacements est limité.

Technologie de Stockage: Flash vs HDD

HDD

- ✓ Capacité: large et peu coûteux
- x Lectures aléatoires inefficace

Mémoire Flash

- x Capacité: petite et chère
- ✓ Lectures aléatoires très efficaces

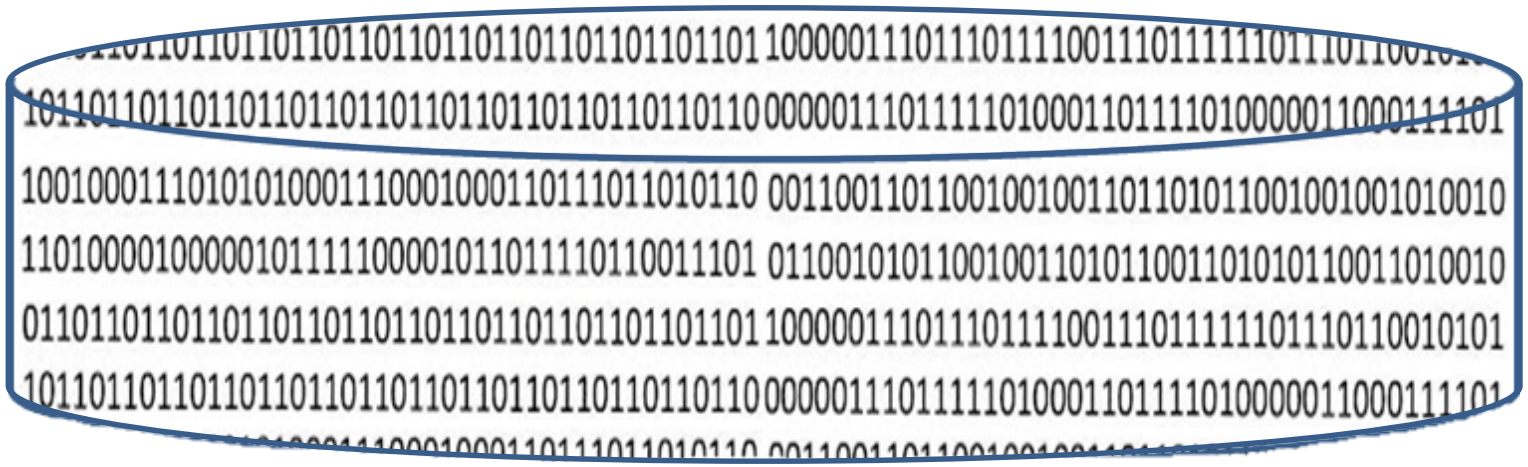
2. Structuration du stockage des données

- Dans la hiérarchie de la mémoire on *calcule*

- Dans la hiérarchie du stockage on *stocke* et on *récupère* des données
  - Nous avons besoin de structure dans le stockage des données
  - Types de structures de stockage: séquentielles, hiérarchiques, relationnelles
  - Identification, localisation, et accès à des données stockées

Le besoin de structure dans le stockage de données

►Imaginons un disque ou autre support sans aucune structure

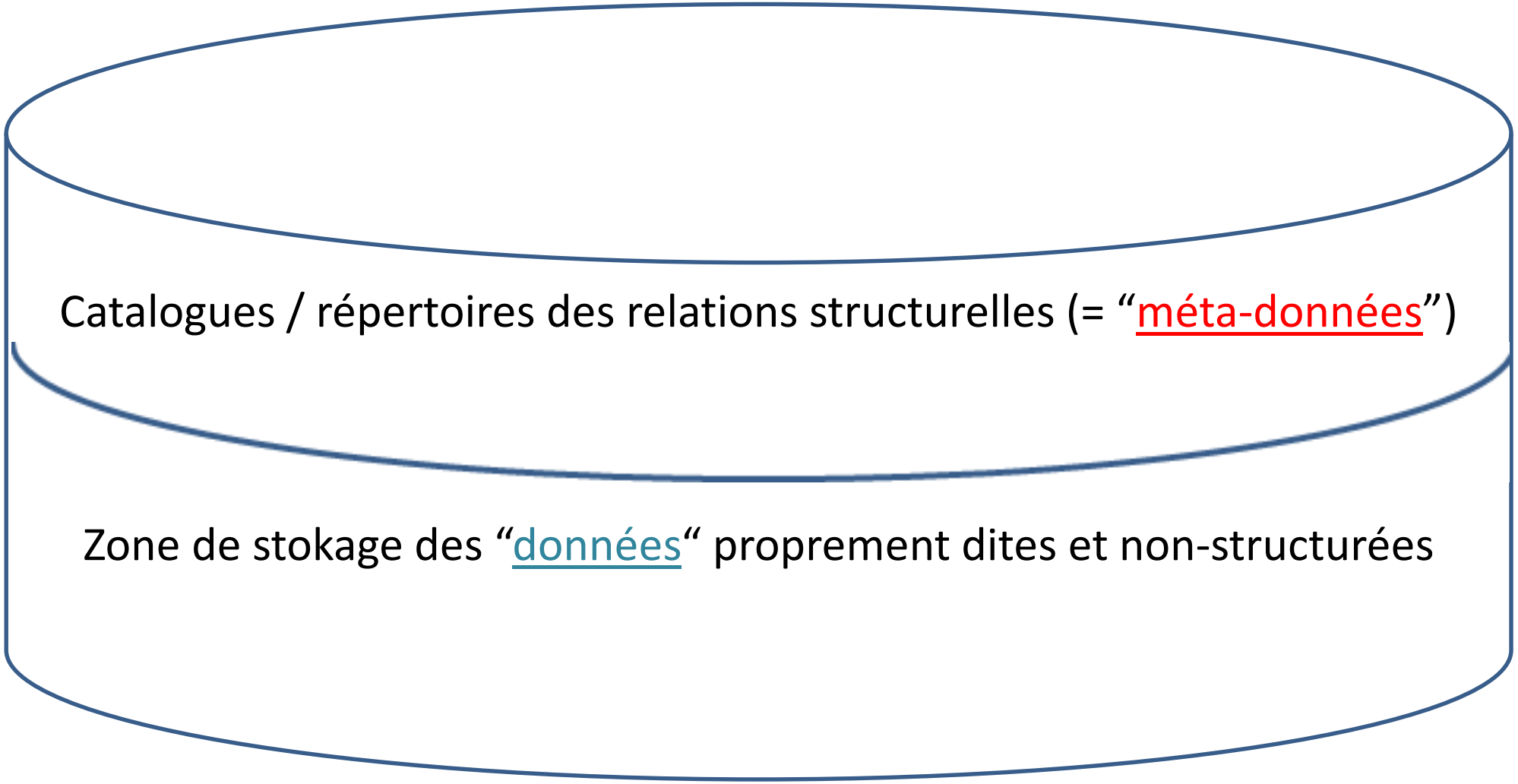


►Comment savoir s'il est plein ou vide? Comment y retrouver une information qu'on cherche?

Autant chercher une épingle dans une botte de foin

- **Même Google a besoin de structure pour retrouver ce qu'on lui demande !!**

Principe de base de la structuration des données stockées

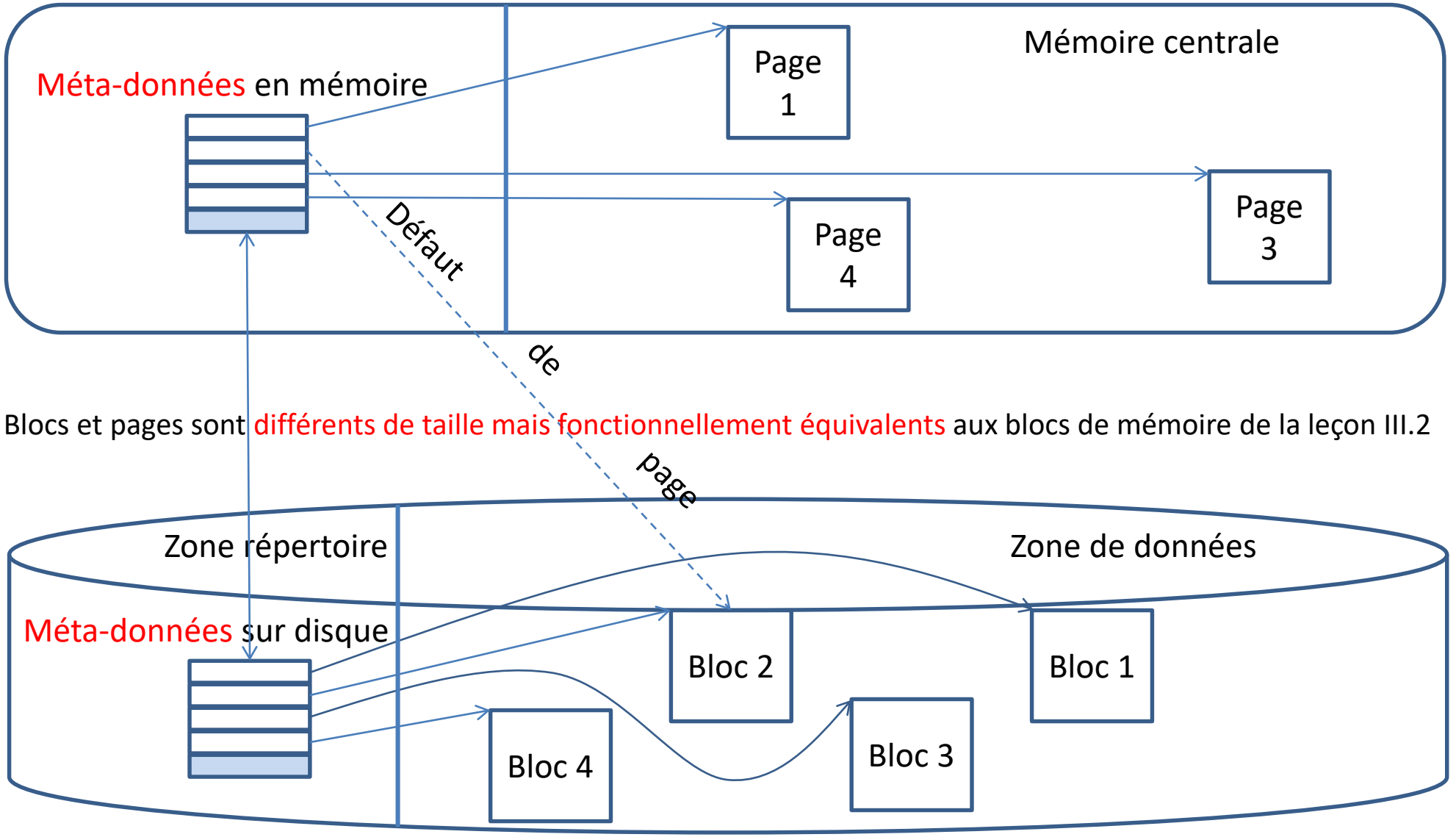


Exemple d'organisation en Données-Métadonnées
Gestion d'un fichier sur disque et en mémoire centrale

► Organisation physique des fichiers:

- Données: les fichiers sont décomposés en blocs de disque ou des pages
- Méta-Données: les adresses de ces pages se trouvent dans une table de page
- pour récupérer une partie des données en mémoire centrale, nous avons besoin de l'adresse physique: le numéro de page et le décalage dans la page de cette donnée
- Une partie de la table de pages et quelques-unes des pages de fichiers sont dans la mémoire selon les besoins (voir transparent suivant)

► Risque de *fragmentation* de l'espace disque au cours du temps



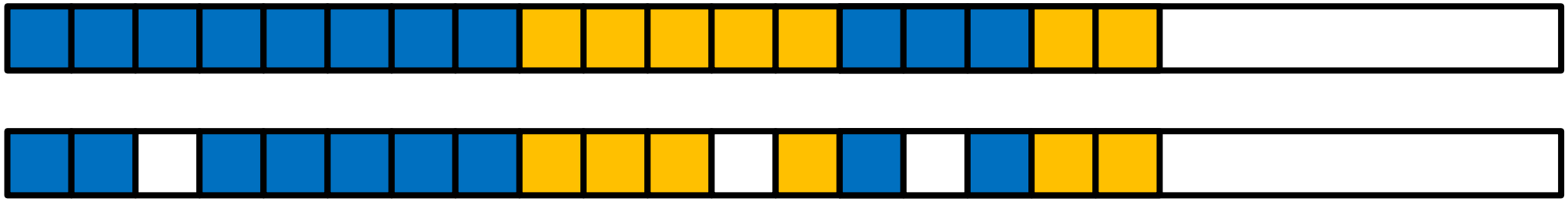
Blocs et pages sont différents de taille mais fonctionnellement équivalents aux blocs de mémoire de la leçon III.2

Fragmentation d'un fichier sur disque

► A sa création les blocs de données d'un fichier sont consécutifs dans la zone des données, s'il existe de l'espace disponible (ex: fichier1 en bleu et fichier2 en orange)

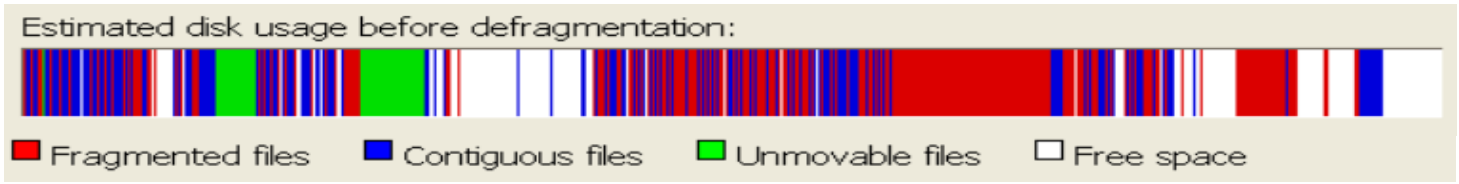
► Le fichier peut subir de nombreuses manipulations (ajouts, retraits)

- Un ajout se traduit par l'occupation de blocs là où il y a de la place, sans bouger les blocs pré-existants
- Un retrait se traduit la libération de blocs
- (ex: on ajoute des blocs à fichiers puis à fichier2, puis on en enlève à fichier1 puis fichier2, etc…)



► Conséquence: les blocs de données d'un même fichier ne sont plus consécutifs

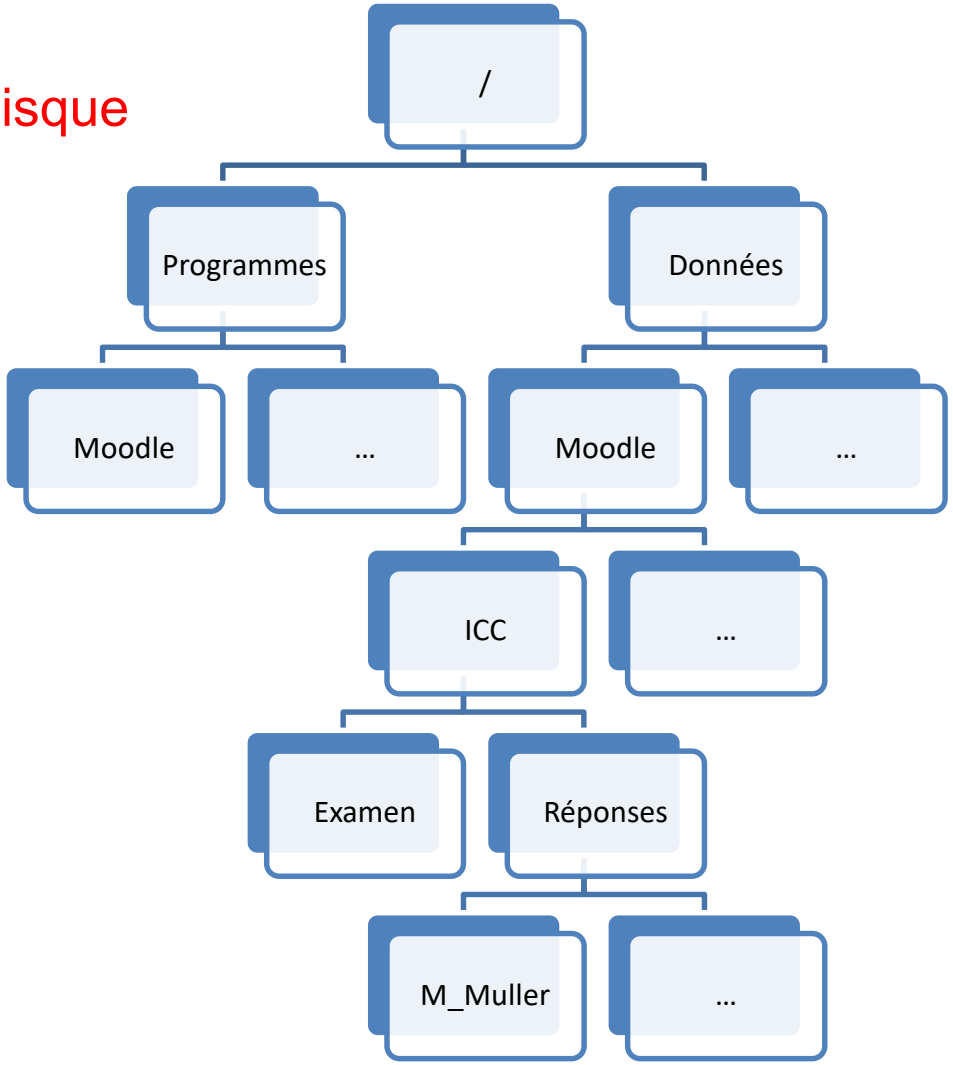
- => fragmentation => temps d'accès plus long / le système d'exploitation doit défragmenter le disque



Système de fichiers hiérarchiques – Accès au disque

► Comment l'ordinateur retrouve-t-il
l'adresse de stockage d'un fichier sur disque
à partir du nom de ce fichier ?

► Chaque **répertoire** contient
les adresses de stockage de tous les
répertoires et fichiers qu'il contient



Cf série 1 de C++ pour l'organisation
des fichiers à l'aide de repertoires
et l'accès aux fichiers à l'aide de l'indication
d'un chemin absolu (depuis la racine)
ou relative (depuis le repertoire courant)

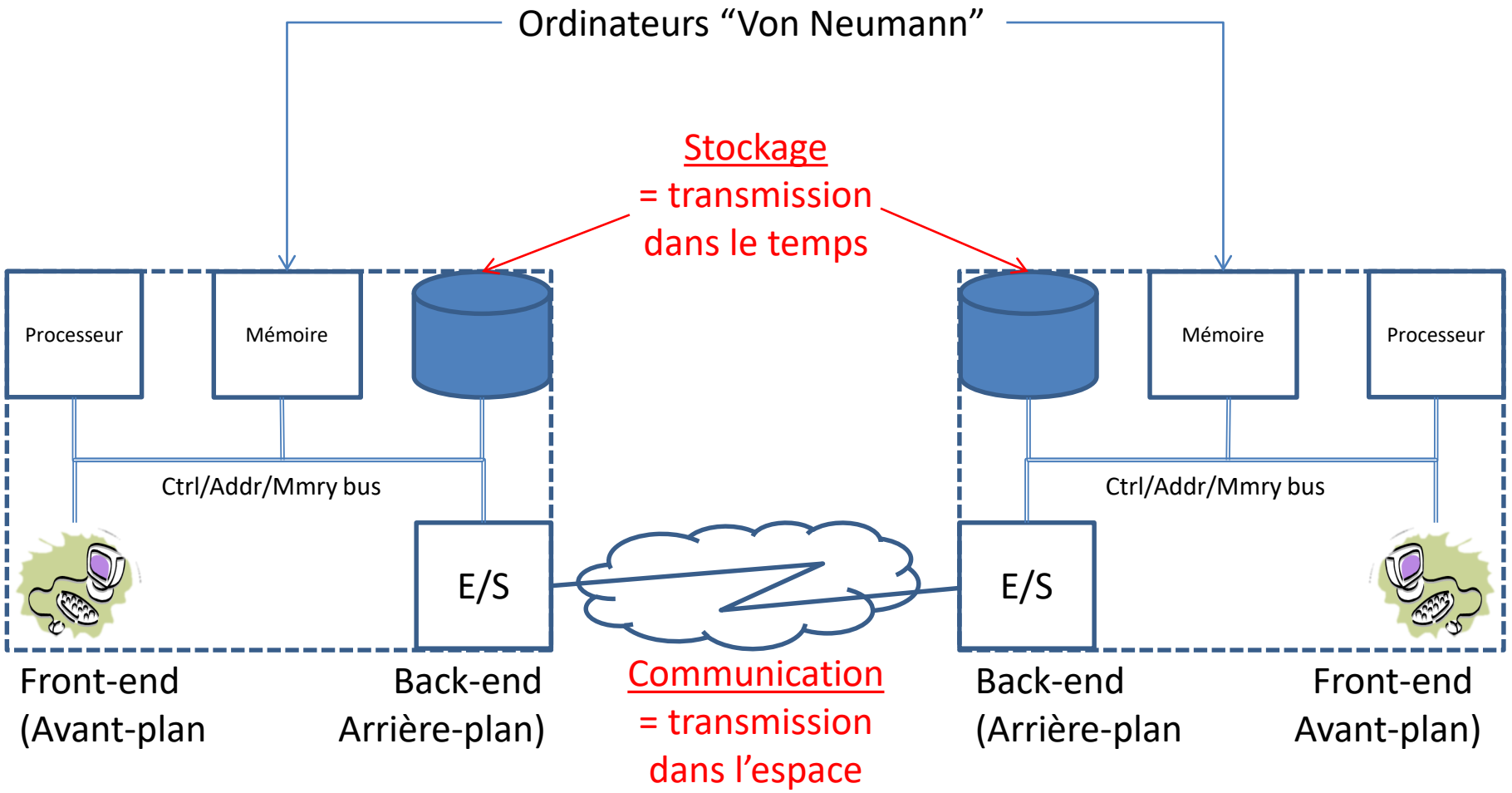
Information, Calcul et Communication

Leçon III.3.2: Réseaux

Faculté Informatique et Communications

A. Ailamaki, P. Janson, W. Zwaenepoel

Stockage et Communication sont deux problèmes similaires



Plan

- La notion de protocole de communication
- La commutation par paquets
- La structure d'un protocole et l'entête d'un paquet
- L'Internet
- La modularisation des protocoles
- Exemple: les protocols de l'Internet (TCP/IP)

Le besoin de structure dans la transmission de données

- Imaginons un signal arrivant d'un réseau, sans aucune structure

- Comment savoir si c'est un signal électrique réel ou juste du bruit ?
  Comment le convertir en un signal numérique binaire ?
  Comment savoir où il commence et où il finit ?
  Comment savoir d'où il vient ?
  Comment interpréter ce qu'il contient ?

un protocole de communication
structure la transmission de données

Un protocole de communication

- Un jeu de règles qui structure une communication

La notion de protocole

- Toute communication est gouvernée par un protocole

- Y compris, la communication entre êtres humains, p. ex.,
  - Salle de cours
  - Au téléphone
  - Message écrit

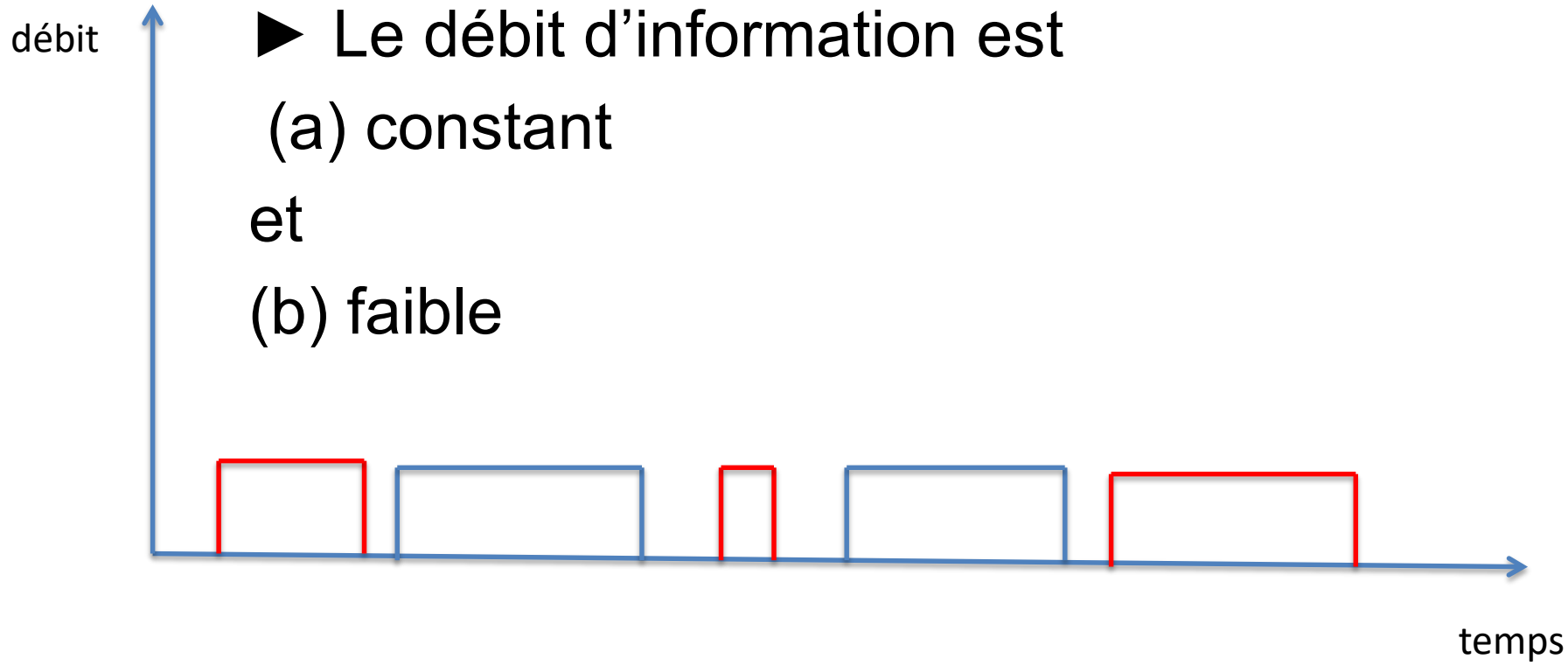
La différence avec la communication entre ordinateurs

- Un protocole entre êtres humains peut être vaguement défini
- Un protocole entre ordinateurs doit être fixé dans tous les details

Plan

- La notion de protocole de communication
- **La commutation par paquets**
- La structure d'un protocole et l'entête d'un paquet
- L'Internet
- La modularisation des protocoles
- Exemple: les protocoles de l'Internet (TCP/IP)

La communication par téléphone



► Le débit d'information est

(a) constant

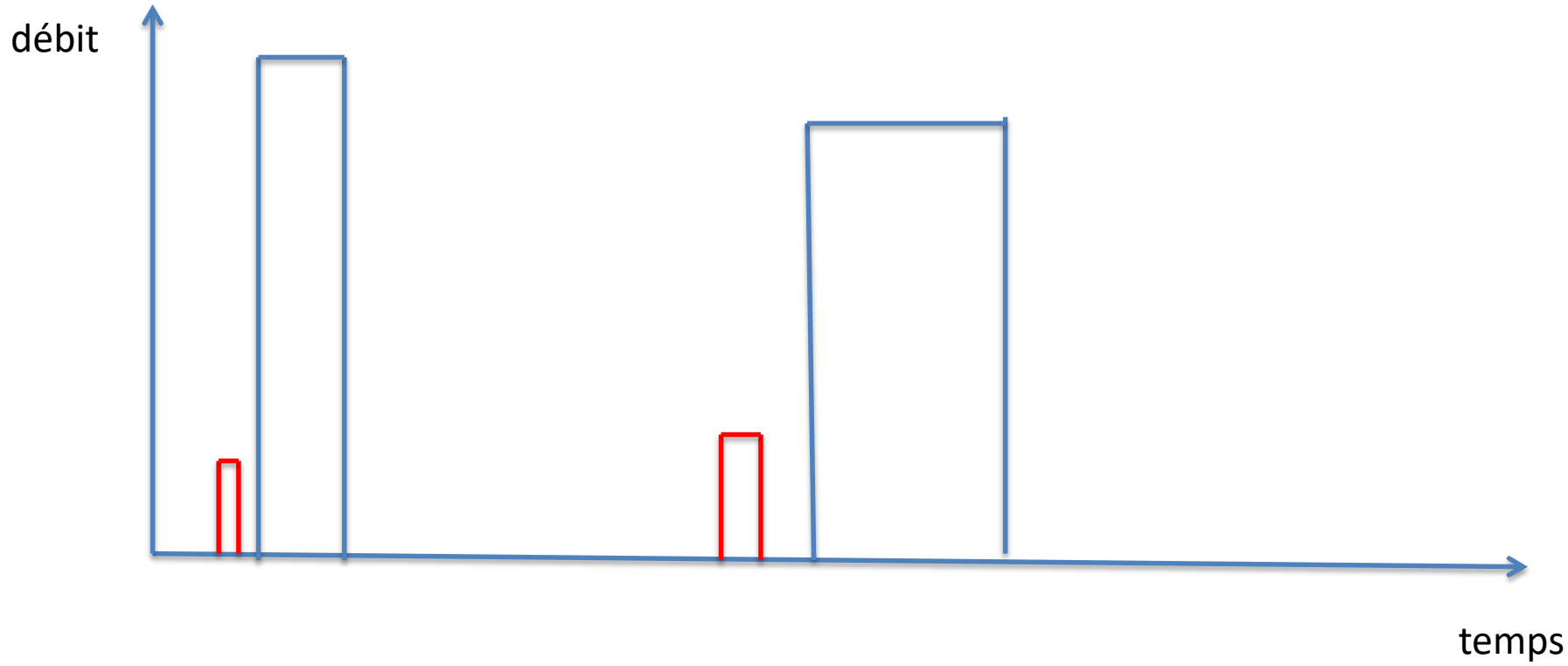
et

(b) faible

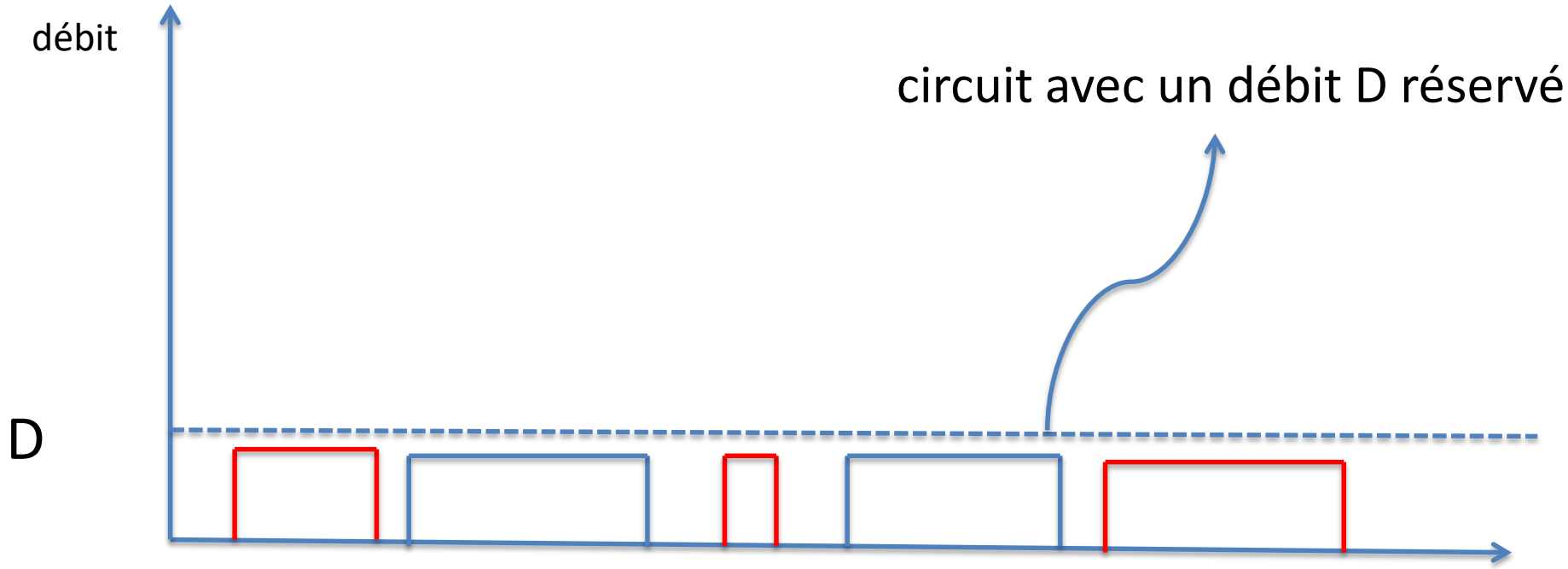
La communication entre ordinateurs

- L'information arrive en *rafale* (périodes de volume faible suivies par des hauts et soudains volumes d'information)
- Le débit est tres élevé quand il y a communication
- Le débit est nul autrement

Une situation typique de communication entre ordinateurs

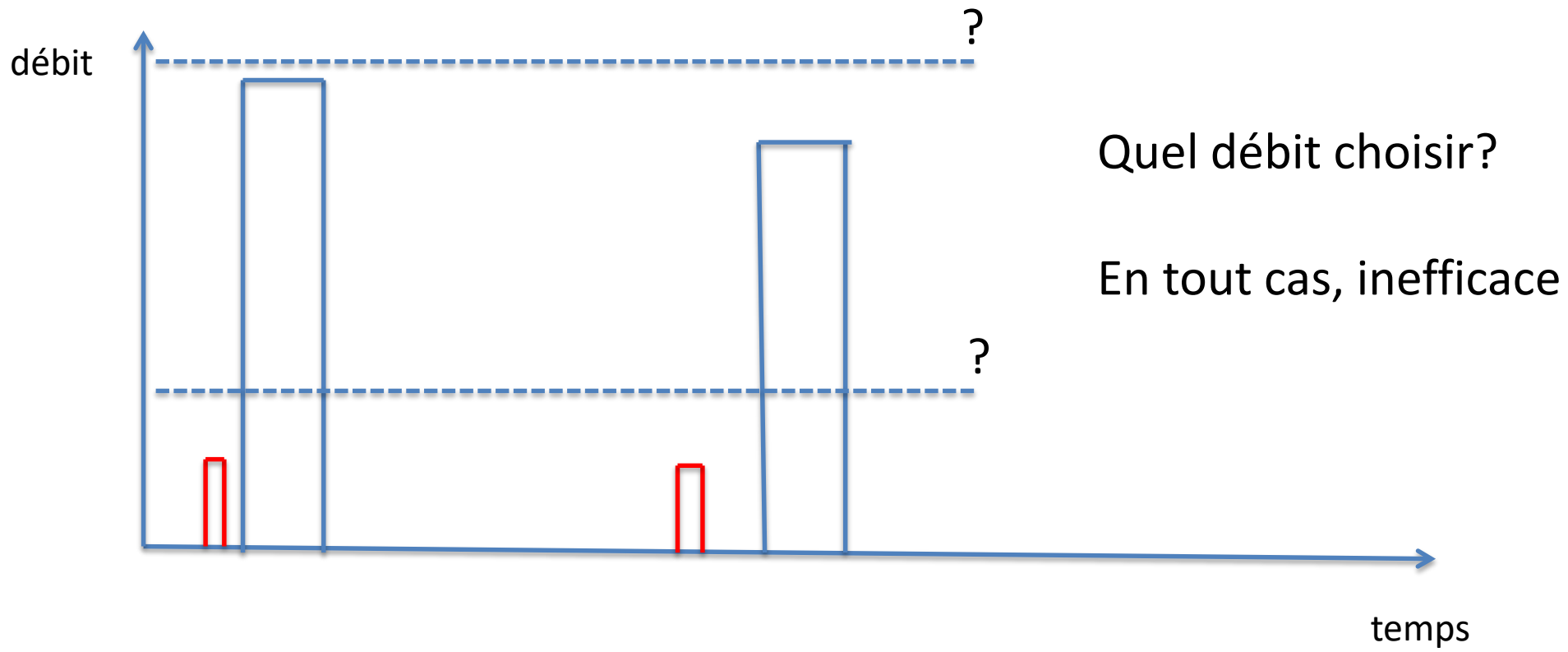


La commutation de circuits dans la téléphonie



- Réserver un "circuit" entre source et destination
- "Commutation de circuits"

Pas approprié pour la communication par ordinateur



Quel débit choisir?

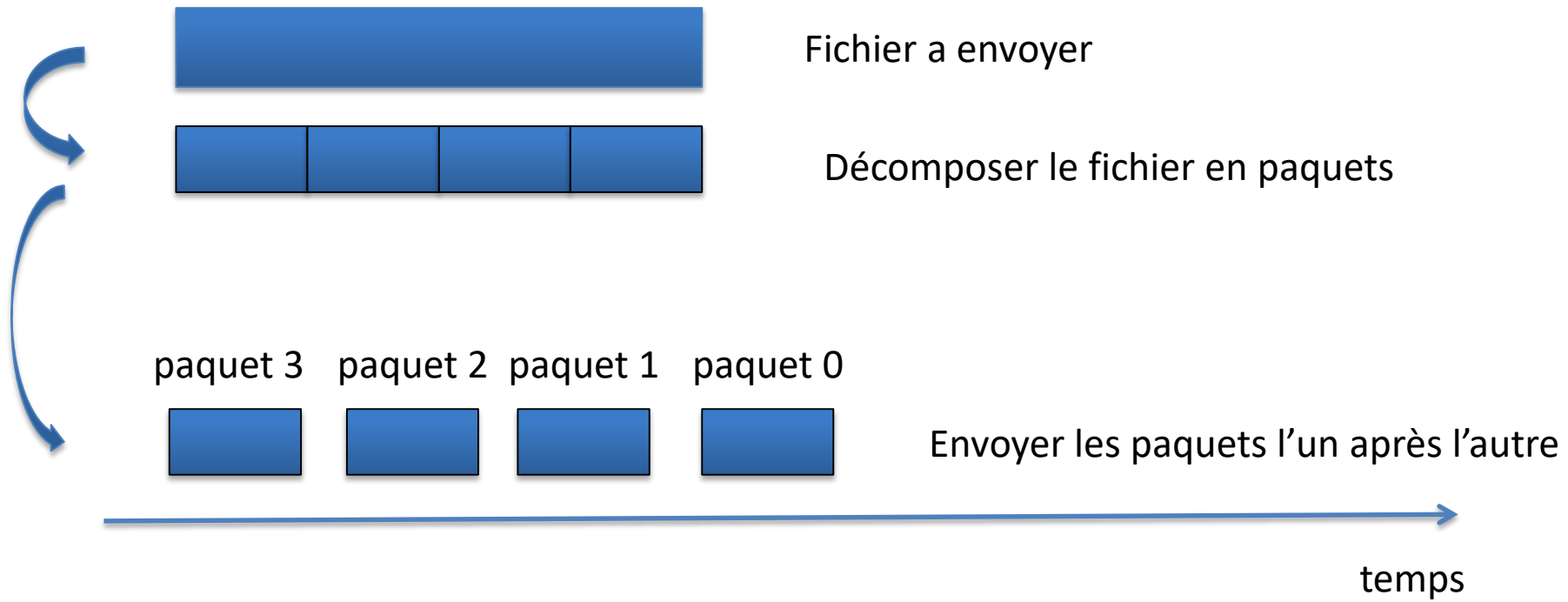
En tout cas, inefficace

►La réservation d'un circuit serait très inefficace

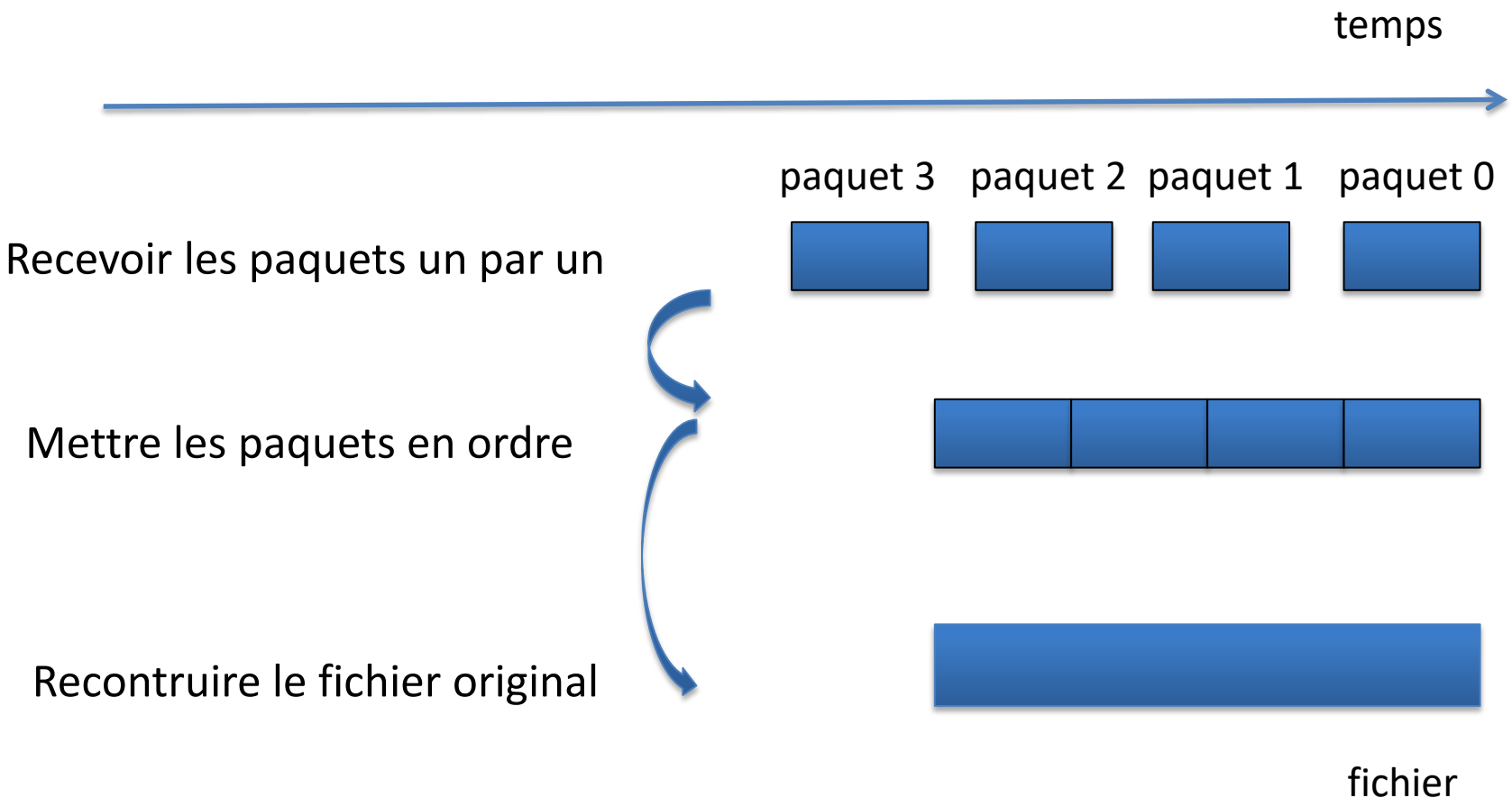
Commutation par paquets

- On découpe l'information en paquets
- On les envoie à destination, un paquet à la fois
- Modèle "La Poste"

Exemple: Comment envoyer un fichier



Exemple: comment recevoir un fichier



Commutation par paquets: problèmes!

- Des paquets peuvent être **perdus** en route
- Des paquets peuvent arriver en **désordre**
- On verra plus tard comment traîter ces cas

Commutation par paquets est omniprésente

- L'internet
- Presque tout réseau pour ordinateur
- Même la téléphonie (Skype!)
- Protocole: le jeu de règles
  - Exemple: WiFi, Ethernet, …

Plan

- La notion de protocole de communication
- La commutation par paquets
- **La structure d'un protocole et l'entête d'un paquet**
- L'Internet
- La modularisation des protocoles
- Exemple: les protocoles de l'Internet (TCP/IP)

Un exemple de protocole

- Disons que l'ordinateur A veut
  - Envoyer un paquet à l'ordinateur B
  - Savoir si le paquet est arrivé

Le principe d'un acquittement (ACK)

►B envoie à **A** un paquet de type spécial qui s'appelle paquet **d'acquittement** (ACK qui vient du mot anglais acknowledgement)

►ACK signifie "c'est ok, je l'ai recu"

Attention!

- **A** peut confondre un paquet d' acquittement avec un paquet des données!

- Il faut maintenant distinguer
  - Les paquets avec des **données**
  - Les paquets **d'acquittement**

- Comment faire?
  - On met dans le paquet une information de "type"

Inclusion du type de paquet

type = paquet de **données** ou paquet **d'acquittement**

Mais alors ?

- Que fait A quand l'acquittement n'arrive pas?

Le principe de la retransmission

- Après un temps t, A retransmet le paquet à B
- Jusqu'à ce qu'un acquittement soit reçu
- Apres un nombre d'essais max, A abandonne

Pendant ce temps, du coté de la destination...

- Comment B sait qu'un paquet est :
  - Une retransmission ?
  - Un nouveau paquet ?
- Il faut **identifier** le paquet
- On met dans le paquet un **numéro de séquence**

Inclusion du numéro de séquence

seq# = 0, 1, 2, …

Adressage

- A doit indiquer que le paquet est destiné à B
- A doit indiquer qu'il est à l'origine du paquet
- Mettre les adresses de A et B dans le paquet

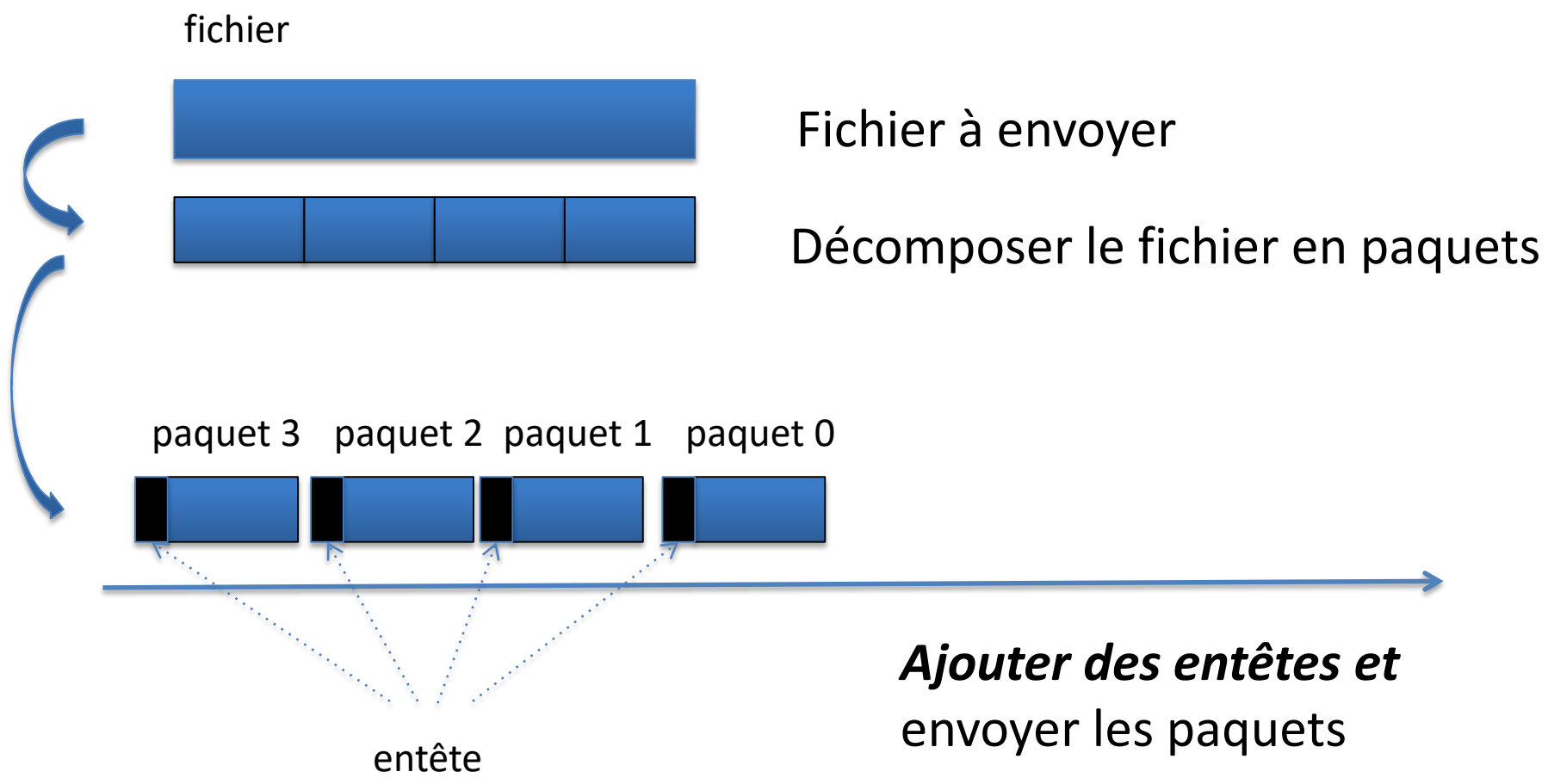
Inclusion d'adressage

L'entête (header) d'un paquet

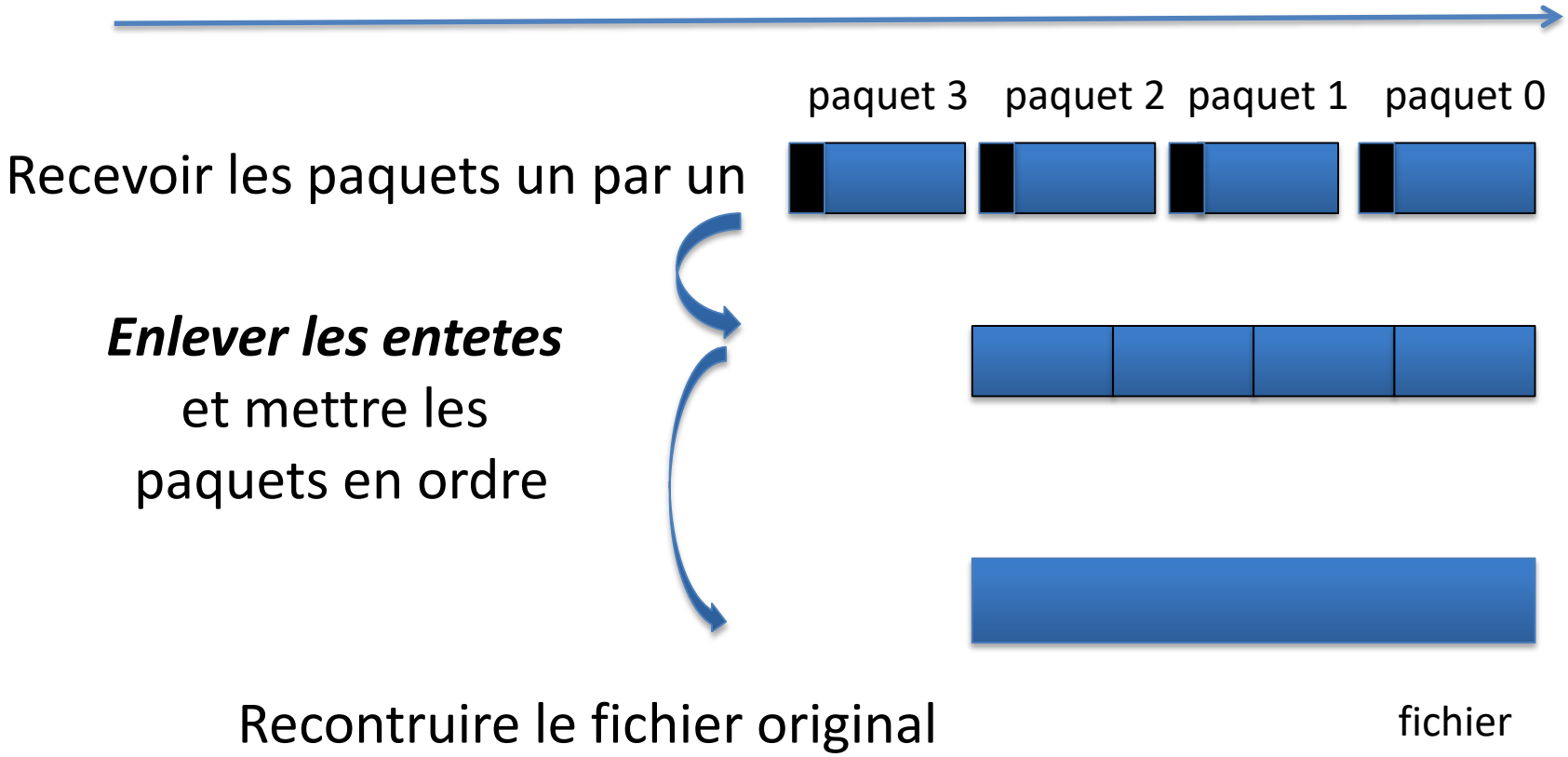
- Chaque paquet consiste en :
  - Des **données**
  - D'un **entête**: type, numéro de séquence, taille, adresse source et destination, etc.

Structure du paquet

Envoi d'un fichier en paquets



Réception d'un fichier en paquets



Opération d'un protocole

- A la transmission d'un paquet, le protocole…
  - reçoit les données
  - ajoute au paquet un entête
  - transmet le paquet

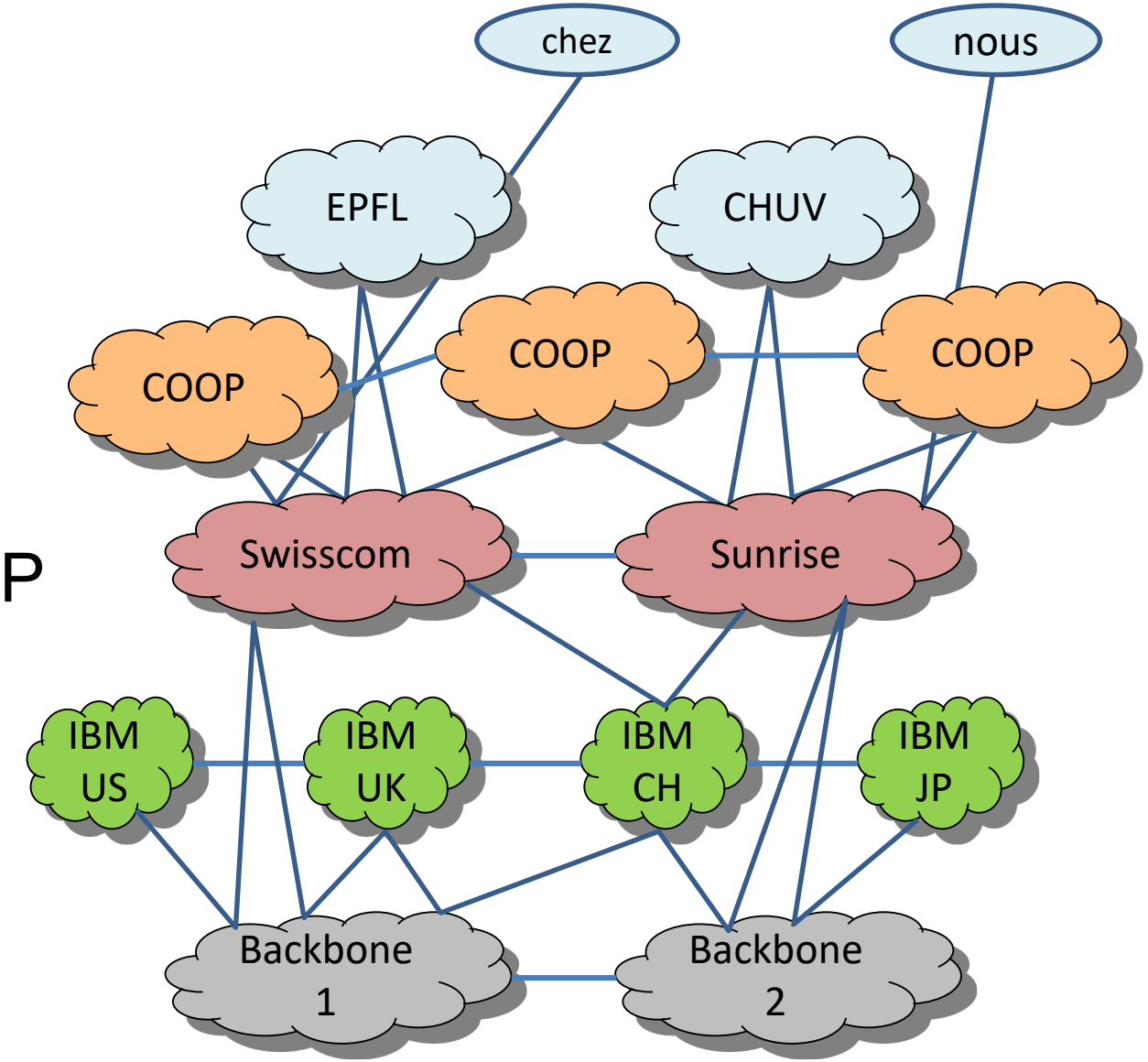
- A la réception d'un paquet, le protocole…
  - enlève l'entête du paquet
  - passe les données

Plan

- La notion de protocole de communication
- La commutation par paquets
- La structure d'un protocole et l'entête d'un paquet
- **L'Internet**
- La modularisation des protocoles
- Exemple: les protocole de l'Internet (TCP/IP)

L'Internet

- L'internet est composé
  - Des réseaux nationaux
  - Des réseaux des ISP
  - De réseaux "backbone"
  - …

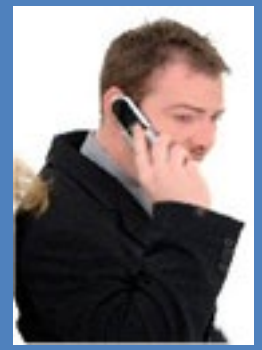


Comment gérer cette complexité?

- Chaque réseau a son propre protocol
  - WiFi, Ethernet, …
- Comment faire pour communiquer entre deux machines sur différents réseaux?
- Comment distinguer traffic Web d'autre traffic?
- Comment distinguer le traffic Web d'un utilisateur particulier?

Mise en oeuvre du Principe d'Abstraction
et de la décomposition top-down des problèmes
à l'échelle des systèmes de communications

Les couches = L'abstraction des protocoles
A chaque niveau son protocole – Exemple téléphonique

Application	 Conversation entre interlocuteurs 
Transport	 Connexion électronique entre Natels 
Réseau	 Routage entre antennes via commutateurs 
Lien	 Connexion entre votre Natel et sa station GSM 
Physique	 Conversion audio -> signal électrique sur Natel 

=> Chaque couche gère et abstrait les phénomènes de son niveau pour affranchir les autres couches de ces détails

Plan

- La notion de protocole de communication
- La commutation par paquets
- La structure d'un protocole et l'entête d'un paquet
- L'interface et le pilote réseau
- L'Internet
- **La modularisation des protocoles**
- Exemple: les protocoles de l'Internet (TCP/IP)

La modularisation: le but

- Un protocole peut devenir très compliqué
- La modularisation cherche à maîtriser cette complexité

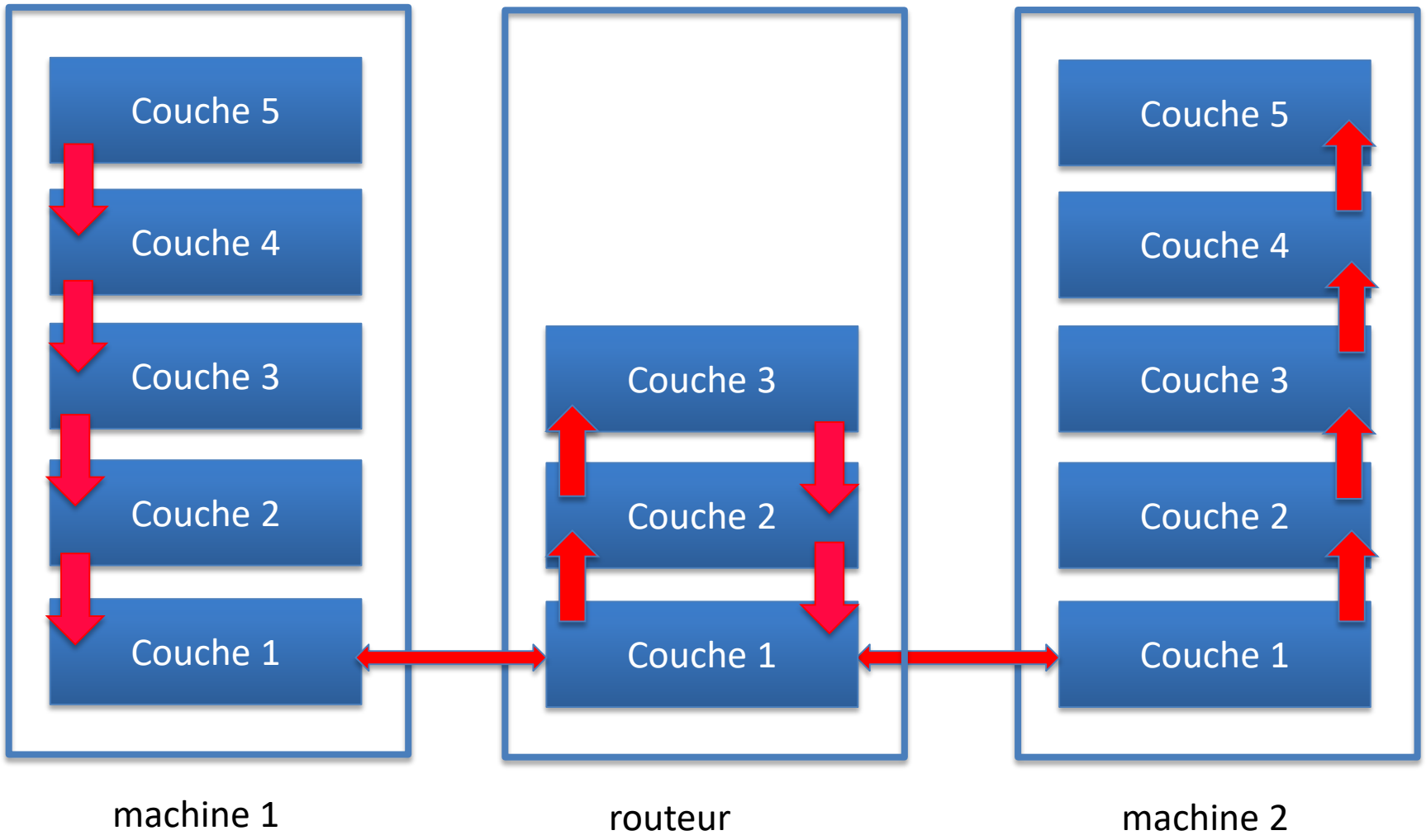
La modularisation: c'est quoi?

- Le protocole est divisé en couches
- Chaque couche a
  - Sa propre fonction
  - Son propre entête
  - Son propre logiciel
- Les couches se reposent l'une sur l'autre

Comment ça marche?

- Chaque couche existe sur chaque machine
- Du point de vue logique
  - Communication entre les couches du même niveau
- Du point de vue physique
  - Communication entre les couches sur une seule machine
  - Sauf au niveau 1

La réalité est plus compliquée



Le principe d'encapsulation - Transmission

- Chaque couche a son entête
- La fonction de la couche N:
  - Prend le paquet venant de la couche supérieure N+1
    (qui inclut les entêtes des couches N+1, N+2, …)
  - Met un nouveau entête pour sa propre couche
- La couche N>1 passe le paquet au niveau N-1
- La couche N=1 transmet le paquet sur le réseau

Plan

- La notion de protocole de communication
- La commutation par paquets
- La structure d'un protocole et l'entête d'un paquet
- L'Internet
- La modularisation des protocoles
- **Exemple: les protocoles de l'Internet (TCP/IP)**

Structure d'Internet – Protocoles

5. Application	Terminal interactif ex. SSH	Transfert de fichiers ex. FTP	Courrier électronique SMTP	Naviguer sur la toile HTTP	Le botin Internet DNS	Etc. ….
4. Transport	TCP		SSL / TLS		UDP	
3. Réseau	**IP** (adressage et routage)					
2. Lien	CSMA / CD			PPP	Trunk lines	
1. Physique	Wi-Fi	Ethernet	CATV	ADSL	Trunk lines	

Couche 3 – "Réseau"

- Entre deux machines n'importe où
- IP (Internet Protocol)
- Responsable pour le routage

La couche IP en plus de détail

- Envoyer un paquet
  - D'une machine quelconque
  - A une machine quelconque

IP - Adressage

- Une machine A a une adresse IP **unique**, IP_A

- Deux versions:
  - IPv4: 32 bits - 2^{32} (~4.10^9) systèmes
  - IPv6: 128 bits - ~$256.10^9.10^9.10^9.10^9$ systèmes

A veut envoyer un paquet à B

- La couche IP de A met IP_A et IP_B dans l'entête IP

Le problème du routage

- Comment trouver une voie de A à B?
- Comment trouver le chemin le plus court?

Routage IP

- Un “tableau de routage” pour chaque noeud
  - Pour chaque destination, quel chemin à suivre
  - Quelle est la distance à la destination par ce chemin

Exemple réseau

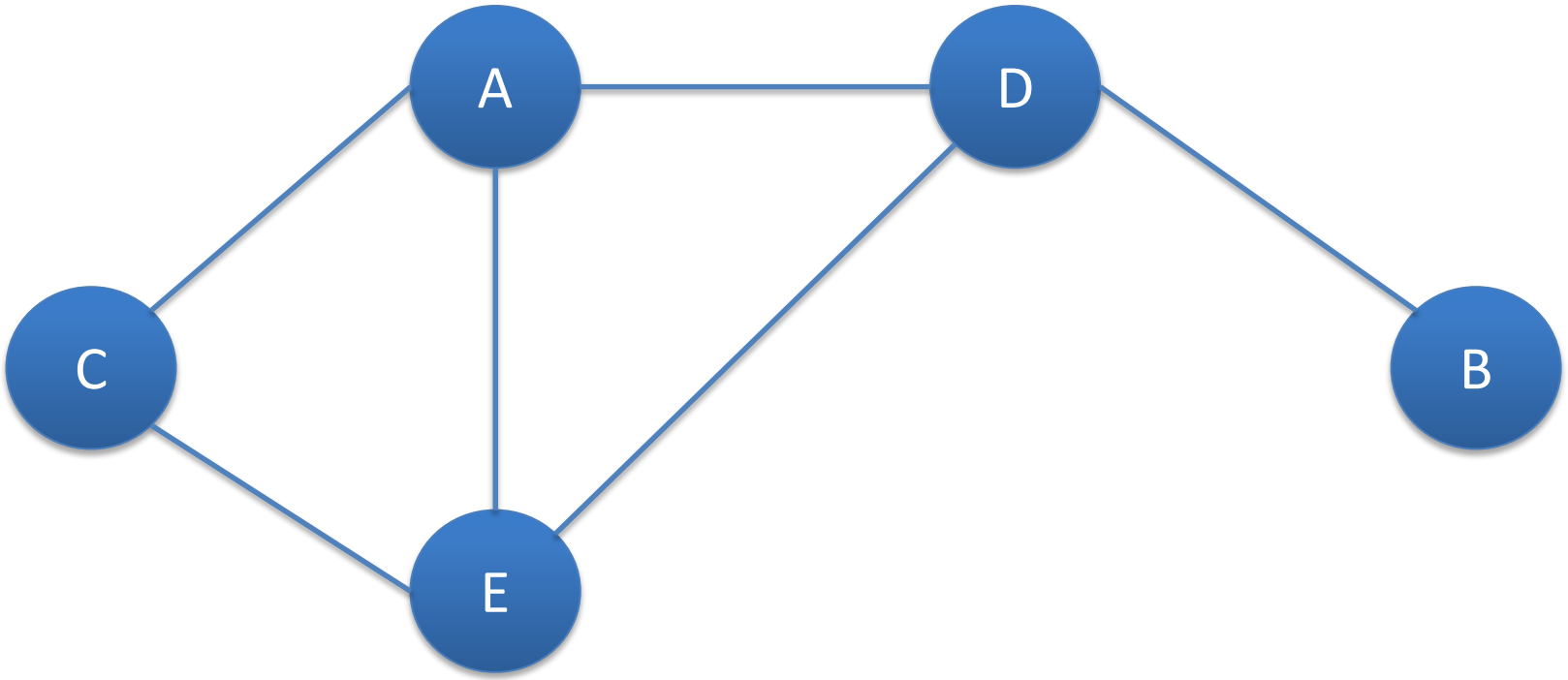


Table de routage de A

dest	voie	longeur
B	D	2
C	C	1
D	D	1
E	E	1

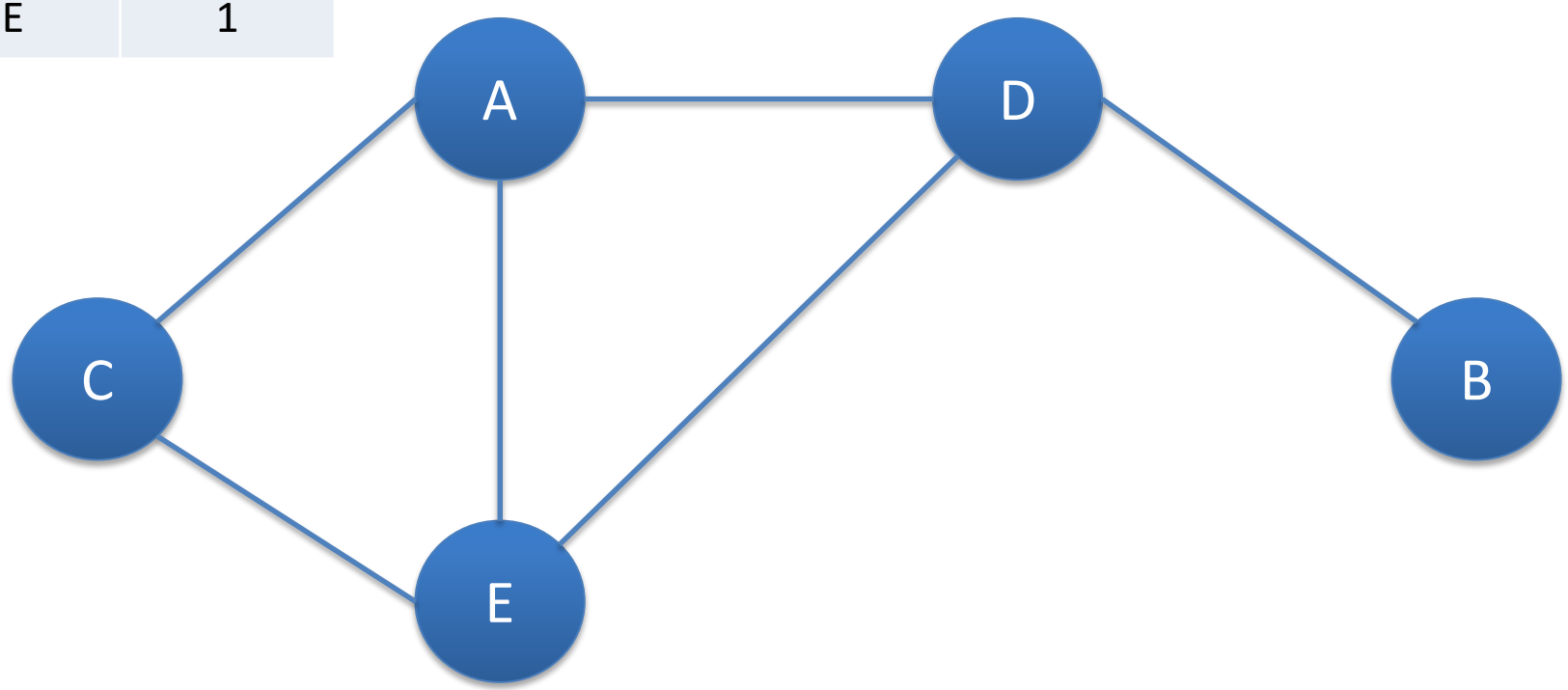


Table de routage de D

dest	voie	longeur
A	A	1
B	B	1
C	A/E	2
E	E	1

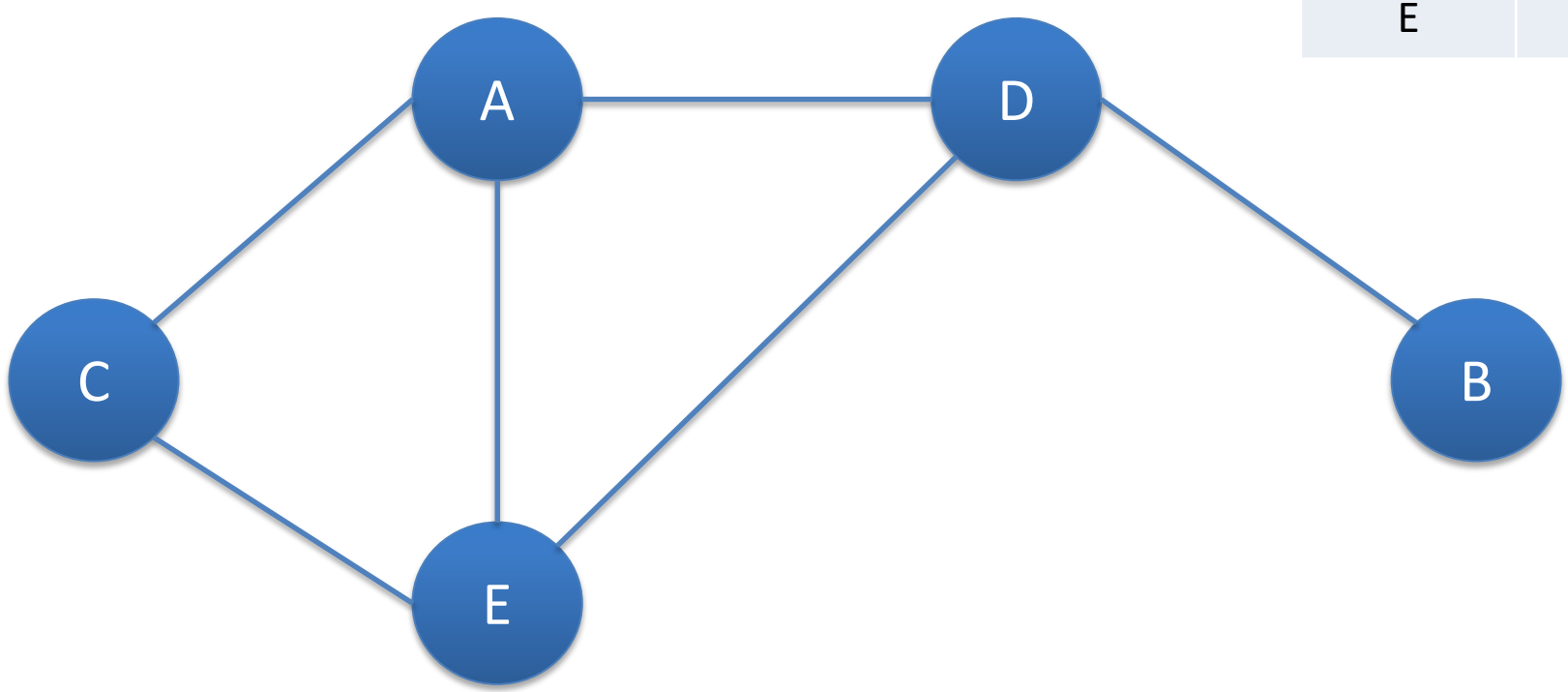
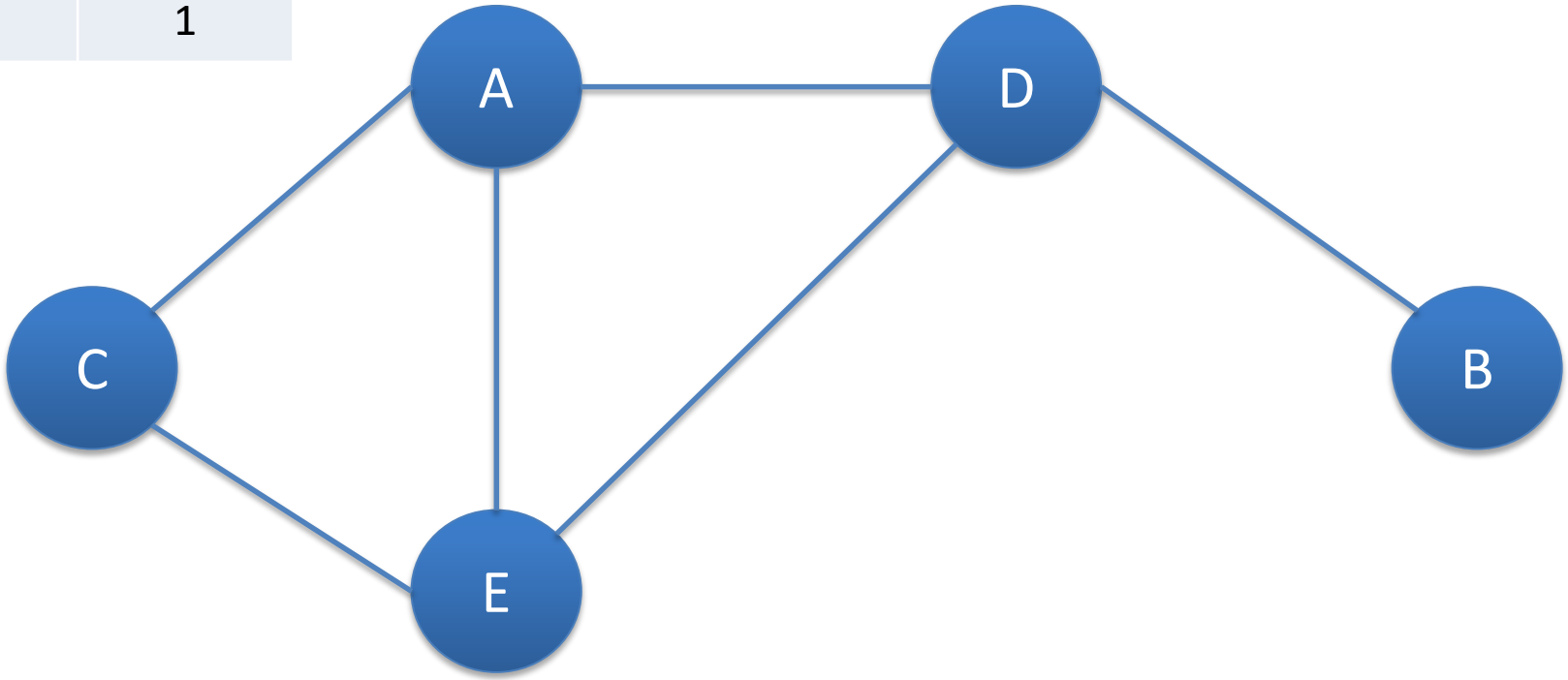


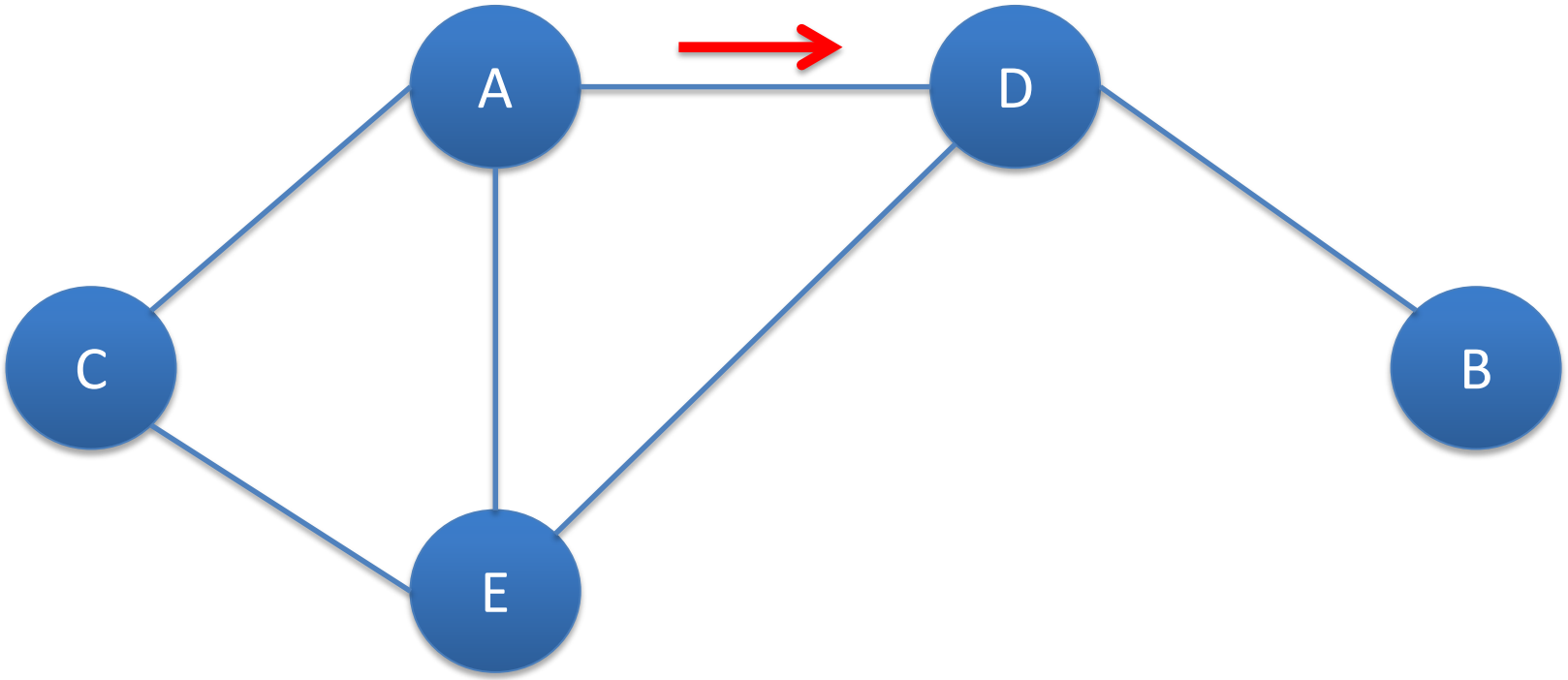
Table de routage de C

dest	voie	longeur
A	A	1
B	A/E	3
D	A/E	2
E	E	1



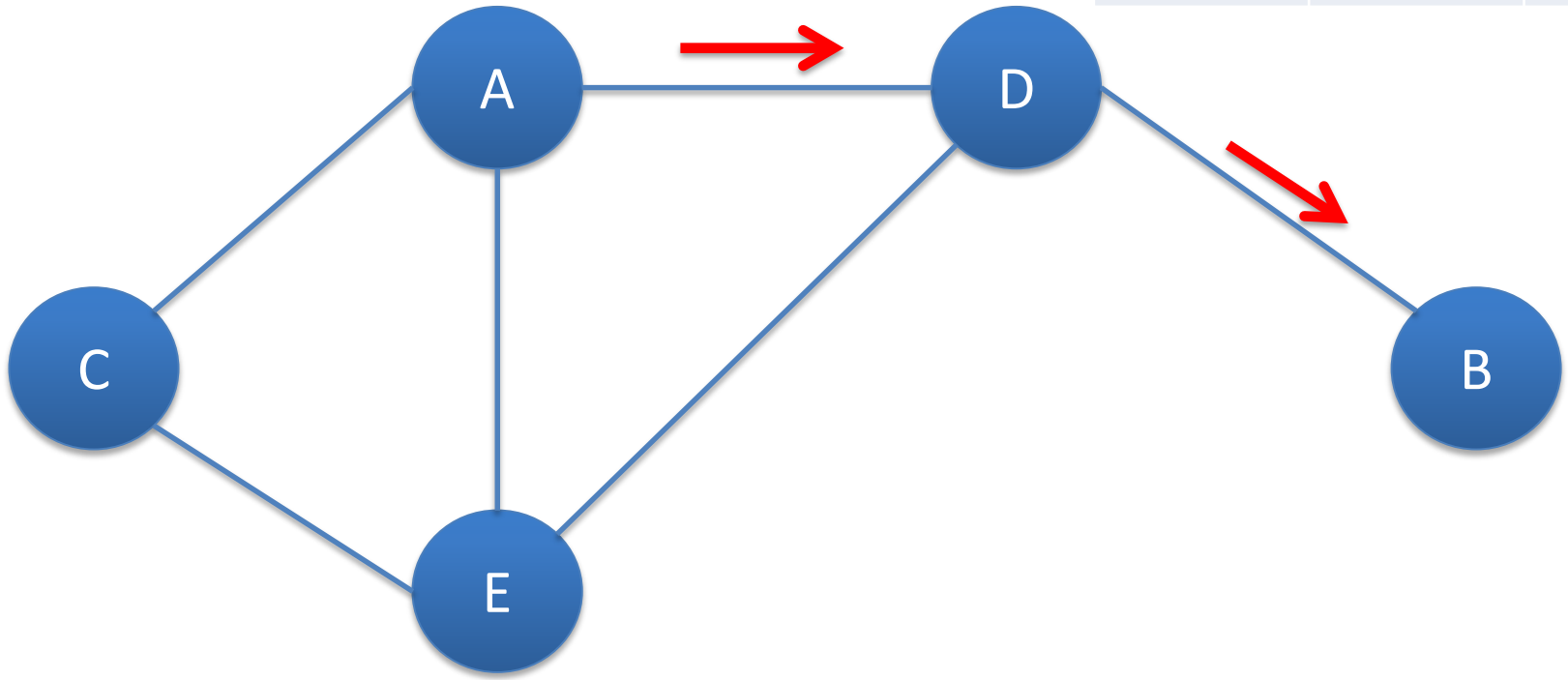
A veut envoyer un paquet à B

dest	voie	longeur
B	D	2
C	C	1
D	D	1
E	E	1



D forward le paquet à B

dest	voie	longeur
A	A	1
B	B	1
C	A/E	2
E	E	1



A chaque étape

►On regarde l'adresse IP de la destination

►On regarde le tableau de routage

►On envoie le paquet au noeud indiqué dans la table pour cette destination

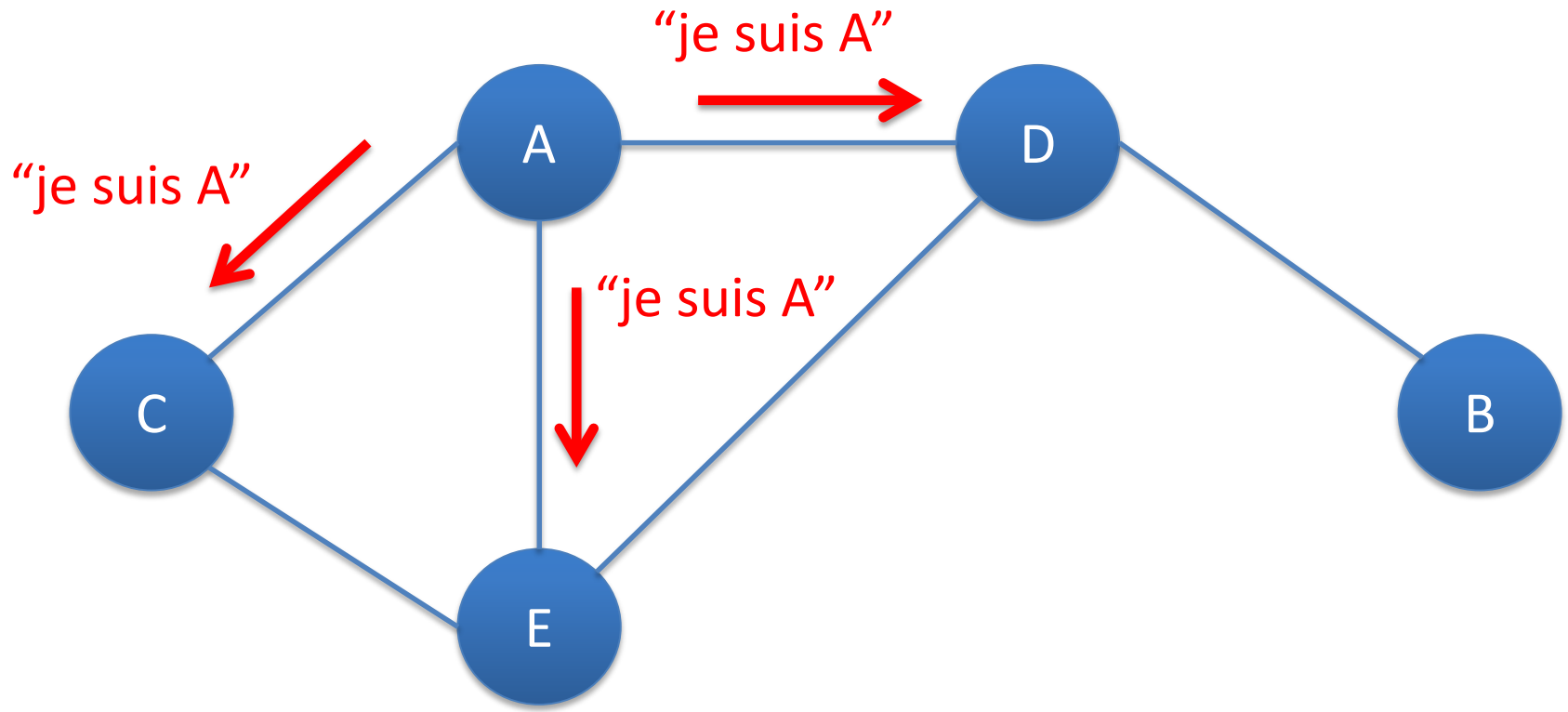
Comment construire le tableau de routage?

►Le routage se fait par “ouï-dire”

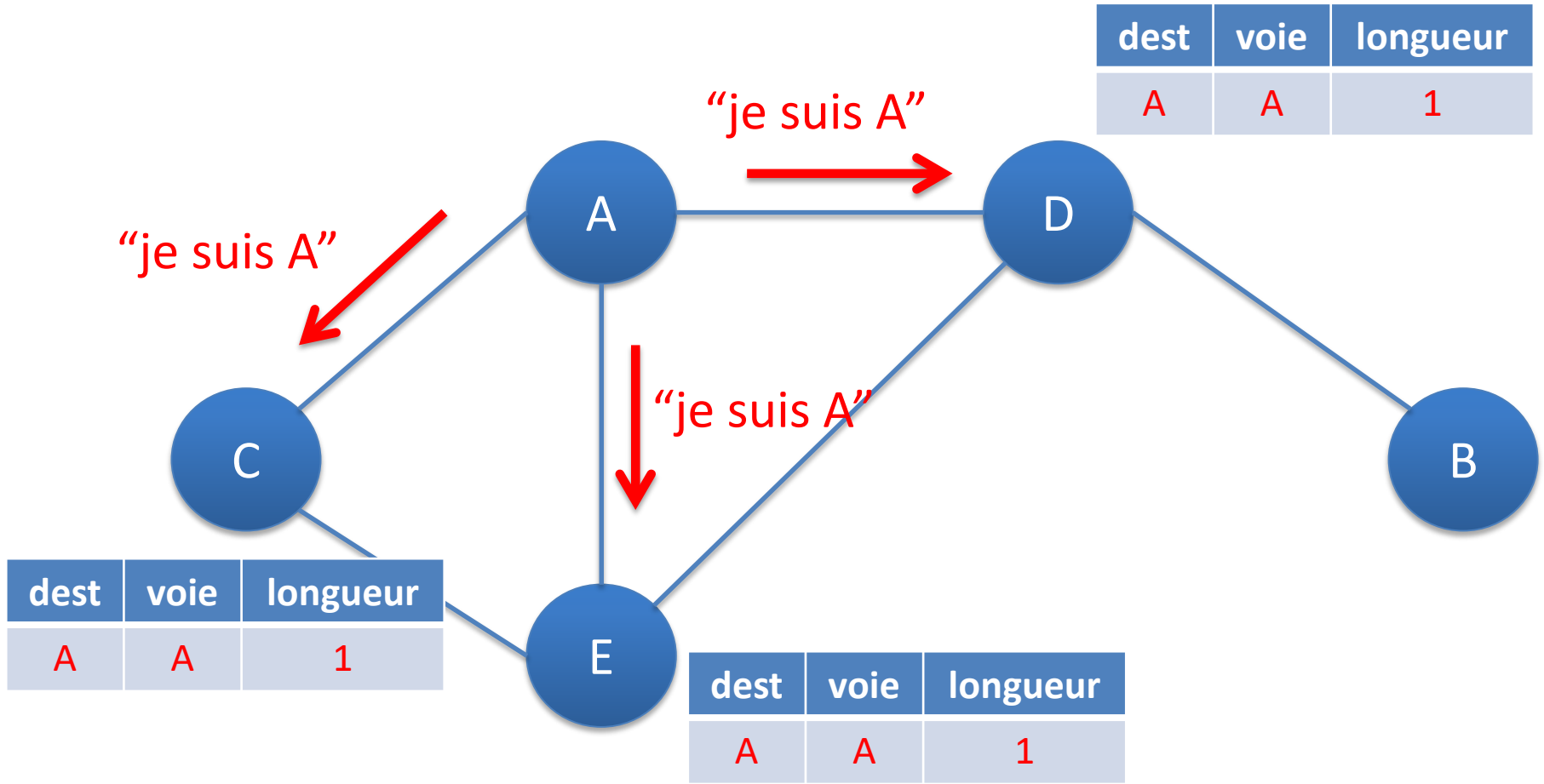
►Chaque noeud annonce à ses voisins la “longueur” des chemins qu’il connaît

►Chaque noeud retient et propage le chemin le plus court parmi ceux annoncés par ses voisins

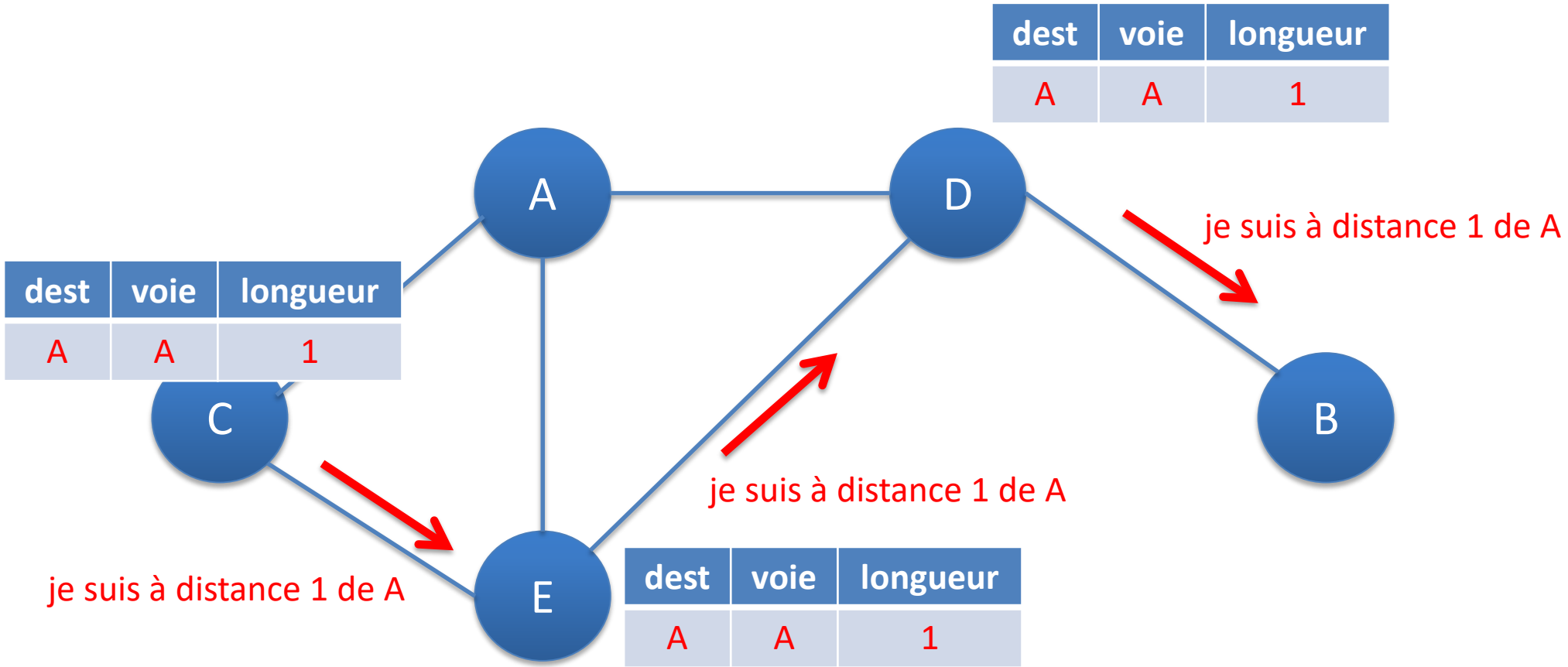
Par exemple



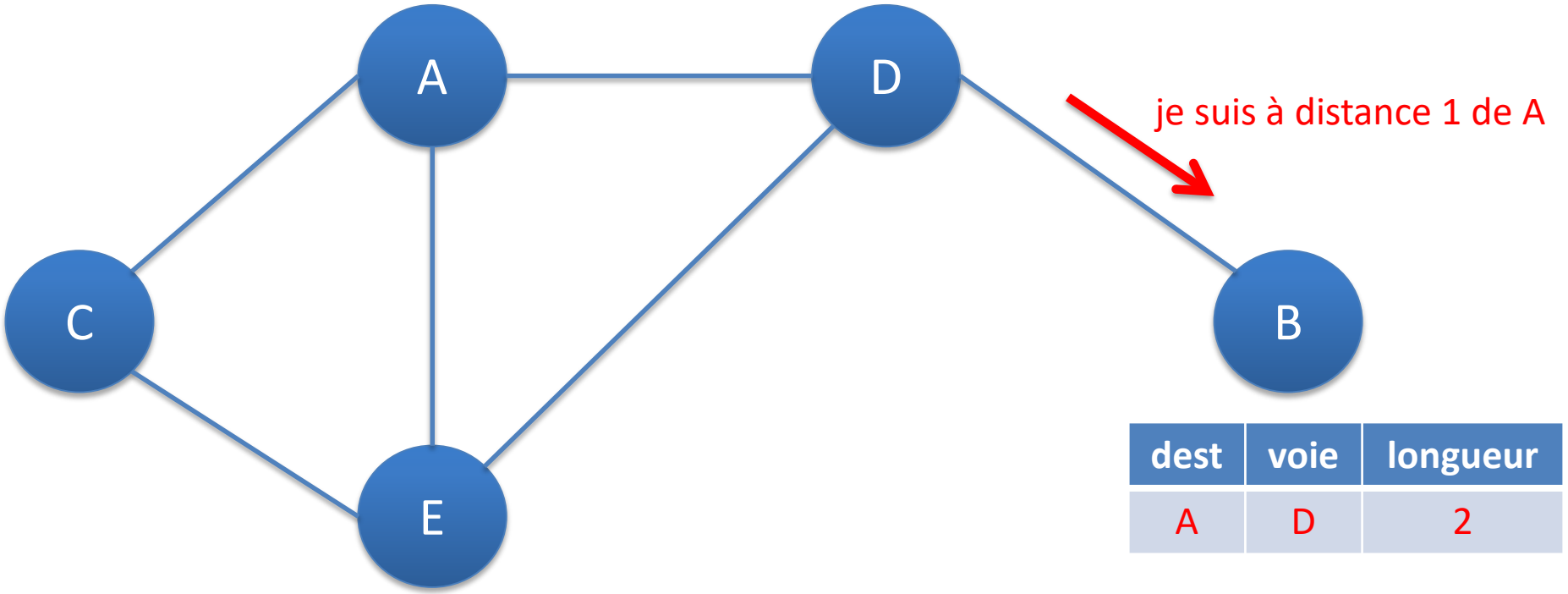
Par exemple


"je suis A"
"je suis A"
"je suis A"
A
D
C
B
E
dest voie longueur
A A 1
dest voie longueur
A A 1
dest voie longueur
A A 1

Par exemple


dest voie longueur
A A 1
A
D
je suis à distance 1 de A
dest voie longueur
A A 1
C
B
je suis à distance 1 de A
je suis à distance 1 de A
E
dest voie longueur
A A 1

Par exemple


dest
voie
longueur
A
A
1
A
D
je suis à distance 1 de A
C
B
E
dest
voie
longueur
A
D
2

Résumé

- La notion de protocole de communication
  - Jeu de règles pour la transmission des données au sein d'un seul ordinateur ou entre plusieurs ordinateurs
- La commutation par paquets
  - Grouper les données dans des paquets de taille fixe
  - Plusieurs types des paquets sont nécessaires pour l'implementation des protocoles
  - Entêtes des paquets
- Les couches des protocoles
- Exemple: le protocole de l'Internet (TCP/IP)