

# Formation SI

Rémi Géraud, Dominique Martinet, Valentin Roussellet



Décembre 2008

Modèle OSI et réseaux

Empaquetage des données

Transport des paquets

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

Ordonnancement des tâches

Autres problèmes d'optimisations

Bases de données et SQL

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

Des Questions

## Modèle OSI et réseaux

Empaquetage des données

Transport des paquets

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

## Ordonnancement des tâches

## Autres problèmes d'optimisations

## Bases de données et SQL

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

## Des Questions

### Modèle OSI et réseaux

Empaquetage des données

Transport des paquets

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

### Ordonnancement des tâches

### Autres problèmes d'optimisations

### Bases de données et SQL

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

### Des Questions

# Modèle OSI et réseaux

## Modèle OSI et réseaux

Empaquetage des données

Transport des paquets

1. Couche physique

2. Couche liaison

3. Couche réseau

4. Couche transport

7. Couche application

Ordonnancement des tâches

Autres problèmes d'optimisations

Bases de données et SQL

1. Base de données

2. Du modèle logique à la BDD

3. le SQL

Des Questions

# Modèle OSI : un modèle en 7 couches

- ▶ 1. Couche physique
- ▶ 2. Couche liaison
- ▶ 3. Couche réseau
- ▶ 4. Couche transport
- ▶ 5. Couche session
- ▶ 6. Couche présentation
- ▶ 7. Couche application

## Modèle OSI et réseaux

Empaquetage des données

Transport des paquets

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

Ordonnancement des tâches

Autres problèmes d'optimisations

Bases de données et SQL

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

Des Questions

# Modèle OSI : un modèle en 7 couches

- ▶ 1. Couche physique
  - ▶ 2. Couche liaison
  - ▶ 3. Couche réseau
  - ▶ 4. Couche transport
  - ▶ 5. Couche session
  - ▶ 6. Couche présentation
  - ▶ 7. Couche application
- ▶ Chaque couche est indépendante de celles au-dessus et en-dessous.

## Modèle OSI et réseaux

### Empaquetage des données

#### Transport des paquets

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

## Ordonnancement des tâches

## Autres problèmes d'optimisations

## Bases de données et SQL

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

## Des Questions

# Entêtes de paquets (*headers*)

- ▶ Les données (suite d'octets) sont regroupées en paquets avant transmission

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

# Entêtes de paquets (*headers*)

- ▶ Les données (suite d'octets) sont regroupées en paquets avant transmission
- ▶ Chaque couche peut ajouter des entêtes (parfois des enqueues) pour renseigner le paquet

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# Entêtes de paquets (*headers*)

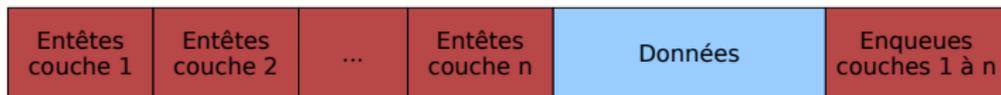
- ▶ Les données (suite d'octets) sont regroupées en paquets avant transmission
- ▶ Chaque couche peut ajouter des entêtes (parfois des enqueues) pour renseigner le paquet
- ▶ Paquet = Entêtes 1 + Entêtes 2 + ... + Données

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

# Entêtes de paquets (*headers*)

- ▶ Les données (suite d'octets) sont regroupées en paquets avant transmission
- ▶ Chaque couche peut ajouter des entêtes (parfois des enqueues) pour renseigner le paquet
- ▶ Paquet = Entêtes 1 + Entêtes 2 + ... + Données
- ▶ Exemple :



1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# Transport des paquets

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

Le problème :

- ▶ Comment dire à un paquet d'aller à telle destination ?
- ▶ Comment être sûr que la transmission a réussi (si besoin) ?
- ▶ Comment partager une connexion entre plusieurs applications ?
- ▶ C'est le rôle des différentes couches et de leurs protocoles associés.

# Modèle OSI : 1. Couche physique

- ▶ Transforme une série de bits en un signal

# Modèle OSI : 1. Couche physique

- ▶ Transforme une série de bits en un signal
- ▶ Support : Electricité, ondes hertziennes, fibres optiques...
- ▶ Matériel : Emetteurs/récepteurs wifi, hubs, câbles

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

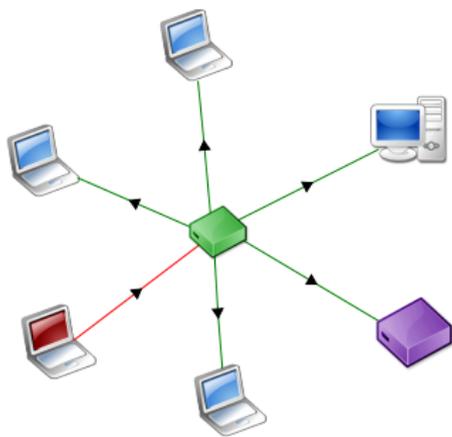
# Modèle OSI : 1. Couche physique

- ▶ Transforme une série de bits en un signal
- ▶ Support : Electricité, ondes hertziennes, fibres optiques...
- ▶ Matériel : Emetteurs/récepteurs wifi, hubs, câbles



# Fonctionnement d'un hub (concentrateur)

Un hub envoie l'information reçue sur un port sur tous les autres ports branchés :



Modèle OSI et réseaux

Empaquetage des données

Transport des paquets

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

Ordonnancement des tâches

Autres problèmes d'optimisations

Bases de données et SQL

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

Des Questions

## Modèle OSI : 2. Couche liaison

- ▶ Assure la communication entre deux machines *voisines*.
- ▶ Découpage des informations en *trames* : correction des erreurs, gestion des collisions

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# Modèle OSI : 2. Couche liaison

- ▶ Assure la communication entre deux machines *voisines*.
- ▶ Découpage des informations en *trames* : correction des erreurs, gestion des collisions
- ▶ Support : Sous-couches MAC et LLC, protocoles Ethernet, Token Ring...
- ▶ Matériel : Switches (commutateurs réseaux)



1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# Media Access Control (MAC)

Gère l'accès direct au média physique

Adresses MAC :

- ▶ Forme : XX:XX:XX:XX:XX:XX (où XX sont des entiers hexadécimaux entre 00 et FF)
- ▶ Adresse "physique" quasi unique



1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

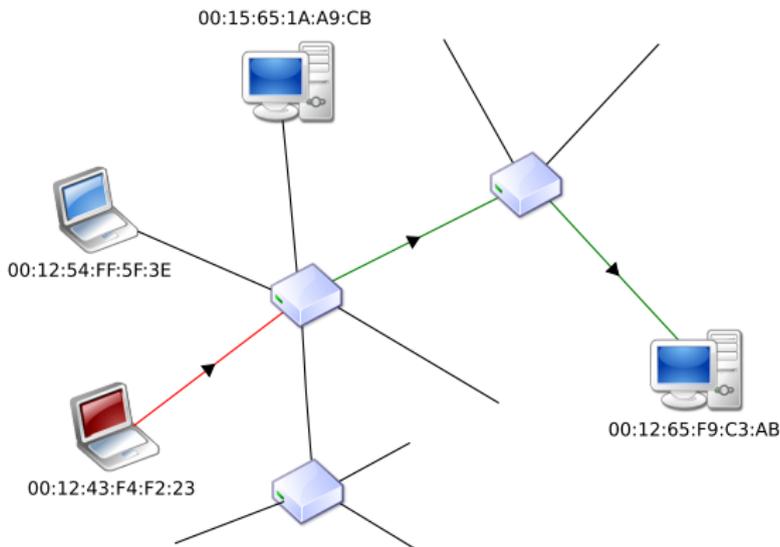
1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# Fonctionnement d'un switch

Un switch lit l'adresse MAC destination d'un paquet pour déterminer sur quel port le retransmettre :

# Fonctionnement d'un switch

Un switch lit l'adresse MAC destination d'un paquet pour déterminer sur quel port le retransmettre :



1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

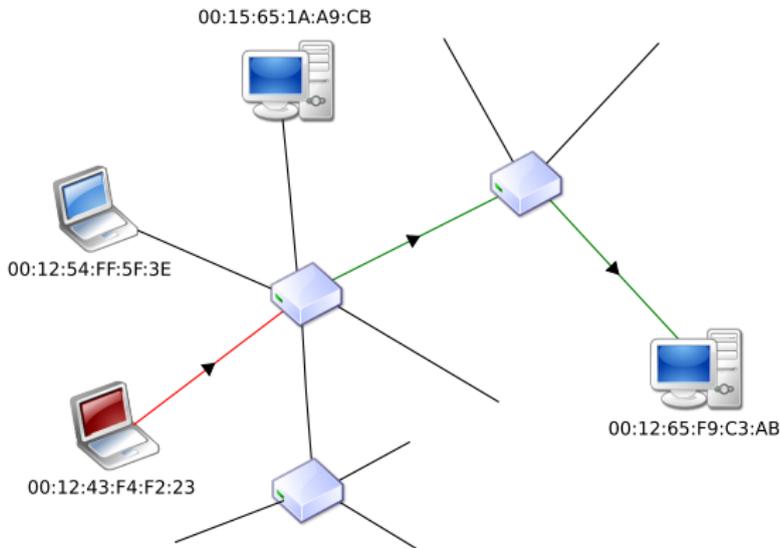
1. Base de données
2. Du modèle logique à la BDD
3. le SQL

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# Fonctionnement d'un switch

Un switch lit l'adresse MAC destination d'un paquet pour déterminer sur quel port le retransmettre :



Un réseau switché est limité à peu de machines. On parle de LAN (Local Area Network).

# Modèle OSI : 3. Couche réseau

- ▶ Assure la communication entre deux nœuds *distants*

# Modèle OSI : 3. Couche réseau

- ▶ Assure la communication entre deux nœuds *distants*
- ▶ Support : Protocoles IP, ARP, ICMP...
- ▶ Matériel : Routeurs



1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# Classes d'IP

Une adresse IP est de la forme x.x.x.x où les x sont des entiers entre 0 et 255.

IP est un protocole géographique (hiérarchie).

Classes d'IP :

- ▶ A (0.0.0.0 – 126.0.0.0) : 126 réseaux, 16777214 hôtes / réseau
- ▶ B (128.0.0.0 – 191.255.0.0) : 16384 réseaux, 65534 hôtes / réseau
- ▶ C (192.0.0.0 – 223.255.255.0) : 2097152 réseaux, 254 hôtes / réseau
- ▶ 224.0.0.0 et au-delà : IP réservées
- ▶ 127.0.0.0 : IPs spéciales
- ▶ 10/8, 172.16/12, 192.168/16 : IPs privées

Notation : 138.195.0.0/16 ou 138.195.0.0 netmask  
255.255.0.0

# CIDR : Classless Inter-Domain Routing

- ▶ Élargir la notion de classe d'IP et s'en affranchir

# CIDR : Classless Inter-Domain Routing

- ▶ Élargir la notion de classe d'IP et s'en affranchir
- ▶ Exemple : 138.195.153.1/25
- ▶ “/25” = “netmask 255.255.255.128”

# CIDR : Classless Inter-Domain Routing

- ▶ Élargir la notion de classe d'IP et s'en affranchir
- ▶ Exemple : 138.195.153.1/25
- ▶ “/25” = “netmask 255.255.255.128”
- ▶ VIA : 138.195.128.0/19

# CIDR : Classless Inter-Domain Routing

- ▶ Élargir la notion de classe d'IP et s'en affranchir
- ▶ Exemple : 138.195.153.1/25
- ▶ “/25” = “netmask 255.255.255.128”
- ▶ VIA : 138.195.128.0/19

Les rôles d'un routeur :

- ▶ Connecter deux réseaux (LAN) ensemble
- ▶ Déterminer la route à suivre pour un paquet selon l'IP destination
- ▶ Scinder un paquet en fragments s'il est trop gros pour le réseau suivant
- ▶ Enregistre et met à jour une table de routage

Modèle OSI et réseaux

Empaquetage des données

Transport des paquets

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

Ordonnancement des tâches

Autres problèmes d'optimisations

Bases de données et SQL

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

Des Questions

# ROUTAGE d'un paquet

## Modèle OSI et réseaux

Empaquetage des données

Transport des paquets

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

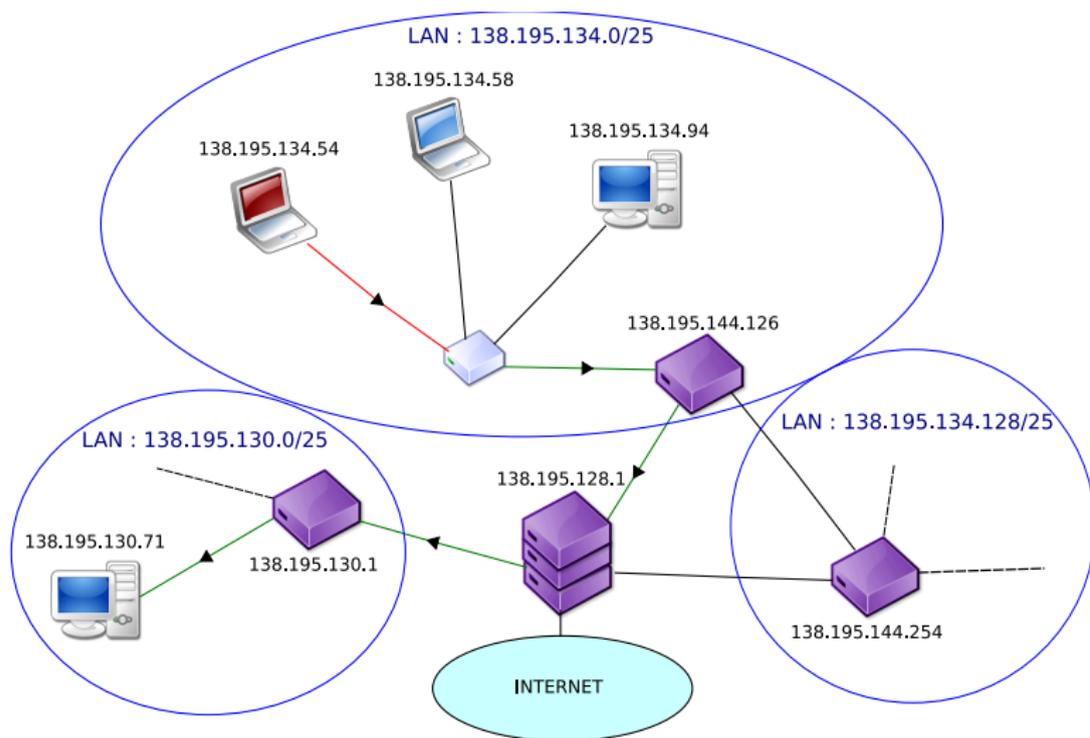
## Ordonnancement des tâches

## Autres problèmes d'optimisations

## Bases de données et SQL

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

## Des Questions



# Découverte automatique de routes IP

Un routeur peut mettre à jour automatiquement ses routes IP :

1. Couche physique
2. Couche liaison
- 3. Couche réseau**
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

# Découverte automatique de routes IP

Un routeur peut mettre à jour automatiquement ses routes IP :

- ▶ OSPF = Open Shortest Path First
- ▶ BGP = Border Gateway Protocol
- ▶ Échanges de routes entre les routeurs
- ▶ Choix préférentiel de la route la plus courte
- ▶ OSPF : structures autonomes
- ▶ BGP : échanges entre structures autonomes

⇒ Réseaux dynamiques et résistants aux pannes

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# Modèle OSI : 4. Couche transport

- ▶ Permet à deux applications de dialoguer
- ▶ Permet, notamment grâce aux *ports*, de partager un lien physique unique entre plusieurs applications

# Modèle OSI : 4. Couche transport

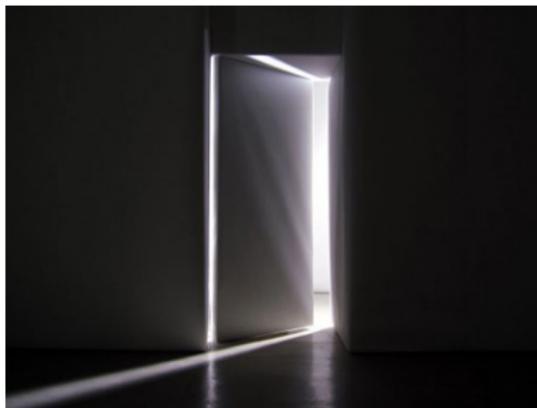
- ▶ Permet à deux applications de dialoguer
- ▶ Permet, notamment grâce aux *ports*, de partager un lien physique unique entre plusieurs applications
- ▶ Support : Protocoles TCP, UDP

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

# Modèle OSI : 4. Couche transport

- ▶ Permet à deux applications de dialoguer
- ▶ Permet, notamment grâce aux *ports*, de partager un lien physique unique entre plusieurs applications
- ▶ Support : Protocoles TCP, UDP



# TCP & UDP (niveau 4)

Caractéristiques du TCP (Transmission Control Protocol) :

- ▶ Envoi avec accusé de réception (SYN/ACK)
- ▶ Fiable car vérification des erreurs
- ▶ Utilisation de ports source et destination
- ▶ Adapté aux communications en mode *connecté*

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

# TCP & UDP (niveau 4)

Caractéristiques du TCP (Transmission Control Protocol) :

- ▶ Envoi avec accusé de réception (SYN/ACK)
- ▶ Fiable car vérification des erreurs
- ▶ Utilisation de ports source et destination
- ▶ Adapté aux communications en mode *connecté*

Caractéristiques de l'UDP (User Datagram Protocol) :

- ▶ Pas d'accusé de réception, envoi simple
- ▶ Pas de vérification des erreurs (non fiable)
- ▶ Utilisation de ports source et destination
- ▶ Adapté aux flux en mode *non connecté*.
- ▶ Entêtes très simples (port src, port dest, longueur, checksum)

# TCP & UDP (niveau 4)

Caractéristiques du TCP (Transmission Control Protocol) :

- ▶ Envoi avec accusé de réception (SYN/ACK)
- ▶ Fiable car vérification des erreurs
- ▶ Utilisation de ports source et destination
- ▶ Adapté aux communications en mode *connecté*

Caractéristiques de l'UDP (User Datagram Protocol) :

- ▶ Pas d'accusé de réception, envoi simple
- ▶ Pas de vérification des erreurs (non fiable)
- ▶ Utilisation de ports source et destination
- ▶ Adapté aux flux en mode *non connecté*.
- ▶ Entêtes très simples (port src, port dest, longueur, checksum)
- ▶ Exemple : TV sur VideoLAN, Webcam...

# Le principe des ports

Ils mettent en relation deux applications distantes (ex. client et serveur Web)

Ils permettent de partager une connexion entre plusieurs applications

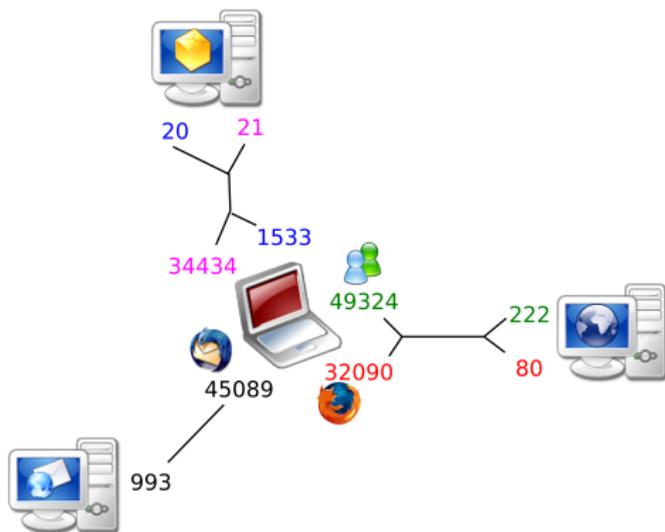
1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

# Le principe des ports

Ils mettent en relation deux applications distantes (ex. client et serveur Web)

Ils permettent de partager une connexion entre plusieurs applications



# Modèle OSI : 7. Couche application

- ▶ Donne du sens aux informations

# Modèle OSI : 7. Couche application

- ▶ Donne du sens aux informations
- ▶ Support : Protocoles HTTP (Web), SMTP (Envoi de mail), POP3 (Réception de mail), FTP (Fichiers), SSH, SNMP, IRC...

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

# Modèle OSI : 7. Couche application

- ▶ Donne du sens aux informations
- ▶ Support : Protocoles HTTP (Web), SMTP (Envoi de mail), POP3 (Réception de mail), FTP (Fichiers), SSH, SNMP, IRC...
- ▶ Spécifique à chaque application

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

# Ordonnancement des tâches

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

# First in, First out

	$\tau_i$	$t_i$
$T_1$	30	0
$T_2$	5	e
$T_3$	2	2e

On exécute les tâches dans l'ordre où elles arrivent

$$T_{moyen} = \frac{30 + (35 - e) + (37 - 2e)}{3} = 34 - e$$

$$T_{total} = 30 + 5 + 2 = 37$$

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# Plus court temps d'exécution

	$\tau_i$	$t_i$
$T_1$	30	0
$T_2$	5	e
$T_3$	2	2e

On attend un peu pour pouvoir lancer la tâche la plus rapide en première

$$T_{moyen} = \frac{2 + (7 + e) + (37 + 2e)}{3} = 15,3 + e$$

$$T_{total} = 2e + 2 + 5 + 30 = 37 + 2e$$

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# Round robin

On coupe le temps en quantum :

q=4	t=0	t=4	t=8	t=12	t=16	t=20	t=40	t=44
$T_1$	30	26	26	26	22	22	2	0
$T_2$	5	5	1	1	1	0		
$T_1$	2	2	2	0				

$$T_{moyen} = \frac{44 + 20 + 12}{3} = 25.3$$

Avec un quantum plus court (q=1) :

$$T_{moyen} = \frac{37 + 12 + 6}{3} = 18.3$$

Modèle OSI et réseaux

Empaquetage des données

Transport des paquets

1. Couche physique

2. Couche liaison

3. Couche réseau

4. Couche transport

7. Couche application

Ordonnancement des tâches

Autres problèmes d'optimisations

Bases de données et SQL

1. Base de données

2. Du modèle logique à la BDD

3. le SQL

Des Questions

# Plus court temps d'exécution restant

On exécute le plus court à la fin de chaque quantum

q=4	t=0	t=4	t=8	t=12	t=16	t=20	t=40	t=44
$T_1$	30	30	30	30	26	22	2	0
$T_2$	5	5	1	0				
$T_1$	2	0						

$$T_{moyen} = \frac{44 + 12 + 4}{3} = 20$$

Avec un quantum plus court (q=1) :

$$T_{moyen} = \frac{37 + 7 + 2}{3} = 15.3$$

Modèle OSI et réseaux

Empaquetage des données

Transport des paquets

1. Couche physique

2. Couche liaison

3. Couche réseau

4. Couche transport

7. Couche application

Ordonnancement des tâches

Autres problèmes d'optimisations

Bases de données et SQL

1. Base de données

2. Du modèle logique à la BDD

3. le SQL

Des Questions

# Conclusion

Quelques problèmes :

- ▶ On ne connaît pas le temps d'exécution...
- ▶ La commutation de contexte n'est pas gratuite.

En pratique, on utilise un round-robin qui va favoriser les programmes utilisant des périphériques interagissant avec l'utilisateur (clavier, souris, écran), et la priorité augmente avec le temps d'attente.

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

# Autres problèmes d'optimisation

# Autres problemes d'optimisations

- ▶ Acces au disque :
  - FIFO
  - Plus court temps de recherche
  - Balayage
  - Plus court temps de latence (comme recherche, mais utilise sens de rotation)
- ▶ Memoire :
  - First fit
  - Best fit
  - Worst fit

# Autres problemes d'optimisations

- ▶ **Pagination**
  - FIFO
  - Least recently used
  - Second chance
- ▶ **Couverture réseau**
  - Nombre d'utilisateurs
  - Surface

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

# Bases de données et SQL

Formation SI

Sha- Asmadeus-  
Louen

Modèle OSI et  
réseaux

Empaquetage des  
données

Transport des paquets

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

Ordonnancement  
des tâches

Autres problèmes  
d'optimisations

**Bases de données  
et SQL**

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

Des Questions

- ▶ Comment les modéliser (cardinalité... ) et les représenter ?
- ▶ Comment passer du modèle à une véritable base de donnée (fichiers) ?

- ▶ Comment les modéliser (cardinalité...) et les représenter ?
- ▶ Comment passer du modèle à une véritable base de donnée (fichiers) ?
- ▶ Structured Query Language : un langage permettant d'interroger une base de données.
- ▶ Utilisable par d'autres langages (Python, PHP...), très répandu (MySQL...)
- ▶ Il n'y a qu'une instruction à connaître pour cette année : SELECT element FROM table.
- ▶ Que l'on peut compléter avec différentes commandes : WHERE, AND, OR... INNER JOIN...

# Structure générale

- ▶ Une base de données est faite de une ou plusieurs **tables** (ex : Chocolat)
- ▶ Chaque table comporte des **champs** (ou colonnes) (ex : Nom, Prix, Provenance...)
- ▶ Les **enregistrements** (ou entrées) d'une table correspondent aux données (ex : Côte d'Or, 5 , 1 Chili...)

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# Modèle logique des données

- ▶ Entités (ex : livre) ayant des propriétés (ex : LivreID, Titre, Prix...)
- ▶ Relations (ex : écrit par) ayant éventuellement des propriétés
- ▶ Cardinalité : (0,n), (1,n), (1,1) (rarement (0,1))

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# Modèle logique

- ▶ Elève (1,1) reçoit (1,n) Note en TT
  - ▶ Eleve (Eleveld, Nom, Noteld)
  - ▶ Note\_en\_TT (Noteld, remarques)

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

- ▶ Elève (1,1) reçoit (1,n) Note en TT
  - ▶ Eleve (Eleveld, Nom, Noteld)
  - ▶ Note\_en\_TT (Noteld, remarques)
  - ▶ Chaque Eleve pointe vers une Note en TT
- ▶ Eleve (1,n) proje (1,n) Matiere
  - ▶ Eleve (Eleveld, Nom)
  - ▶ EleveMatiere (Eleveld, MatiereId, Note)
  - ▶ Matiere (MatiereId, Nom)

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

- ▶ Elève (1,1) reçoit (1,n) Note en TT
  - ▶ Eleve (Eleveld, Nom, Noteld)
  - ▶ Note\_en\_TT (Noteld, remarques)
  - ▶ Chaque Eleve pointe vers une Note en TT
- ▶ Eleve (1,n) proje (1,n) Matiere
  - ▶ Eleve (Eleveld, Nom)
  - ▶ EleveMatiere (Eleveld, MatiereId, Note)
  - ▶ Matiere (MatiereId, Nom)
  - ▶ On crée une table intermédiaire EleveMatiere.
  - ▶ Chaque EleveMatiere pointe vers un élève et une matière, et correspond à une note.

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# SELECT

- ▶ L'instruction SELECT permet de sélectionner certains éléments d'une base de donnée, qui vérifient certaines conditions.
- ▶ Elle ne s'utilise jamais seule : nous allons la compléter progressivement.
- ▶ SELECT element FROM table

FROM table indique où chercher les données.

Exemple : SELECT Provenance FROM Chocolat

Chili

Madagascar

Éthiopie

Guatemala

Venezuela

...

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

- Autre exemple : `SELECT * FROM Chocolat`  
`Côte d'Or, 5, Chili, Chocolat noir...`  
`Madanga , 7, Madagascar, Chocolat noir...`  
 ...

Remarques :

- ▶ `SELECT` affiche toute une entrée (Nom, Prix, Provenance...).
- ▶ on peut rechercher dans plusieurs tables à la fois, dans ce cas, il faut préciser la table concernée en faisant précéder ses colonnes par `nom_de_la_table`.

Exemple : `SELECT Chocolat.Provenance,`  
`Banane.Provenance FROM Chocolat, Banane`  
`Chili, Maroc`  
`Madagascar, Inde`  
 ...

Modèle OSI et réseaux

Empaquetage des données

Transport des paquets

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

Ordonnancement des tâches

Autres problèmes d'optimisations

Bases de données et SQL

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

Des Questions

# WHERE

- ▶ WHERE condition permet de ne sélectionner que les éléments vérifiant une condition

Exemple : `SELECT * FROM Chocolat WHERE Provenance = 'Guatemala'`

Ethiquable, 7, Guatemala, Chocolat au quinoa...  
Jacques Vabre, 6, Guatemala, Chocolat noir...  
...

- ▶ On peut compléter une condition par les opérations AND et OR

Exemple : `SELECT Nom FROM Chocolat WHERE Prix = '6' AND Provenance = 'Guatemala'` Jacques Vabre, 6, Guatemala, Chocolat noir...

- ▶ Les conditions peuvent être des égalités (=), des comparaisons (< ou >)...

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# DISTINCT

SELECT DISTINCT retourne une liste exempte de doublons

- ▶ `SELECT Ville FROM Monuments WHERE Pays = 'France'`  
Lyon, Paris, Paris, Paris, Paris, Marseille, Marseille
- ▶ `SELECT DISTINCT Ville FROM Monuments WHERE Pays = 'France'`  
Lyon, Paris, Marseille

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

# ORDER BY et ASC/DESC

ORDER BY permet de trier les résultats

▶ `SELECT * FROM Employes`

Albert, Einstein, Balayeur...

Constantin, Taine, Éclairagiste...

Jean, Taine, Chauffeur...

▶ `SELECT * FROM Employes ORDER BY Nom DESC`

Jean, Taine, Chauffeur...

Constantin, Taine, Éclairagiste...

Albert, Einstein, Balayeur...

# ORDER BY et ASC/DESC

On peut trier les résultats par rapport à plusieurs entrées,

- ▶ `SELECT * FROM Employes ORDER BY Nom, Prenom DESC`

Constantin, Taine, Éclairagiste...

Jean, Taine, Chauffeur...

Albert, Einstein, Balayeur...

- ▶ ASC : ascendant
- ▶ DESC : descendant

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

# INNER JOIN

- ▶ La seule instruction un peu subtile...
- ▶ INNER JOIN réalise une \*intersection\* de deux tables
- ▶ `SELECT Employe.Nom, Poste.Nom FROM Employe  
INNER JOIN Poste ON Employe.PosteID =  
Poste.PosteID`

Taine, Chauffeur

Taine, Éclairagiste

Einstein, Balayeur

...

- ▶ `SELECT Employe.Nom, Poste.Nom FROM Employe,  
Poste WHERE Employe.PosteID = Poste.PosteID`

Fait exactement la même chose que la commande avec  
INNER JOIN !

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

# INNER JOIN

L'intérêt de INNER JOIN apparaît lorsqu'il y a plus de deux tables en jeu, ou que l'on veut effectuer une recherche plus précise. Exemple (contrôle de l'année dernière) :

```
SELECT Recette.NomRecette
FROM
Recette INNER JOIN (IngredientsRecette INNER
JOIN Ingredient
ON IngredientsRecette.IngredientID =
Ingredient.IngredientID)
ON Recette.RecetteID =
IngredientsRecette.IDRecette
WHERE NomIngredient = 'Chocolat'
```

Que fait cette requête ?

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique  
à la BDD
3. le SQL

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

## Résumé et méthode

- ▶ Identifier ce que l'on veut récupérer : `SELECT element`
- ▶ Identifier là où on veut le récupérer : `FROM table, FROM table1 INNER JOIN table2 ON condition`
- ▶ Éventuellement :
  - ▶ identifier les conditions que doit vérifier l'élément recherché : `WHERE condition`
  - ▶ ordonner les résultats : `ORDER BY element2`
  - ▶ éviter les doublons : `DISTINCT`

Remarque : on peut sauter des lignes pour améliorer la lisibilité :

```
SELECT Chocolat.Nom
FROM
Chocolat INNER JOIN Cacao ON Chocolat.CacaoID =
Cacao.CacaoID
WHERE
Chocolat.Prix < 10
ORDER BY Chocolat.Prix ASC
Qu'affiche cette commande ?
```

# Des questions ?

- ▶ sha@via.ecp.fr
- ▶ asmadeus@via.ecp.fr
- ▶ louen@via.ecp.fr

## Modèle OSI et réseaux

### Empaquetage des données

#### Transport des paquets

1. Couche physique
2. Couche liaison
3. Couche réseau
4. Couche transport
7. Couche application

## Ordonnancement des tâches

## Autres problèmes d'optimisations

## Bases de données et SQL

1. Base de données
2. Du modèle logique à la BDD
3. le SQL

## Des Questions