

# Réseaux : modèles, protocoles, programmation

Pablo Rauzy

pr@up8.edu

pablo.rauzy.name/teaching/rmpp



UFR MITSIC / L2 informatique

## Séance 1

Open Systems Interconnection

La couche physique

La couche liaison de données

# Open Systems Interconnection

---

- ▶ OSI signifie *Open Systems Interconnection*.
- ▶ C'est un modèle standard de communication en réseau pour tous les systèmes informatiques.

- ▶ Conçu dans les années 1970.
- ▶ À l'époque, rivalité entre 3 architectures différentes :
  - DSA (*Distributed System Architecture*), développé en France (pour Cyclades) par CII,
  - Decnet, par DEC,
  - SNA (*Systems Network Architecture*), par IBM.
- ▶ C'est DSA qui inspire le modèle OSI et ses sept couches.
- ▶ En 1978, C. Bachman présente le modèle en sept couches.
- ▶ La standardisation intervient tard (en 1984) et entre temps le modèle TCP/IP voit le jour.
- ▶ TCP/IP est adopté par l'ARPANET en 1983.

- La couche *physique* est chargée de la transmission effective des signaux entre les interlocuteurs. Son service est limité à l'émission et la réception de bits.

- ▶ La couche *physique* est chargée de la transmission effective des signaux entre les interlocuteurs. Son service est limité à l'émission et la réception de bits.
- ▶ La couche *liaison de données* gère les communications entre 2 machines directement connectées entre elles.

- ▶ La couche *physique* est chargée de la transmission effective des signaux entre les interlocuteurs. Son service est limité à l'émission et la réception de bits.
- ▶ La couche *liaison de données* gère les communications entre 2 machines directement connectées entre elles.
- ▶ La couche *réseau* gère les communications de proche en proche, généralement entre machines : routage et adressage des paquets.

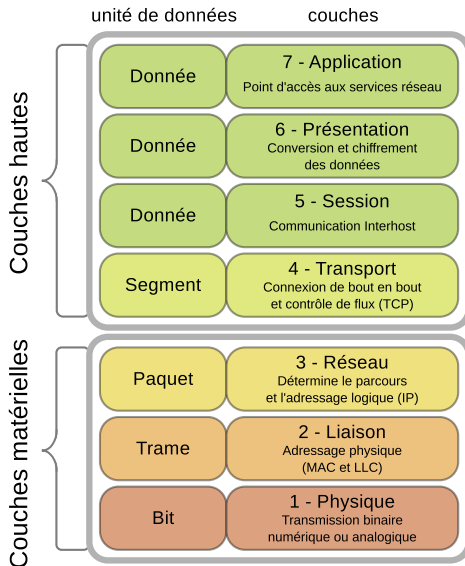
- ▶ La couche *physique* est chargée de la transmission effective des signaux entre les interlocuteurs. Son service est limité à l'émission et la réception de bits.
- ▶ La couche *liaison de données* gère les communications entre 2 machines directement connectées entre elles.
- ▶ La couche *réseau* gère les communications de proche en proche, généralement entre machines : routage et adressage des paquets.
- ▶ La couche *transport* gère les communications de bout en bout entre programmes.



- ▶ La couche *physique* est chargée de la transmission effective des signaux entre les interlocuteurs. Son service est limité à l'émission et la réception de bits.
- ▶ La couche *liaison de données* gère les communications entre 2 machines directement connectées entre elles.
- ▶ La couche *réseau* gère les communications de proche en proche, généralement entre machines : routage et adressage des paquets.
- ▶ La couche *transport* gère les communications de bout en bout entre programmes.
- ▶ La couche *session* gère la synchronisation des échanges et les transactions.

- ▶ La couche *physique* est chargée de la transmission effective des signaux entre les interlocuteurs. Son service est limité à l'émission et la réception de bits.
- ▶ La couche *liaison de données* gère les communications entre 2 machines directement connectées entre elles.
- ▶ La couche *réseau* gère les communications de proche en proche, généralement entre machines : routage et adressage des paquets.
- ▶ La couche *transport* gère les communications de bout en bout entre programmes.
- ▶ La couche *session* gère la synchronisation des échanges et les transactions.
- ▶ La couche *présentation* est chargée de la conversion entre données manipulées au niveau applicatif et chaînes d'octets effectivement transmises.

- ▶ La couche *physique* est chargée de la transmission effective des signaux entre les interlocuteurs. Son service est limité à l'émission et la réception de bits.
- ▶ La couche *liaison de données* gère les communications entre 2 machines directement connectées entre elles.
- ▶ La couche *réseau* gère les communications de proche en proche, généralement entre machines : routage et adressage des paquets.
- ▶ La couche *transport* gère les communications de bout en bout entre programmes.
- ▶ La couche *session* gère la synchronisation des échanges et les transactions.
- ▶ La couche *présentation* est chargée de la conversion entre données manipulées au niveau applicatif et chaînes d'octets effectivement transmises.
- ▶ La couche *application* est le point d'accès au service réseau.



# La couche physique

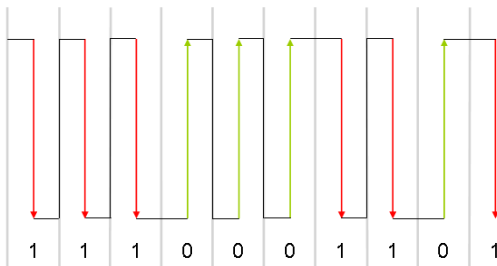
---

- ▶ Première couche du modèle OSI.
- ▶ Chargée de la transmission des signaux électriques ou optiques (ou autres !) entre les interlocuteurs.
- ▶ En charge de la conversion entre bits logiques et signaux physiques.
- ▶ Unité de données (PDU) : bit.
- ▶ En pratique, réalisé par un circuit électronique spécifique (inclut dans les cartes réseaux).

- ▶ Pour transmettre les bits, on les traduit en signaux électriques avec un *codage en ligne*.
- ▶ Quelques exemples de tels codages sont :
  - les codages à deux niveaux (e.g., Manchester),
  - les codages à trois niveaux (e.g., AMI).

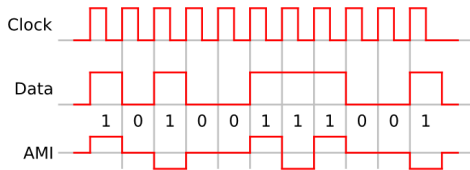
## Exemple : le codage Manchester

- ▶ Fonctionne avec une valeur haute et une valeur basse du signal.
- ▶ Encodage des bits par transition entre ces valeurs :
  - 0 logique : front montant.
  - 1 logique : front descendant.





- ▶ Fonctionne avec trois niveaux de signal : positif, neutre, négatif.
- ▶ Encodage des bits :
  - 0 logique : absence de symbole (signal neutre).
  - 1 logique : impulsion de polarité alternative.



- ▶ Dans le cas d'Ethernet, on parle de *PHYceiver*.
- ▶ C'est un dispositif électronique capable d'envoyer et recevoir des *trames* Ethernet.
- ▶ Utilisé directement dans un *hub* (concentrateur) ou les *switchs* (commutateur).
- ▶ Dans une carte réseau ce dispositif est complété par (entres autres) une interface MAC, et parfois par des circuits avancés (Wake-on-LAN, etc.).

# La couche liaison de données

---

- ▶ Seconde couche du modèle OSI.
- ▶ Chargée du transfert des données entre machines d'un même réseau local ou étendu.
- ▶ Elle fournit aux couches supérieures les moyens procéduraux pour effectuer sa tâche.
- ▶ Parfois en charge de la détection d'erreurs au niveau de la couche physique.
- ▶ Unité de données (PDU) : trame.

- ▶ Le transfert de trames par des systèmes de couche 2 utilise des adresses non-ambiguës de matériel.
- ▶ Les trames contiennent les adresses source et destination dans leurs entêtes.
- ▶ Ces adresses sont “plates” (pas de routage possible).

- ▶ Il peut y avoir des collisions si deux machines différentes tentent de communiquer simultanément sur le même support.
- ▶ Les protocoles de liaisons sont chargés de détecter et de se remettre de ces collisions, mais pas de les empêcher.
- ▶ Certains protocoles à ce niveau là font de la détections d'erreurs (avec des sommes de contrôle par exemple), mais pas tous.

- ▶ La couche liaison de données est en fait subdivisée en deux sous-couches :
  - la sous-couche haute LLC,
  - la sous-couche basse MAC.

- ▶ LLC signifie *Logical Link Control* (contrôle de la liaison logique).
- ▶ Cette sous-couche est en charge du multiplexage des protocoles des couches supérieures.
- ▶ Optionnellement, elle fournit le contrôle (voire la correction) d'erreurs et l'accusé de réception.



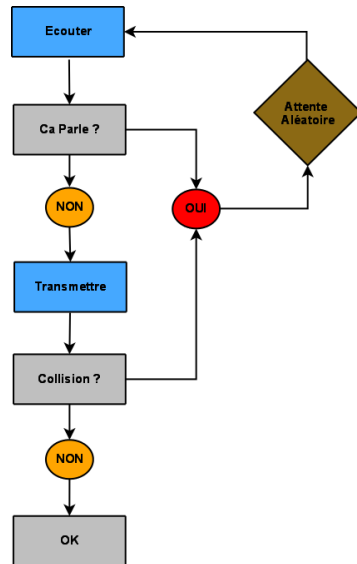
- ▶ MAC signifie *Media Access Control* (contrôle d'accès au support).
- ▶ Cette sous-couche est en charge de réguler les émissions sur un support donné, typiquement quand plusieurs systèmes indépendants sont susceptibles d'émettre en même temps sur le même support.
- ▶ C'est aussi à ce niveau là que sont définis les formats de trames et les méthodes d'adressages des systèmes (d'où les adresses MAC, normalement univoque).
- ▶ Il y a deux formes de contrôle d'accès :
  - centralisée, où un des systèmes joue le rôle de chef d'orchestre ;
  - distribuée, où il faut attendre que plus personne ne parle pour prendre la parole, et être poli en cas de conflit.

- ▶ MAC signifie *Media Access Control* (contrôle d'accès au support).
- ▶ Cette sous-couche est en charge de réguler les émissions sur un support donné, typiquement quand plusieurs systèmes indépendants sont susceptibles d'émettre en même temps sur le même support.
- ▶ C'est aussi à ce niveau là que sont définis les formats de trames et les méthodes d'adressages des systèmes (d'où les adresses MAC, normalement univoque).
- ▶ Il y a deux formes de contrôle d'accès :
  - centralisée, où un des systèmes joue le rôle de chef d'orchestre ;
  - distribuée, où il faut attendre que plus personne ne parle pour prendre la parole, et être poli en cas de conflit.
- ▶ C'est donc aussi ce niveau qui est chargé de délimiter les trames, ce qui est possible de 3 façons :
  - approche basée sur le temps,
  - comptage de caractères (taille annoncée dans l'entête),
  - le bourrage en début et fin de trame.

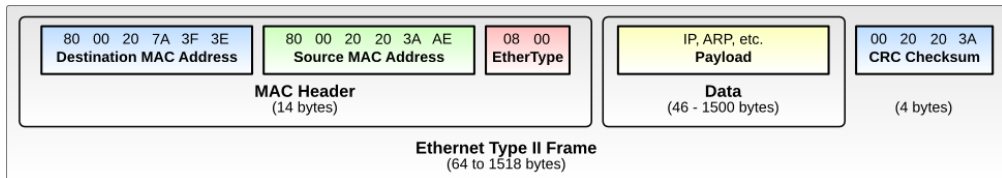
- ▶ Il existe de nombreux protocoles de niveau 2.
- ▶ Nous allons en regarder deux :
  - Ethernet,
  - PPP.

- ▶ Ethernet est utile sur les réseaux locaux.
- ▶ Il fonctionne par *commutation de paquets* (par opposition à la *commutation de circuits*).
- ▶ Il peut s'appuyer une *paire torsadée* (pour éviter la diaphonie) ou même sur de la fibre optique.
- ▶ Le Wi-Fi est une variante sans fil d'Ethernet.
- ▶ Il existe de nombreuses variantes d'Ethernet (10M, 100M, Gigabit, et même 10 gigabits).

- Le principe d'Ethernet est le même que celui des ondes radio : on communique sur un canal commun et chacun possède une clef unique (adresse MAC).
- L'orchestration se fait avec le protocole CSMA/CD (Carrier Sense Multiple Access with Collision Detection) expliqué ci-contre.



- Le format des trames Ethernet est le suivant :



- ▶ PPP signifie *Point-to-Point Protocol* (protocole point à point).
- ▶ Il s'agit d'un protocole de liaison point à point, c'est à dire direct entre deux hôtes uniquement.
- ▶ Il est massivement utilisé, notamment dans le cadre des connexion internet de particulier où il est encapsulé (PPPoX).

- PPP signifie *Point-to-Point Protocol* (protocole point à point).
- Il s'agit d'un protocole de liaison point à point, c'est à dire direct entre deux hôtes uniquement.
- Il est massivement utilisé, notamment dans le cadre des connexion internet de particulier où il est encapsulé (PPPoX).
- Sa trame est la suivante :

| Flag de début         | Adresse               | Commande              | Protocole    | Données | FCS        | Flag de fin           |
|-----------------------|-----------------------|-----------------------|--------------|---------|------------|-----------------------|
| 8 bits<br>(01111110b) | 8 bits<br>(11111111b) | 8 bits<br>(00000011b) | 8 ou 16 bits | ...     | 16/32 bits | 8 bits<br>(01111110b) |