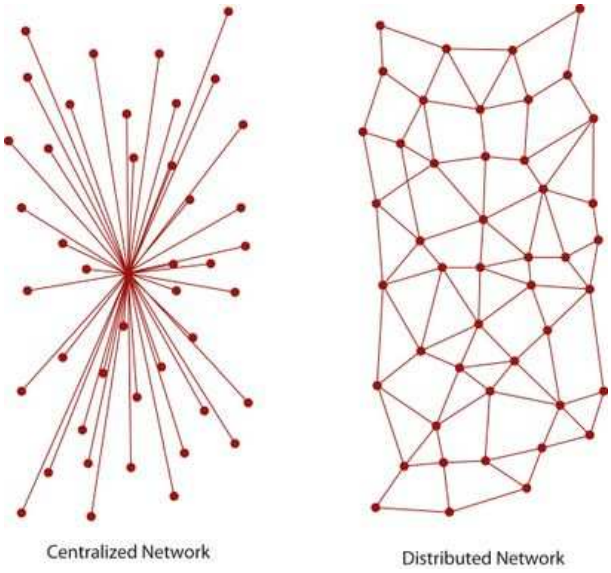
Réseaux

1) Introduction

Dans les **années 1950**, on ne pouvait communiquer qu'avec une seule machine à la fois. Les chercheurs qui devaient communiquer avec leurs collègues lors de réunions, se sont rendu compte qu'il serait intéressant de le faire en temps réel, plutôt que de passer d'un interlocuteur à l'autre successivement.

Ils ont donc cherché à créer un nouveau moyen de communication qui ne serait plus **centralisé**, mais **maillé**.



Ainsi toute information pourrait passer par différents points, et si certains points disparaissaient, cela n'empêcherait pas l'information de circuler, car l'information pourrait passer par différents chemins.

Les chercheurs vont travailler sur cette idée et mettre en place un réseau pour l'armée Américaine début 1960. A la fin des années 1960, l'**Arpanet**, ancêtre d'**Internet**, ne comporte que 4 machines.

Aujourd'hui, **Internet** est un réseau informatique mondial, connectant des centaines de millions de machines, et accessible au public.

Il rend accessible à ses utilisateurs un certain nombre de **services de communication** comme :

- la messagerie ;
- la publication de données (le Web) ;
- la communication instantanée (le chat) :
- les transferts de fichiers.

Il est en effet capable de **transmettre des informations** de toute nature : texte, image et son, de manière très rapide.

Les machines du monde entier sont inter-connectées de telle façon que chacune puisse communiquer avec toutes les autres.

Maintenant, cela nécessite des **règles** pour que tout fonctionne correctement : des **protocoles de communication** tels que **TCP/IP**.

2) Réseau d'ordinateurs

Un **réseau informatique** est un ensemble de composants informatiques (ordinateurs, routeurs, concentrateurs...) reliés entre eux par divers liens (câbles de cuivre, fibre optique, liaisons satellites, ondes radios).

Un **protocole** est un ensemble de règles permettant d'établir, mener et terminer une communication entre deux entités.

On les utilise pour garantir lors de la transmission de données, l'absence d'erreur (pas de perte de données), l'efficacité (le plus rapidement possible), ou la confidentialité (seul le destinataire peut obtenir l'information).

Un **serveur** est un dispositif matériel ou un logiciel exécutant un service. On utilise le même mot pour désigner à la fois la machine où s'exécute le service et le logiciel qui permet son exécution.

On appelle **client** à la fois la machine et l'application se connectant par le réseau à un serveur.

Le modèle **client-serveur** est la principale manière de concevoir des applications en réseau sur Internet. Dans ce modèle, un serveur propose un service ; il est perpétuellement en attente de connexions. Les clients se connectent à tout moment et sont traités de manière individuelle. Ils ne communiquent qu'avec le serveur et ne communiquent pas entre eux.

3) Modèles en couches

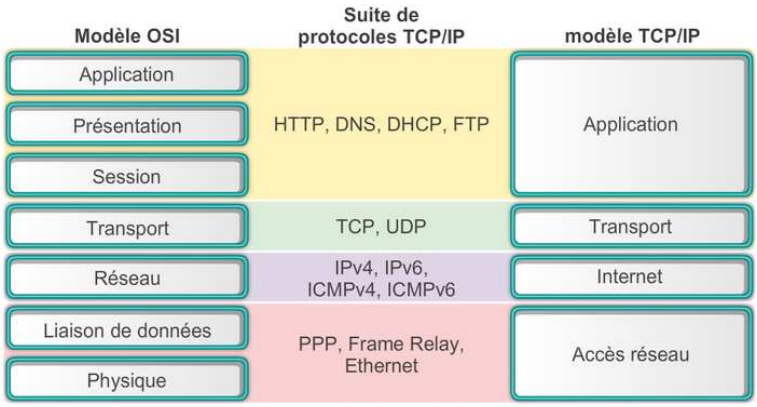
a) Le modèle OSI

Les chercheurs ont créés une norme que devront respecter les personnes se connectant à Internet : le **modèle OSI**.

Il est créé **en 1984**, mais il est apparu après la naissance d'Internet. C'est un modèle théorique, le modèle réellement utilisé est le **modèle TCP/IP** qui s'est développé en même temps que le réseau Internet.

Le modèle OSI indique aux personnes voulant mettre en place des réseaux, comment le faire, pour garantir la **compatibilité** entre les différents matériels.

Le modèle OSI est un **modèle en 7 couches**, chacune ayant un rôle défini. Le modèle TCP/IP, lui, n'en comporte que 4.



Chaque couche est **indépendante** et ne peut communiquer qu'avec une **couche adjacente**.

Ainsi les informations utilisés par une couche ne pourront pas être utilisées par une autre. Cela veut dire qu'on pourra changer un protocole associé à une couche sans avoir besoin de changer toutes les autres couches.

Lors de l'**envoi de données**, on parcours le modèle OSI **de haut en bas**, en traversant toutes les couches.

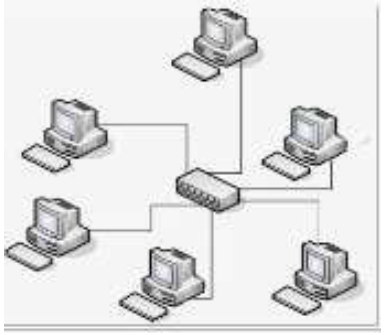
b) Le modèle TCP/IP ou Internet

★ Dans ce modèle, **la couche la plus basse** regroupe tous les aspects physiques du réseau, on l'appelle couche **accès réseau** ou couche **liaison**.

Son rôle est de fournir le **support de transmission** de la communication.

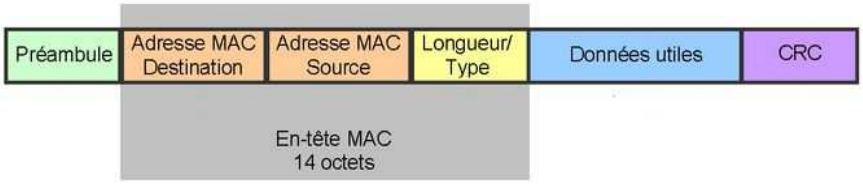
Un **réseau local (LAN)** est constitué de machines reliées directement entre elles par la même technologie (ex : un réseau Wi-Fi ou un réseau câblé utilisant le protocole Ethernet...)

Dans le cas d'un réseau Ethernet, elles sont reliées aux autres par un **périphérique réseau** (Concentrateur/-Hub ou Commutateur/Switch).



Chaque machine du réseau local possède au moins une **carte réseau** possédant une adresse codées sur 6 octets en hexadécimal, l'**adresse MAC** (*Exemple :* **5E :FF :56 :A2 :AF :15**).

Lorsqu'une machine souhaite communiquer avec une autre, elle envoie un paquet d'octets appelé **trame Ethernet**



On remarque notamment que cette trame contient un entête, puis l'adresse MAC du destinataire, celle de la machine source, la taille des données que l'on souhaite envoyer, les données et enfin une séquence de contrôle.

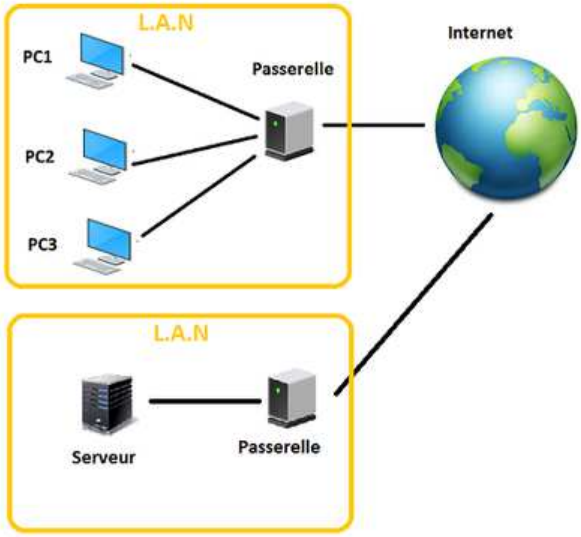
★ La **couche Internet** permet de relier les réseaux locaux entre eux grâce au **protocole IP**. Il permet d'associer à chaque interface de chaque machine un unique identifiant appelé **adresse IP**, indépendamment de la couche liaison sous-jacente, qui permet de la contacter.

La version actuelle du protocole est **IPv4** : l'adresse est représentée par 4 octets, notés en décimal (de 0 à 255) et séparés par des points (par exemple : 149.56.108.199).

Ce protocole permet de savoir si il y a un **problème de connexion** entre la machine et la machine distante, si la connexion est de mauvaise qualité...

Le deuxième rôle du protocole IP est **le routage**, la transmission de proche en proche des données depuis l'émetteur jusqu'à la destination finale. En effet, toutes les machines n'étant pas directement connectées les unes aux autres, le protocole permet de définir **comment acheminer les donnée**s.

Dans un réseau local, une des machines joue le rôle de **passerelle** (gateway) ou de **routeur** : si la machine que l'on souhaite joindre est dans le même réseau local, on lui envoie directement les données ; sinon, on les envoie vers la passerelle. Celle-ci regarde l'adresse IP et envoie les données vers un autre réseau auquel elle est connectée et ainsi de suite jusqu'à arriver à destination.



Pour mener à bien ce protocole de routage, le protocole IP définit la notion de **paquet IP**. Celui-ci est **encapsulé dans une trame Ethernet**. Il est constitué d'un préfixe, des adresses IP du destinataire et de la source et des données.

Le préfixe contient notamment un entier codé sur un octet appelé **TTL** (Time to live ou durée de vie). À chaque routeur rencontré sur le chemin, ce nombre est décrémenté et s'il prend la valeur 0, le paquet est détruit, pour éviter que les paquets ne restent trop longtemps en transfert sur le réseau sans trouver leur destination.

Pour savoir si la machine destinataire des données appartient au même sous-réseau que la machine source, et si l'on peut lui envoyer directement les données, on utilise en plus de l'adresse IP, un **masque de sous-réseau** (exemple : **255.255.255.0**).

Celui-ci permet de "découper" l'adresse IP en deux parties : la **partie réseau** (les bits égaux à 1) et la **partie machine** (les bits égaux à 0). Cela permet d'avoir un certain nombres d'adresses distinctes pour les différentes machines d'un même sous-réseau.

Par exemple, si l'adresse IP 192.168.0.1 est associée au masque 255.255.255.0 ;

255.255.255.0 s'écrit 11111111 11111111 11111111 00000000 en binaire se qui permet de savoir que 192.168.0. est la partie de l'adresse correspondant au réseau local et 1 est l'adresse de la machine sur ce réseau local. Ainsi, une autre machine de ce même réseau pourrait avoir comme adresse 192.168.0.2.

L'adresse IP, le masque de sous réseau et l'adresse du routeur font partie des paramètres de configuration d'un réseau IP.

★ Le protocole IP permet de créer une communication entre deux machines physiques.

Sur une même machine, il y a souvent plusieurs services réseaux : mails, web... Il faut donc identifier pour une même adresse IP donnée, avec quel service on souhaite communiquer. C'est le premier rôle des protocoles de la **couche transport** que sont **UDP** et **TCP**.

Le **protocole UDP** spécifie l'adresse IP et le **numéro de port** sur lequel l'application serveur est en attente de connexion, c'est à dire un identifiant numérique associé à un service particulier.

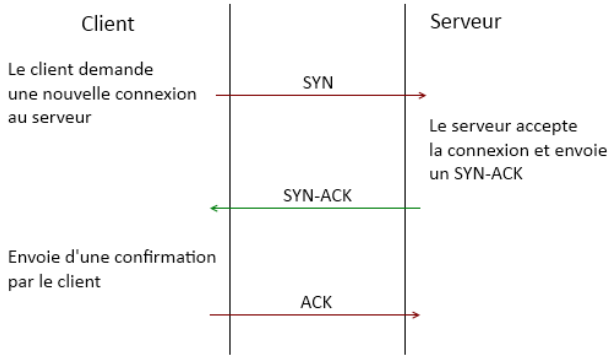
Par exemple, le numéro de port par défaut pour le Web est le 80 (pour HTTP).

Ensuite il envoie un **datagramme** contenant les numéros des ports destination et source, puis les données. Ce datagramme est encapsulé dans le paquet IP.

Le protocole UDP ne permet pas de savoir si un les données sont perdues, et si plusieurs datagrammes sont envoyés à la suite, il ne peut garantir qu'ils seront reçu dans le bon ordre.

Le **protocole TCP** résout ces problèmes grâce à un système d'**accusés de réception**. Il permet d'envoyer des données de **taille arbitraire**, **dans l'ordre**, de **détecter les erreurs** de transmission et, si besoin, de **retransmettre automatiquement les fragments** de données perdus ou corrompus.

Lorsqu'un client souhaite se connecter à un serveur avec le protocole TCP, il initie une **mise en place de connexion en trois temps** (three way handshake).



Le paquet de données est coupé en paquets de même taille (pour des raisons matérielles ou de performance), que l'on envoie avec un numéro.

Ainsi, si le serveur reçoit les **paquets dans le désordre**, il peut les remettre dans l'ordre et reconstituer le paquet initial.

Si le serveur **ne reçoit pas l'un des paquets**, il n'envoie pas le ACK correspondant au client, qui peut alors choisir de ré-émettre le paquet manquant.

Enfin, si le serveur reçoit **des paquets en double**, alors il peut détruire les paquets ayant les mêmes séquences.

Ce processus permet de réaliser un **canal de communication avec détection d'erreurs**.

★ La **couche application** comporte plusieurs protocoles comme le **protocole HTTP** (pour communiquer avec le Web), le **protocole FTP** (pour le partage de fichiers), le **protocole DNS** ...

Pour envoyer un message à une machine, on a besoin de son adresse IP, mais cette suite de nombres est difficile à retenir. De plus, il peut arriver qu'un site Web change d'hébergeur et donc d'adresse IP. C'est pour cela que l'on a mit en place le **protocole DNS** (Domain Name User) qui utilise un système de **noms de domaines**, avec des **adresses symboliques**.

Par exemple : le site web Youtube.

Il a pour adresse symbolique : **www.youtube.fr** et pour adresse IP : **216.58.206.238**.

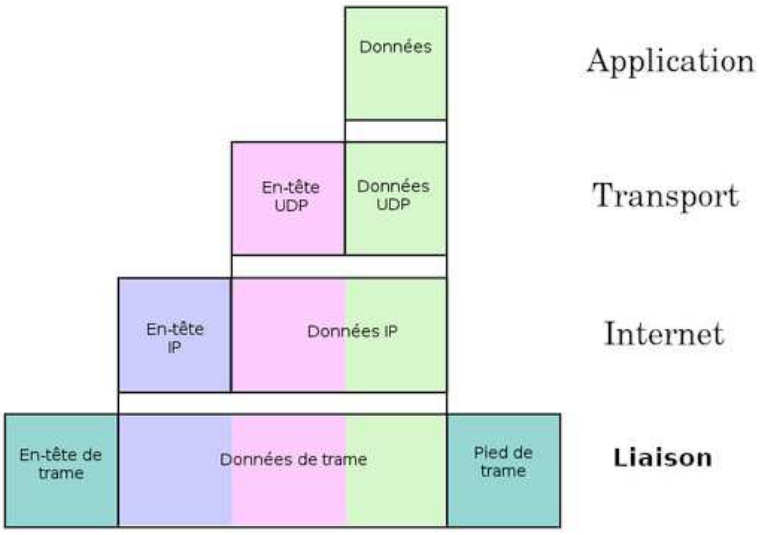
Les **serveurs DNS** sont des machines connectées au réseau internet qui **convertissent** les adresses symboliques des sites Web en adresses IP du serveur qui les héberge et inversement, comme un annuaire téléphonique.

Ils sont **constamment mis à jour**, dès qu'une nouvelle adresse est créée.

Lorsqu'un client entre une adresse symbolique, sont fournisseur d'accès internet (FAI) **envoie une requête** à un serveur DNS.

Celui-ce donnera l'adresse IP nécessaire à l'envoi du message ou indiquera quel autre serveur peut communiquer la réponse.

Pour conclure, il faut retenir que lorsqu'un client envoie des données sur un réseau, celles-ci **traversent toutes les couches** ; à chaque fois, on y ajoute de nouvelles données spécifiques aux protocoles utilisés. Elles sont **encapsulées**, comme le résume le diagramme ci-dessous :



Une fois les données reçues par le serveur, elles traversent à nouveau toutes les couches dans le sens inverse ; elles sont **désencapsulées**.

c) Processus d'acquittement

Il se peut, et cela est courant que des paquets se perdent...

Les causes possibles sont nombreuses ; par exemple :

- engorgement d'un serveur
- délai d'attente trop long qui entraîne la destruction du paquet

Le protocole TCP contrôle l'envoi et la bonne réception des paquets avec des **accusés de réception** (**ACK** : acknowledgement ou acquittement en Français)

De manière générale, **les processus d'acquittement** permettent de détecter les pertes de données, l'idée étant qu'en cas de perte, l'émetteur renvoie les données perdues au destinataire.

Un exemple de processus d'acquittement est le **protocole de bit alterné**.

Il est implémenté au niveau de la **couche "Accès réseau"** du modèle TCP/IP, il ne concerne donc pas les paquets, mais les **trames**.

Le principe de ce protocole est simple, considérons 2 ordinateurs en réseau :

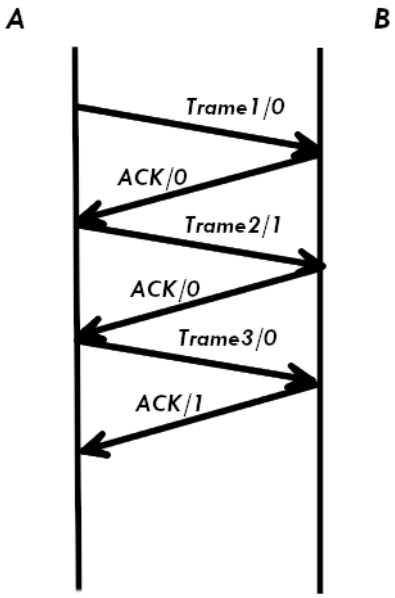
- un **ordinateur A** qui sera l'émetteur des trames et un **ordinateur B** qui sera le destinataire des trames.

Au moment d'émettre une trame, A va **ajouter à cette trame un bit (1 ou 0) appelé drapeau** (flag en anglais).

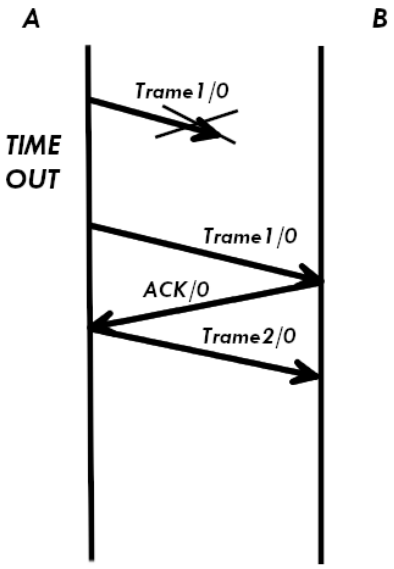
B va envoyer un **accusé de réception** (acknowledge en anglais souvent noté ACK) à destination de A dès qu'il a reçu une trame en provenance de A. À cet accusé de réception on associe aussi un bit drapeau (1 ou 0).

- La première trame envoyée par A aura pour drapeau 0, dès cette trame reçue par B, ce dernier va envoyer un accusé de réception avec le drapeau 1 (ce 1 signifie "la prochaine trame que A va m'envoyer devra avoir son drapeau à 1"). Dès que A reçoit l'accusé de réception avec le drapeau à 1, il envoie la 2e trame avec un drapeau à 1, et ainsi de suite...
- Le système de drapeau est complété avec un **système d'horloge** côté émetteur.

Un "chronomètre" est déclenché à chaque envoi de trame, si au bout d'un certain temps, l'émetteur n'a pas reçu un acquittement correct (avec le bon drapeau), la trame précédemment envoyée par l'émetteur est considérée comme perdue et est de nouveau envoyée.

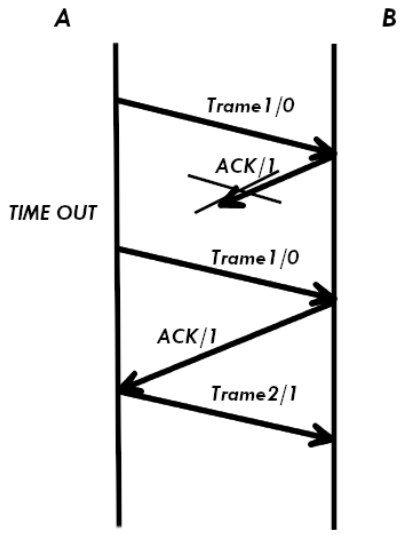


Exemple 1 : **La trame est perdue**



Au bout d'un certain temps ("TIME OUT") A n'a pas reçu d'accusé de réception, la trame est considérée comme perdue, elle est donc renvoyée.

Exemple 2 : **L'accusé de réception est perdu**



A ne reçoit pas d'accusé de réception avec le drapeau à 1, il renvoie donc la trame 1 avec le drapeau 0. B reçoit donc cette trame avec un drapeau à 0 alors qu'il attend une trame avec un drapeau à 1 (puisqu'il a envoyé un accusé de réception avec un drapeau 1), il "en déduit" que l'accusé de réception précédent n'est pas arrivé à destination : il ne tient pas compte de la trame reçue et renvoie l'accusé de réception avec le drapeau à 1. Ensuite, le processus peut se poursuivre normalement.

Dans certaines situations, le protocole de bit alterné **ne permet pas de récupérer les trames perdues**, c'est pour cela que ce protocole est aujourd'hui remplacé par des protocoles plus efficaces, mais aussi plus complexes.