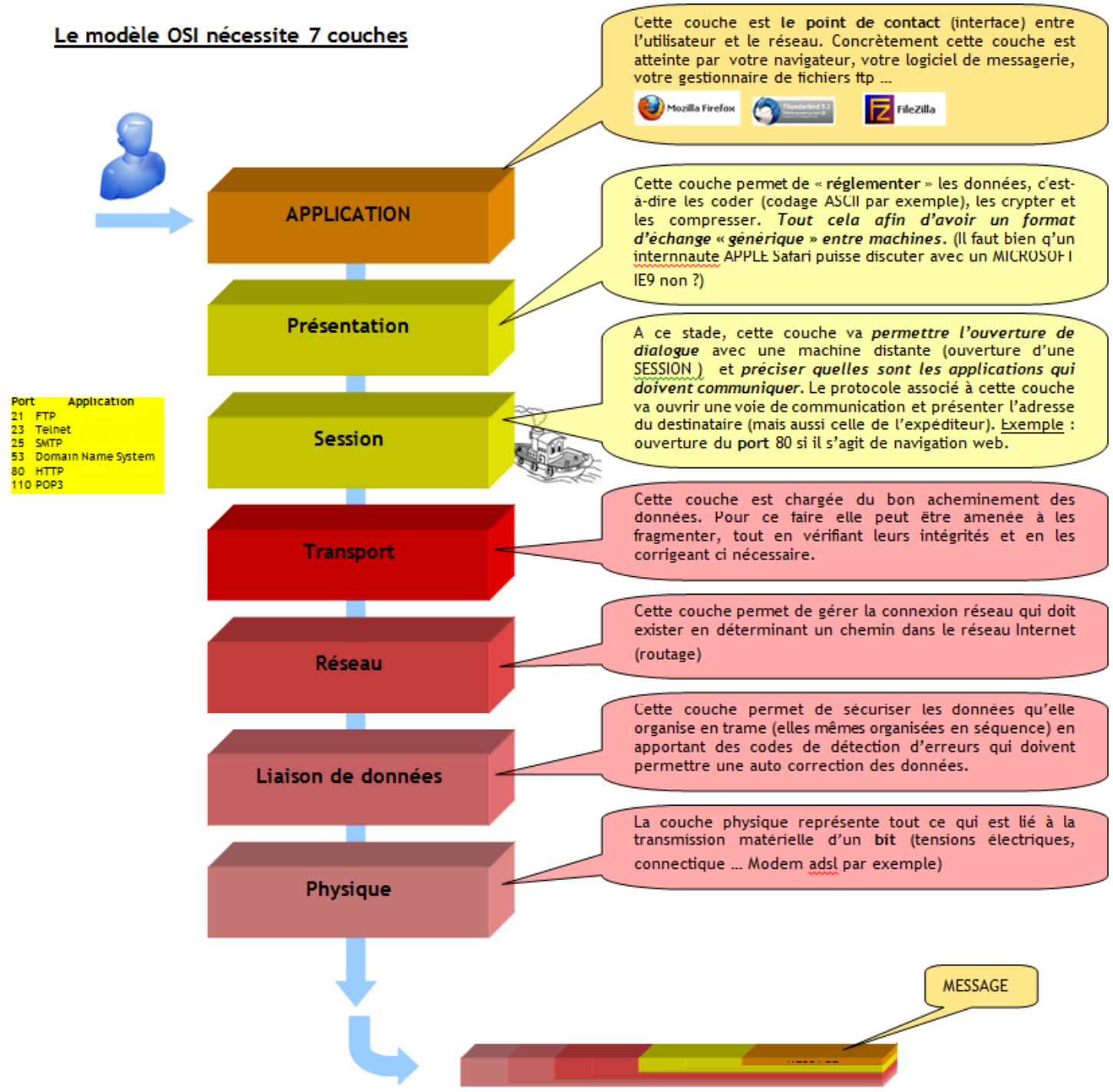
RESSOURCES - RESEAUX

Modèle OSI



Le modèle TCP/IP pour Internet

Et pourquoi pas OSI ?
En fait le modèle TCP/IP était déjà appliqué par les universités américaines au moment où OSI faisait son apparition. Cela n'empêche pas de retrouver dans le modèle TCP/IP des couches présentes dans le modèle OSI.

TCP/IP : **T**ransmission **C**ontrol **P**rotocol on **I**nternet **P**rotocol

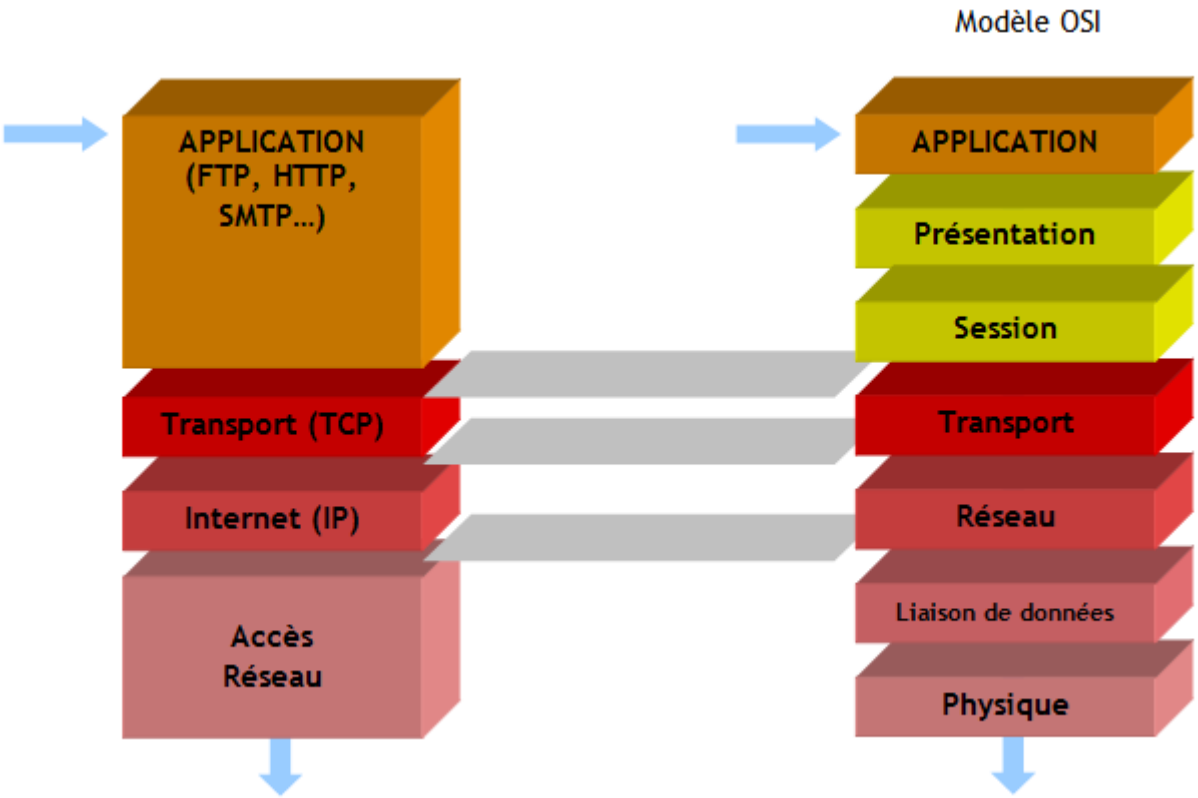
En fait TCP/IP est un assemblage de deux protocoles. Le **TCP** (Transmission Control Protocol) et l'IP (Internet Protocol).

Tout d'abord, l'IP :
C'est certainement le plus connu. C'est le protocole qui permet de transporter un message en informant sur l'adresse du **destinataire** (et également sur l'adresse de **l'expéditeur** … pas d'anonymat !). Ces adresses (IPV4) sont connues sous le nom **d'adresses IP** et se retrouvent sous les formes suivantes :

- 192.168.1.1 ou encore C0.A8.1.1 (en décimal … en hexadécimal)

Ce protocole ne permet cependant pas de s'adresser à une application particulière de la machine destinataire. Il fallait donc adjoindre un protocole de transport permettant cela et assurant également la sécurisation des échanges de données entre expéditeur et destinataire. C'est le protocole **TCP**.
En fait le protocole IP assure la logistique de l'acheminement du message … mais sans contrôle. Il ne peut pas à lui seul donner de la cohérence à un message de type « vidéo » ou « image » qui contient beaucoup de paquets de données.

Le TCP :



Le protocole UDP

Des applications exécutées sur des hôtes différents en réseau et qui veulent s'échanger des données peuvent utiliser principalement deux protocoles de communication au niveau de la couche 4 (transport) :

- Le **protocole UDP** (User Datagram Protocol)

Protocole simple, sans connexion, permettant de transférer des données sans alourdir la communication par du trafic de contrôle.
UDP est chargé de découper un volume important de données en segments ou datagrammes avant leur émission. Les datagrammes seront ré-assemblées dans l'ordre d'arrivée et non dans l'ordre d'émission. Si les datagrammes suivent des routes différentes entre l'émetteur et le récepteur (cas de l'internet), l'application exécutée sur le récepteur doit se charger de ré-ordonner les datagrammes.
UDP rajoute seulement 8 octets d'information en tête de chaque segment de données à transporter dont principalement le port UDP source (2 octets) et le port UDP de destination (2 octets) chargés d'identifier, sur chaque hôte, les applications qui communiquent.
UDP est très utilisé pour les applications comme les jeux en ligne, les services DHCP, DNS, etc.

Remarque : contrairement à UDP,
- TCP impose une connexion entre les applications. Cela accroît les fonctionnalités mais alourdit aussi les communications par du trafic réseau supplémentaire ;
- TCP prévoit la livraison des segments de données dans l'ordre d'émission ;
- TCP assure la fiabilité de l'acheminement
- TCP prévoit la régulation des vitesses de transmission par un système de contrôle de flux afin d'éviter les engorgements.
- TCP rajoute 20 octets en tête de chaque segment de données à transporter dont principalement le port TCP source (2 octets) et le port TCP destination (2 octets) chargés d'identifier, sur chaque hôte, les applications qui communiquent, ainsi que les informations nécessaires au ré-assemblage des segments.

Le protocole ICMP

ICMP, Internet Control Message Protocol, est un protocole de même niveau que l' IP (couche réseau/3) permettant de véhiculer des messages de contrôle et d'erreur pour tester la connectivité à un hôte ou un service.
La commande **ping** utilise le type ICMP « echo » : l'émetteur envoie une demande de renvoi d'informations (echo-request). L'hôte récepteur doit renvoyer le type ICMP « reponse d'echo » (echo-reply). L'interprétation de cette (non)réponse donne des indications sur la connectivité entre les deux hôtes (temps de réponse, etc.).

Le protocole ARP

Le protocole **ARP** (Address Resolution Protocol) permet de demander son adresse MAC à un hôte dont on connaît l'adresse IP par diffusion (broadcast)
Une table **ARP** contient les couples **Adresse IP/Adresse MAC** des hôtes avec lesquels la station a initié une communication. Elle permet donc d'associer de manière transparente pour l'utilisateur, les adresses physiques MAC (utilisées au niveau de la couche 2 (liaison du modèle OSI)) aux adresses logiques IP (utilisées par les couches 3 : réseau et supérieures).

Remarque : L'adresse **MAC** (Media Access Control) est un identifiant unique permettant de déterminer les adresses source et de destination sur un réseau Ethernet (couche 2.Liaison du modèle OSI).

Encapsulation

Les données envoyées sur le réseau traversent la pile de protocoles de haut en bas sur l'hôte émetteur et de bas en haut sur l'hôte récepteur.
Sur l'hôte émetteur, à chaque couche du modèle OSI traversée, le processus d'émission rajoute une étiquette aux données de la couche supérieure afin de garantir un échange correct avec la couche équivalente sur l'hôte récepteur. Ce phénomène est appelé « **encapsulation** ».
Sur l'hôte récepteur les messages sont décodés de bas en haut. Chaque couche vérifie le message lui parvenant de la couche inférieure et envoie à la couche supérieure le message dépouillé de l'étiquette ajoutée lors de l'émission.

- Datagramme UDP

Comme on l'a vu précédemment, UDP est un protocole simple sans connexion. De ce fait, la couche Transport ne rajoutera que quelques octets aux données de la couche Application (dans notre cas) à transmettre dans l'entête UDP :
- Port UDP source : 2 octets qui identifient l'application émettrice sur l'hôte émetteur du datagramme ;
- Port UDP de destination : 2 octets qui identifient l'application réceptrice sur l'hôte récepteur du datagramme ;
- Longueur : longueur du datagramme y compris l'entête ;
- Somme de contrôle : couvre l'ensemble du datagramme

Entête UDP			
octet 1	octet 2	octet 3	octet 4
b7 … b0	b7 … b0	b7 … b0	b7 … b0
Port UDP source		Port UDP de destination	
Longueur		Somme de contrôle	
Donnée de la couche Application (taille variable) ….			

- Paquet IPV4

Le paquet IPv4 encapsule le segment (datagramme) de la couche Transport afin que le réseau le délivre à l'hôte de destination. L'encapsulation réalisée au niveau de la couche réseau consiste à rajouter au segment (datagramme) des informations nécessaires à l'acheminement du paquet dans un entête IP.
Les principales informations sont :

Adresse IP de destination : 4 octets qui identifient de manière unique l'hôte destinataire du paquet sur le réseau dans le cas d'une communication mono-diffusion (unicast). On pourra aussi trouver une adresse de diffusion (adresse de broadcast) dans le cas d'une diffusion à tous les membres d'un même réseau ou une adresse de multi-diffusion (multicast) dans le cas d'une diffusion vers un nombre restreint d'hôtes.

Adresse IP source : 4 octets qui identifient l'hôte qui émet le paquet. Permet au récepteur de pouvoir répondre à l'émetteur le cas échéant.

Durée de vie : Temps de vie restant du paquet avant sa destruction. Chaque routeur traversé décrémente au moins de 1 cette valeur. Quand le compteur arrive à zéro, le paquet est détruit. Cela empêche qu'un paquet tourne « en boucle » sur l'Internet.

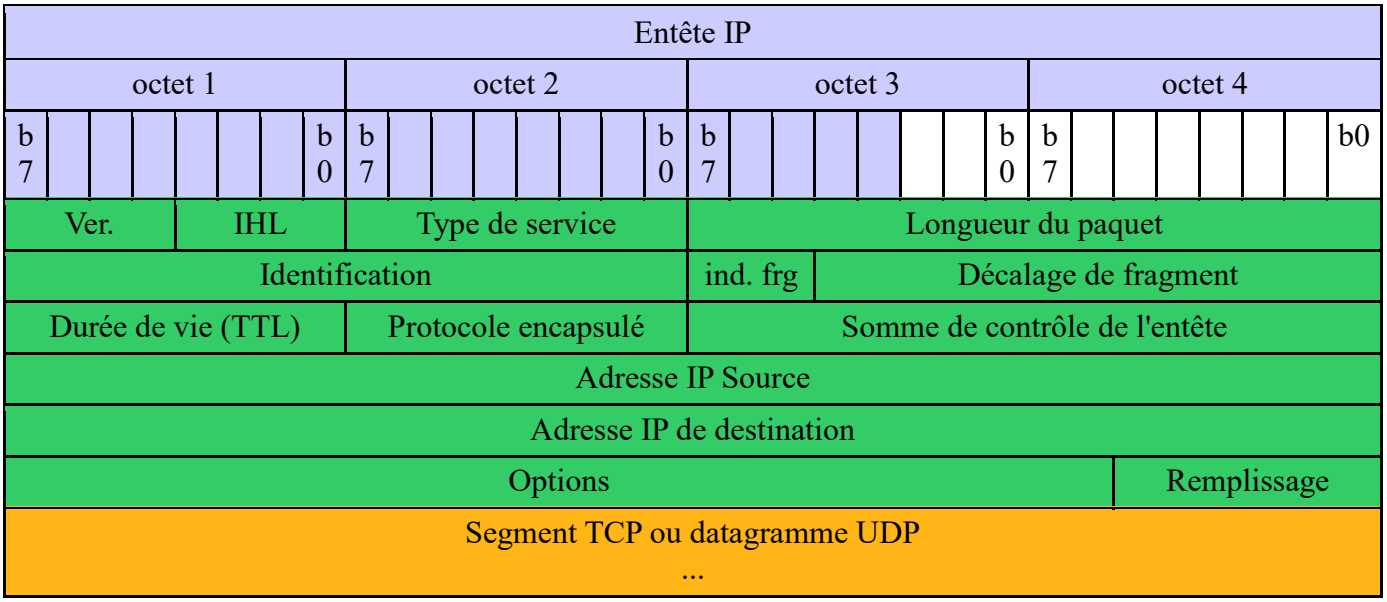
Version : version du protocole IP (4 pour v4)

IHL : longueur de l'entête (multiple de 4 octets).

Longueur du paquet : taille du paquet entier avec entête (en octet).

Protocole encapsulé : donne le type des données encapsulées (segment TCP/datagramme UDP, etc.) :

1 : ICMP (ping)
6 : TCP
17 : UDP
etc.



- Trame Ethernet

Les périphériques Ethernet (cartes réseau) s'échangent des trames dont la taille est comprise entre 64 et 1522 octets (norme IEEE 802.3ac). Les trames dont la taille est hors norme sont simplement ignorées par les périphériques.
Une trame Ethernet est composée de plusieurs champs :

Préambule : succession de 1 et 0 sur 8 octets se terminant par 11 permettant de synchroniser les périphériques récepteurs sur la trame qui va arriver. Généralement, le préambule n'est pas affiché par les analyseurs de réseau (ex. wireshark).

Adresse de destination : 6 octets identifiant le destinataire. Dans le cas de la **mono-diffusion (unicast)**, les récepteurs comparent cette adresse à leur propre adresse MAC et accepte ou rejette la trame suivant le résultat de la comparaison. Une adresse de destination égale à FF:FF:FF:FF:FF:FF identifie une trame de **diffusion** (**broadcast**) destinée à l'ensemble des périphériques. Certaines autres adresses de destination spécifique peuvent adresser un groupe de périphériques (**multidiffusion** ou **multicast**)

Adresse source : 6 octets identifiant le périphérique émetteur.

Type : permet d'identifier le type de protocole encapsulé dans la zone de données (0x0800 pour IPv4, 0x0806 pour ARP, 0x86DD pour IPv6, etc.)

Données : de 46 à 1500 octets de données encapsulées (exemple : paquet IP). Si la taille des données est inférieure à 46 octets, des octets de **bourrage** sont placés en queue pour assurer la taille minimale.

Séquence de contrôle de trame: 4 octets calculés par le périphérique émetteur à partir des octets de la trame Ethernet (CRC, Cyclic Redundancy Check) et placés en queue de trame. Le périphérique récepteur contrôle la validité de la trame en recalculant le CRC puis en le comparant à celui placé par l'émetteur. Une différence indique une trame corrompue du fait d'une perturbation électrique ou électromagnétique. Dans ce cas, la trame sera abandonnée. Dans le cas contraire, la trame est acceptée par le périphérique récepteur.