

Systèmes et réseaux



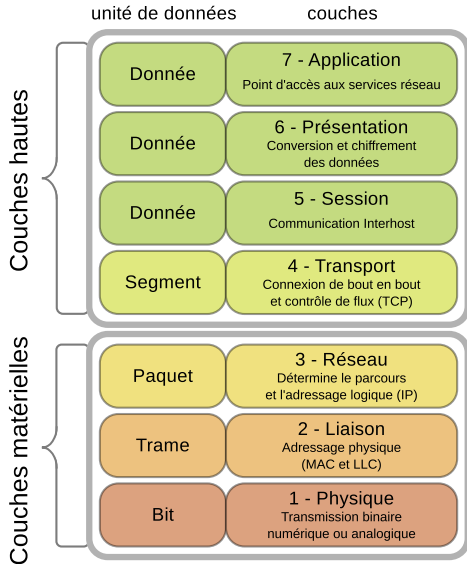
Chapitre 9 Couches application

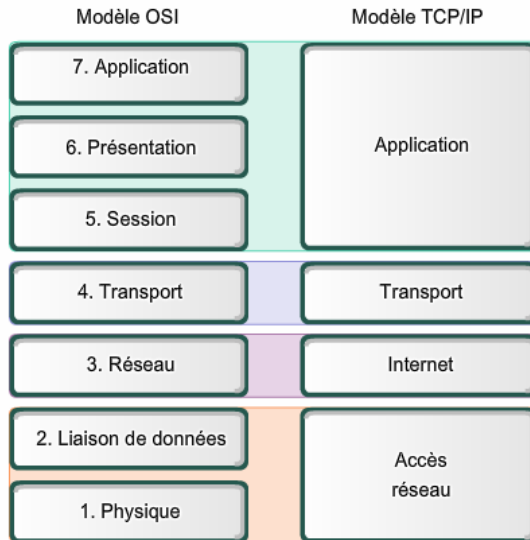


Pablo Rauzy <pr@up8.edu>
pablo.rauzy.name/teaching/sr

Couches application

Rappel





- ▶ La couche session est la cinquième couche du modèle OSI.
- ▶ Elle est en charge de la synchronisation des communications et de la gestion des transactions :
 - permettre à n'importe quel participant de parler à tout moment,
 - rétablir la connexion en cas de coupure,
 - synchroniser deux flux (e.g., audio et vidéo),
 - gérer les communications multipoints.
- ▶ En pratique, donc dans le modèle TCP/IP, la couche session n'a pas vraiment d'existence propre.

- ▶ La couche présentation est la sixième couche du modèle OSI.
- ▶ Les cinq couches du dessous ne transportent que des données brutes, vu simplement comme une suite d'octets, la couche présentation est donc en charge :
 - de l'encodage (ASCII, Latin-1, Unicode, ...),
 - de la compression,
 - du chiffrement (mais en pratique *TLS* fait ça au niveau transport de TCP/IP).
- ▶ En pratique, donc dans le modèle TCP/IP, la couche présentation n'a pas vraiment d'existence propre, et chaque protocole implémente cela à sa façon.

- ▶ Dans le monde TCP/IP, on convertit typiquement tout en texte.
- ▶ MIME signifie *Multipurpose Internet Mail Extensions*.
- ▶ Il s'agit à l'origine d'un standard pour les courriels, permettant de spécifier l'encodage des caractères en les convertissant vers de l'ASCII, seul format supporté nativement par les protocoles de courriel.
- ▶ Aujourd'hui les types MIME prennent aussi en compte les différents formats de médias (document, image, audio, vidéo, etc.).
- ▶ Son utilisation s'est répandue au delà du courriel : HTTP utilise aussi le standard MIME.

Entêtes MIME

- ▶ **Content-type:** donne le type MIME du contenu (par défaut, `text/plain`).
 - C'est ici qu'on précise un *charset* dans le cas de type texte.
 - Exemples :
 - `text/html; charset=utf-8`
 - `image/png`

- ▶ **Content-transfer-encoding:** donne l'encodage de transfert (la méthode de conversion vers ASCII).
 - Ne pas confondre avec le charset.
 - Exemples :
 - `quoted-printable`
 - `base64`

- ▶ Le **Content-type** peut aussi spécifier
 - qu'il y a plusieurs parties indépendantes dans les données (e.g., pour les pièces jointes)
 - `multipart/mixed; boundary="====sep===="`
 - l'existence d'alternatives (e.g., pour envoyer un courriel en texte et en HTML)
 - `multipart/alternative`

- ▶ Voyons comment établir une connexion TCP entre deux services pour pouvoir mettre en œuvre les couches applications par dessus.
- ▶ Rappel :
 - <https://beej.us/guide/bgnet/>

Établir la connexion

▶ `#include <sys/socket.h>`

▶ Côté serveur :

- `int socket(int domain, int type, int protocol);`
- `int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);`
- `int listen(int sockfd, int backlog);`
- `int accept(int sockfd, struct sockaddr *_Nullable restrict addr, socklen_t *_Nullable restrict addrlen);`

▶ Côté client :

- `int socket(int domain, int type, int protocol);`
- `int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);`

▶ `#include <sys/socket.h>`

▶ Envoyer des données :

- `ssize_t send(int sockfd, const void buf[.len], size_t len, int flags);`
 - Si `flags` vaut 0, équivalent à `write`.

▶ Recevoir des données :

- `ssize_t recv(int sockfd, void buf[.len], size_t len, int flags);`
 - Si `flags` vaut 0, équivalent à `read`.

Fermer la connexion

► Rappel :

- `#include <sys/socket.h>`
`int shutdown(int sockfd, int how);`
- `#include <unistd.h>`
`int close(int fd);`