

# Réseaux : modèles, protocoles, programmation

Pablo Rauzy

`pr@up8.edu`

`pablo.rauzy.name/teaching/rmpp`



UFR MITSIC / L2 informatique

Séance 0

Chargement d'une page web

Présentation du cours

# Chargement d'une page web

---

- Que se passe-t-il quand on demande à son navigateur de charger une page web ?

- ▶ Que se passe-t-il quand on demande à son navigateur de charger une page web ?
  - requête DNS (*Domain Name System*),

- ▶ Que se passe-t-il quand on demande à son navigateur de charger une page web ?
  - requête DNS (*Domain Name System*),
  - requête HTTP (*HyperText Transfer Protocol*).

- ▶ Que se passe-t-il quand on demande à son navigateur de charger une page web ?
  - requête DNS (*Domain Name System*),
  - requête HTTP (*HyperText Transfer Protocol*).
- ▶ On va prendre l'exemple de `www.univ-paris8.fr`.

- ▶ Le but de la requête DNS est de trouver l'adresse IP associée au nom de domaine du site web qu'on veut consulter.
- ▶ Cette requête va être adressée à un *serveur de nom*, sous forme d'un *paquet*.
- ▶ Un paquet est constitué d'entêtes et d'un corps.

- ▶ Votre système crée donc un paquet DNS.
- ▶ Les entêtes de ce paquet disent (entre autres) :
  - je suis une requête (et pas une réponse),
  - si le message arrive en une seule fois ou si il est tronqué,
  - j'autorise (ou non) les requêtes récursives.
- ▶ Le corps de ce paquet dit :
  - pour le nom de domaine `www.univ-paris8.fr`,
  - je veux l'adresse IPv4 (**A**),
  - sur internet (**IN**).



- ▶ La requête ne part pas encore de votre système, avant ça elle est *encapsulée* dans un paquet UDP.
- ▶ UDP signifie *User Datagram Protocol*.
- ▶ Le rôle de UDP est celui d'une enveloppe de *transport* des données (ici notre paquet DNS) d'un programme source à un programme destination.

- ▶ La requête ne part pas encore de votre système, avant ça elle est *encapsulée* dans un paquet UDP.
- ▶ UDP signifie *User Datagram Protocol*.
- ▶ Le rôle de UDP est celui d'une enveloppe de *transport* des données (ici notre paquet DNS) d'un programme source à un programme destination.
- ▶ Le système crée donc un paquet UDP.
- ▶ Les entêtes de ce paquet UDP contiennent (entre autres) :
  - le port source (par exemple 43234),
  - le port destination (53, le port assigné aux serveurs de noms).
- ▶ Le corps de ce paquet contient le paquet DNS.

## Encapsulation dans un paquet IP

- ▶ La requête ne part toujours pas de votre système !  
Avant ça, elle est encapsulée dans un paquet IP.
- ▶ IP signifie *Internet Protocol*.
- ▶ Le rôle de IP est de servir de transporteur sur le *réseau*.

- ▶ La requête ne part toujours pas de votre système !  
Avant ça, elle est encapsulée dans un paquet IP.
- ▶ IP signifie *Internet Protocol*.
- ▶ Le rôle de IP est de servir de transporteur sur le *réseau*.
- ▶ Le système crée donc un paquet IP.
- ▶ Les entêtes de ce paquet IP contiennent (entre autres) :
  - la version du protocole utilisé (4 ou 6),
  - le protocole utilisé au niveau supérieur (pour nous 17 pour UDP),
  - l'adresse source (par exemple 192.168.1.51),
  - l'adresse destination (par exemple 192.168.1.254).

## Encapsulation dans un paquet Ethernet

- ▶ La requête ne part toujours pas de votre système !  
Avant ça, elle est encapsulée dans un paquet Ethernet.
- ▶ Dans notre analogie Ethernet est en quelques sortes la sacoche du postier qui récupère le courrier dans la boîte au lettre et l'amène au centre de tri.

- ▶ La requête ne part toujours pas de votre système !  
Avant ça, elle est encapsulée dans un paquet Ethernet.
- ▶ Dans notre analogie Ethernet est en quelques sortes la sacoche du postier qui récupère le courrier dans la boîte au lettre et l'amène au centre de tri.
- ▶ Le système crée donc un paquet Ethernet.
- ▶ Les entêtes de ce paquet Ethernet contiennent (entre autres) :
  - le protocole utilisé au niveau supérieur (pour nous `0x0800` pour IPv4),
  - l'adresse mac source (par exemple `3c:a9:f4:78:78:78`),
  - l'adresse mac destination (par exemple `e8:f1:b0:78:78:78`).

- ▶ Arrive enfin le déplacement “physique” des données.
- ▶ Dans mon cas, elles sont transmises suivant le protocole 802.11n (WiFi) jusqu’à ma box ADSL.
- ▶ Dans notre analogie, il s’agit de la mobylette du postier.

- ▶ La réponse à ma requête DNS est transmise par ma box ADSL à mon ordinateur de la même manière.
- ▶ Si la réponse n'était pas en cache au niveau de la box, celle-ci fait alors récursivement la requête à un autre serveur de noms (celui de mon fournisseur d'accès) :
  - Cette fois-ci, le déplacement physique s'est fait en ADSL (au moins au début),
  - la liaison avec le protocole ATM (par exemple, sur la partie ADSL).



- ▶ Une fois la réponse DNS reçue, on connaît l'adresse IP de la machine qui héberge `www.univ-paris8.fr` : `193.54.174.19`.
- ▶ On peut donc lui faire notre requête HTTP.

- ▶ Le but de la requête HTTP est de récupérer le contenu de la page web qu'on veut visiter.
- ▶ Cette requête va être adressée à un *serveur web*.

- ▶ Votre système crée la requête HTTP.
- ▶ HTTP est un protocole haut niveau (application).
- ▶ La requête commence par un verbe (**GET**), suivie du chemin de la ressource demandée (/), suivie de la version du protocole (**HTTP/1.1**).
- ▶ Viennent ensuite les entêtes, dont :
  - le serveur (**www.univ-paris8.fr**),
  - l'agent utilisateur (**Mozilla (...)**),
  - potentiellement des cookies, etc.
- ▶ Vient ensuite le corps de la requête (mais celui-ci est vide pour une requête **GET**).

## Encapsulation dans des paquets TCP

- ▶ Comme pour la requête DNS, on a besoin d'une enveloppe.
- ▶ Cette fois-ci cependant :
  - la requête peut être trop grande pour rentrer dans une seule enveloppe,
  - on veut être sûr d'avoir transmis correctement toutes les enveloppes,
  - et aussi être capable de les remettre dans l'ordre.

- ▶ Comme pour la requête DNS, on a besoin d'une enveloppe.
- ▶ Cette fois-ci cependant :
  - la requête peut être trop grande pour rentrer dans une seule enveloppe,
  - on veut être sûr d'avoir transmis correctement toutes les enveloppes,
  - et aussi être capable de les remettre dans l'ordre.
- ▶ On utilise donc TCP (*Transmission Control Protocol*) à la place de UDP.
- ▶ TCP permet des transmissions par *sessions* :
  - on commence par établir une connexion,
  - on transfère les données,
  - puis on ferme la connexion.
- ▶ Chaque paquet TCP contient en entête (entre autres) :
  - le port source (par exemple 57743),
  - le port destination (80, le port assigné aux serveurs web).
  - un numéro de séquence,
  - un numéro d'acquittement.
- ▶ Le corps de chaque paquet comprend un morceau de la requête HTTP.

- ▶ Comme les paquets UDP, chaque paquet TCP est encapsulé dans un paquet IP.
- ▶ Qui sont eux-mêmes encapsulés dans des paquets Ethernet (dans un premier temps).
- ▶ Qui eux transitent au niveau de la couche physique (WiFi, puis ADSL, etc.)

- ▶ Le *routage* des paquets est assuré par le protocole IP.
- ▶ Le paquet est transmis de réseaux en réseaux via des *routeurs*, jusqu'à arriver à destination.
- ▶ Un routeurs est un dispositif relié à au moins deux réseaux, avec une table de routage qui lui indique quoi faire en fonction de l'adresse de destination d'un paquet.
- ▶ Dans chaque *routeur* sur le trajet il se passe ceci :
  - Si l'adresse de destination est directement accessible, y envoyer le paquet.
  - Si il y a une entrée qui correspond à l'adresse de destination, y router le paquet.
  - Si il existe une route par défaut, y envoyer le paquet.
  - Sinon renvoyer un message d'erreur.
- ▶ Il existe différents protocoles de routage, en charge aussi du maintien à jour des routes existantes, le plus présent sur internet est BGP (*Border Gateway Protocol*).

- ▶ La réponse HTTP subit le même traitement que la requête, mais depuis le serveur web.
- ▶ Une fois les différents paquets TCP arrivés et remis dans l'ordre, la réponse est recomposée et le contenu de la page web récupéré.
- ▶ Viennent ensuite les requêtes des autres composants de la page web (images, feuilles de styles, scripts, etc.).

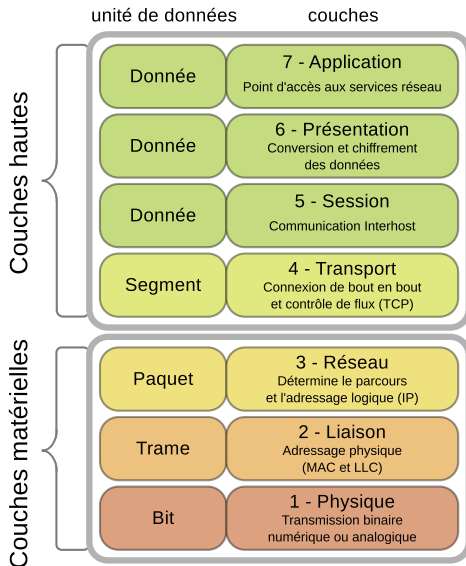


# Présentation du cours

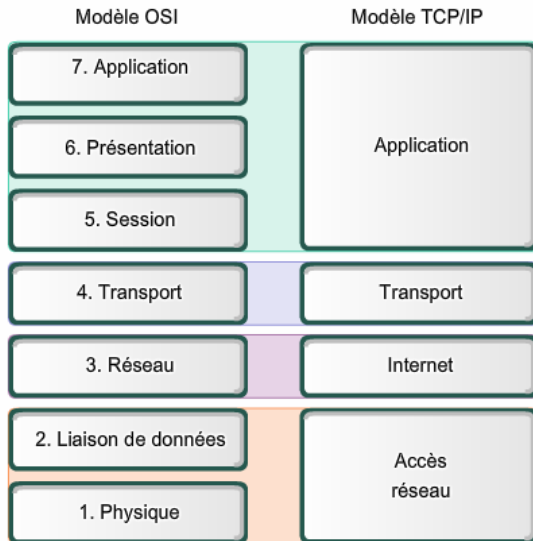
---

► Dans ce cours nous allons étudier :

- les 7 couches du modèle OSI :
  - application,
  - présentation,
  - session,
  - transport,
  - réseau,
  - liaison,
  - physique ;
- et certains des protocoles les plus répandus à chaque couche ;
- la programmation réseau (en particulier dans le modèle TCP/IP, un peu différent du modèle OSI).



## Schéma de comparaison du modèle OSI et du modèle TCP/IP



- En TP on réalisera :
- des versions simplifiées d'outils du type :
    - ping,
    - traceroute,
    - netcat.
  - des implémentations simplifiées de protocoles (et clients/serveurs) ;
  - des séances d'encadrement pour vos projets.

- ▶ Par groupe de 3 à 5.
- ▶ Concevoir en groupe un protocole client-serveur :
  - jeu à plusieurs,
  - tchat,
  - ...,
  - choix libre, à valider avec moi.
- ▶ Chaque membre du groupe implémente un client et un serveur pour le protocole inventé, dans un langage différent des autres membres du groupe.
- ▶ Rapport intermédiaire (en groupe) : les spécifications du protocole.
- ▶ Rendu final (individuel) : les programmes clients et serveurs + rapport.
- ▶ Voir sur la page web du cours pour les dates.

- ▶ L'évaluation de ce cours prendra en compte, par ordre d'importance :
  - le projet,
  - les TPs,
  - l'examen.