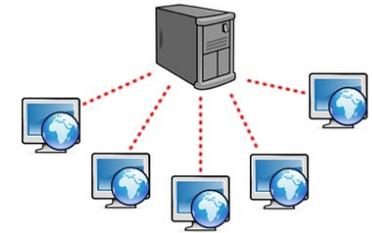
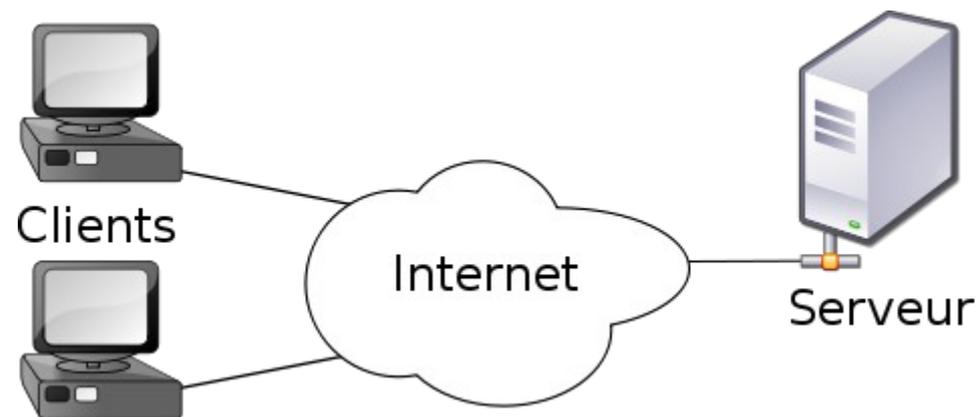


Le modèle client-serveur



De nombreuses applications fonctionnent selon un **environnement client-serveur**, cela signifie que des **machines clientes** contactent un **serveur**, une machine généralement très puissante en terme de capacités d'entrée-sortie, qui leur fournit des **services**. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc.

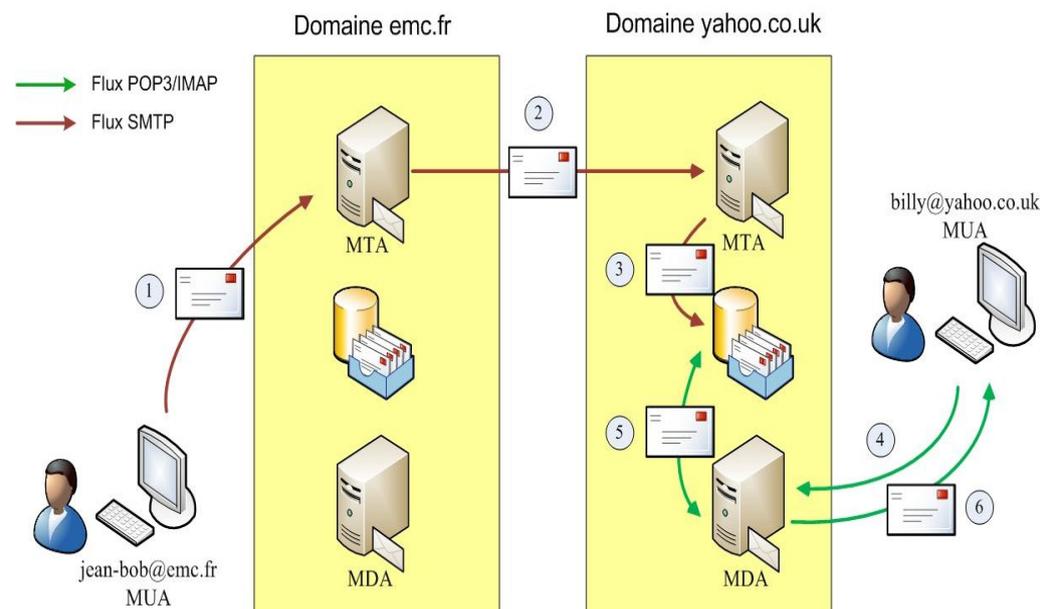


Architecture client - serveur

Le modèle client-serveur

Les services sont exploités par des programmes, appelés **programmes clients**, s'exécutant sur les **machines clientes**.

On parle ainsi de client (client FTP, client de messagerie, etc.) lorsque l'on désigne un programme tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès d'un serveur (dans le cas du client FTP il s'agit de fichiers, tandis que pour le client de messagerie il s'agit de courrier électronique).



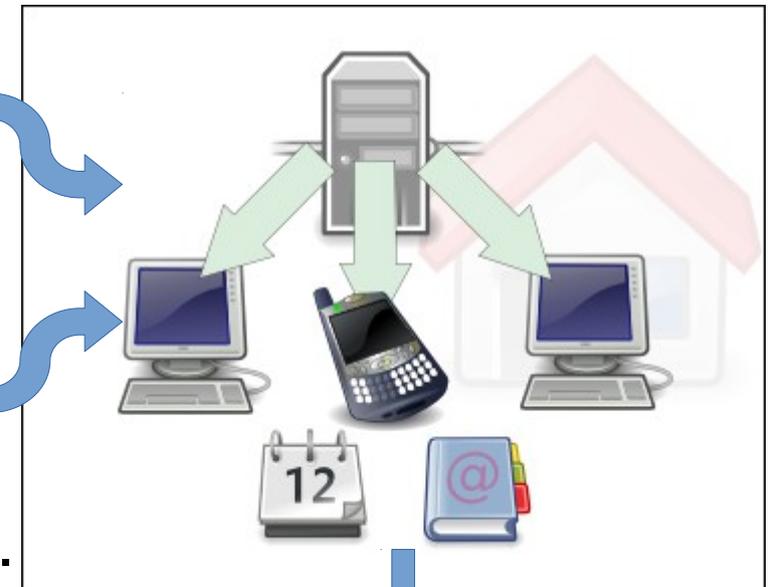
Par extension, le client désigne à la fois le logiciel client et l'ordinateur sur lequel est exécuté le logiciel client, et le serveur, le programme serveur et l'ordinateur sur lequel est exécuté le logiciel serveur.

Le modèle client-serveur en résumé

Les serveurs sont généralement des ordinateurs dédiés au **logiciel serveur** qu'ils abritent, et dotés de capacités supérieures à celles des ordinateurs personnels en termes de puissance de calcul, d'entrées-sorties et de connexions réseau.

Un serveur peut **répondre aux requêtes d'un grand nombre de clients.**

Les clients sont souvent des ordinateurs personnels ou des appareils individuels (téléphone, tablette) munis de **logiciel client** qui **font des requêtes** aux serveurs.



Les services rendus au client par le serveur

Le modèle client-serveur

Il existe une grande variété de logiciels serveurs et de logiciels clients en fonction des besoins à servir :

- un **serveur web** publie des pages web demandées par des navigateurs web ;
- un **serveur de messagerie électronique** envoie des mails à des clients de messagerie ;
- un **serveur de fichiers** permet de stocker et consulter des fichiers sur le réseau ;
- un **serveur de données** à communiquer des données stockées dans une base de données ;
- un **contrôleur de domaine** permet de gérer les utilisateurs et les droits d'accès sur un réseau, etc.

Les serveurs : caractéristiques

- Machines puissantes (processeur, RAM)
- Système de tolérance de pannes (Disques RAID)
- Version Tour ou rack
- Virtualisation des serveurs : une machine physique = plusieurs serveurs (Exemple : une machine = serveur des comptes utilisateur, d'impression, de base de données, ...)
- Doivent être équipés d'un système de ventilation
- Sont protégés contre les coupures électriques (onduleur)

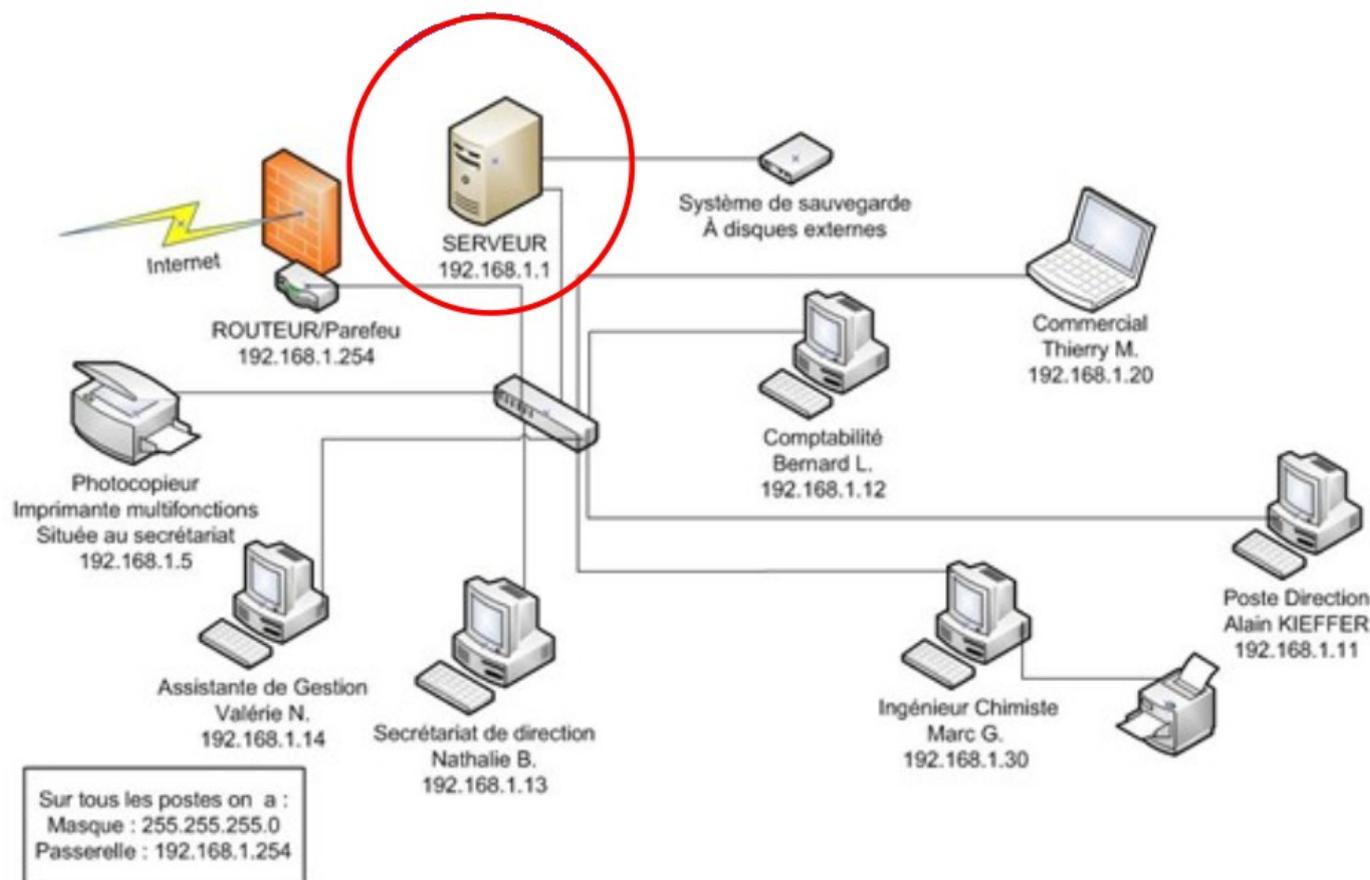


[lien vidéo d'un data center](#)

Architecture client - serveur

Les serveurs / clients dans un réseau local

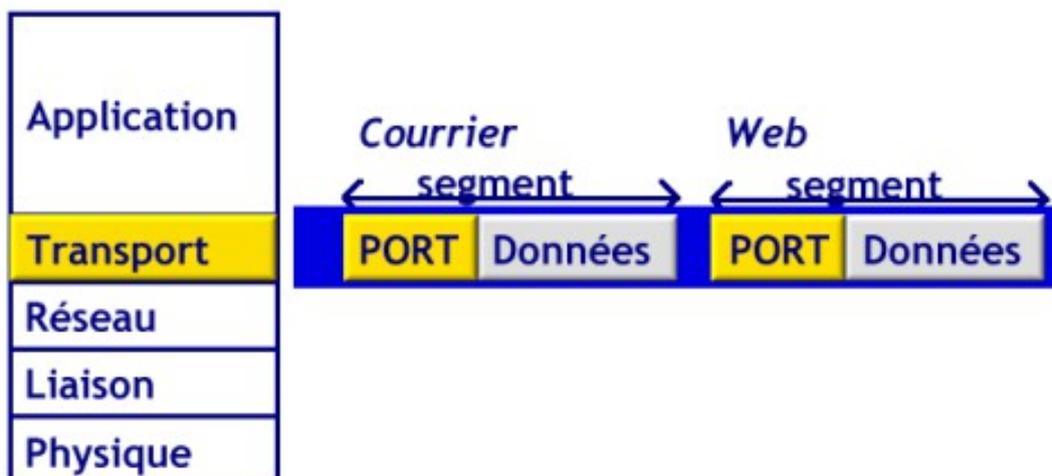
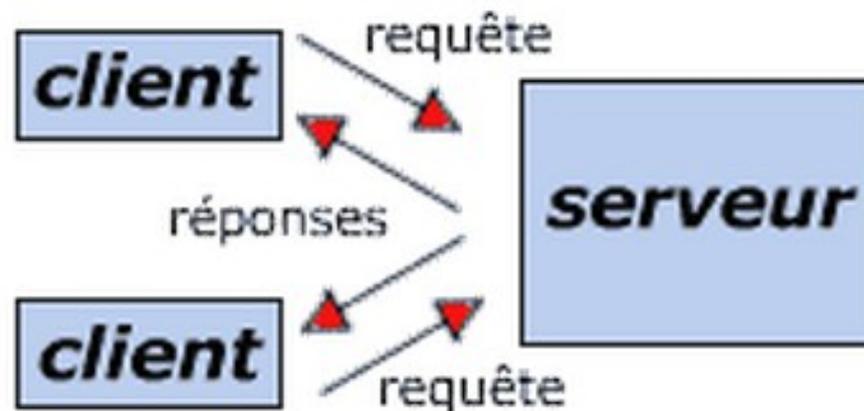
- Les **serveurs** ont des **adresses IP fixes** (statiques)
- Pour les **machines hôtes**, on peut utiliser des **adresses dynamiques** (DHCP)



Architecture client - serveur

La communication entre le client et le serveur

Le client émet une **requête** vers le serveur grâce à son **adresse IP** et le **port**, qui désigne un **service particulier du serveur**, c'est le rôle de la couche **TCP**.

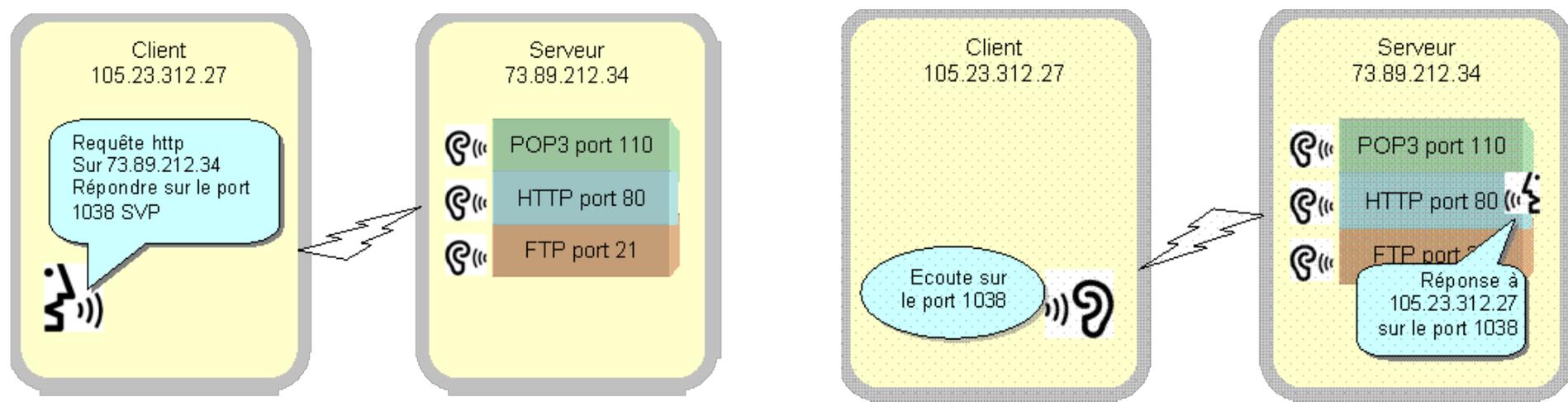


Architecture client - serveur

La communication entre le client et le serveur

L'adresse IP permet d'identifier de façon unique une machine sur le réseau.

Le numéro de port indique l'application à laquelle les données sont destinées. Ce numéro est codé sur 16 bits et peut donc prendre les valeurs allant de 0 à 65535. **Certains numéros de port sont réservés (numéros 0 à 1023)** : le port 21 désigne le service FTP, le port 80 le service HTTP, ...

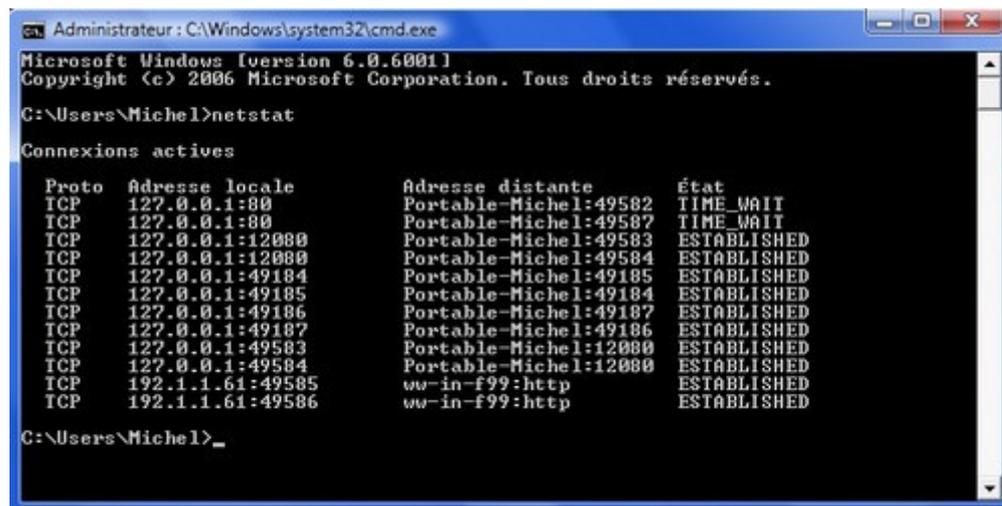


Architecture client - serveur

La communication entre le client et le serveur

Comment voir les ports ouverts ? il s'agit de la commande **netstat**, plus exactement *netstat -an* pour Windows et mac, et *netstat -antp* pour Linux.

A la connexion d'un client sur le port en écoute, le serveur ouvre un **socket** (« connecteur réseau » en Français) puis communique avec le client, suivant le protocole prévu. Le socket est l'interface qui permet la communication entre les 2 machines.



```
Administrateur : C:\Windows\system32\cmd.exe
Microsoft Windows [version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. Tous droits réservés.

C:\Users\Michel>netstat

Connexions actives

Proto  Adresse locale      Adresse distante    État
TCP    127.0.0.1:80         Portable-Michel:49582  TIME_WAIT
TCP    127.0.0.1:80         Portable-Michel:49587  TIME_WAIT
TCP    127.0.0.1:12080     Portable-Michel:49583  ESTABLISHED
TCP    127.0.0.1:12080     Portable-Michel:49584  ESTABLISHED
TCP    127.0.0.1:49184     Portable-Michel:49185  ESTABLISHED
TCP    127.0.0.1:49185     Portable-Michel:49184  ESTABLISHED
TCP    127.0.0.1:49186     Portable-Michel:49187  ESTABLISHED
TCP    127.0.0.1:49187     Portable-Michel:49186  ESTABLISHED
TCP    127.0.0.1:49583     Portable-Michel:12080  ESTABLISHED
TCP    127.0.0.1:49584     Portable-Michel:12080  ESTABLISHED
TCP    192.1.1.61:49585    ww-in-f99:http        ESTABLISHED
TCP    192.1.1.61:49586    ww-in-f99:http        ESTABLISHED

C:\Users\Michel>_
```

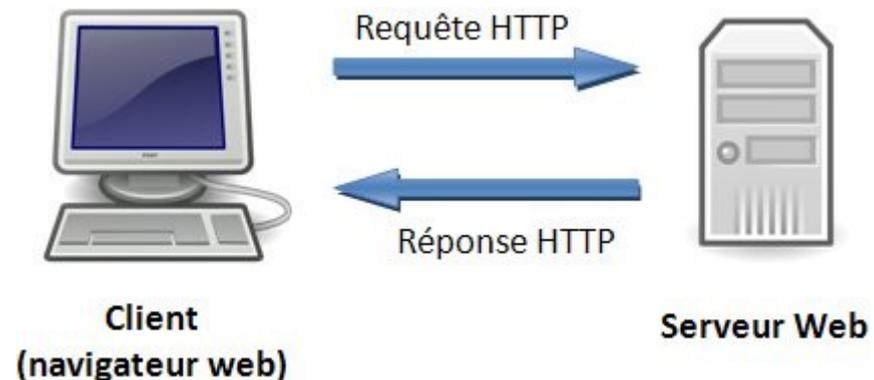
Architecture client - serveur

Le serveur Web port 80

La consultation de pages sur un site Web a un fonctionnement basé sur une **architecture client-serveur**.

Un internaute connecté au réseau via son ordinateur et un navigateur Web est le **client**, le **serveur** est constitué par le ou les ordinateurs contenant les applications qui délivrent les pages demandées. Dans ce cas, c'est le **protocole de communication HTTP**, pour *HyperText Transfer Protocol*, qui est utilisé.

HTTP est un protocole de la **couche application** (couche 7 du modèle OSI). Il utilise le protocole TCP comme couche de transport. Un serveur HTTP utilise alors par défaut le port 80 (443 pour HTTPS).

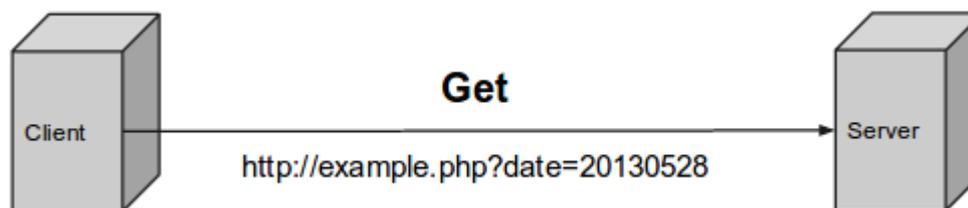


Architecture client - serveur

Le serveur Web : les méthodes GET et POST

Dans le protocole HTTP, une **méthode** est une commande spécifiant un type de requête, c'est-à-dire qu'elle demande au serveur d'effectuer une action. Les méthodes les plus utilisées sont **GET** et **POST** :

- **GET** : c'est la méthode la plus courante pour demander une ressource.



- **POST** : cette méthode doit être utilisée pour ajouter une nouvelle ressource, comme un message sur un forum, un article dans un site ou encore un login et un mot de passe.

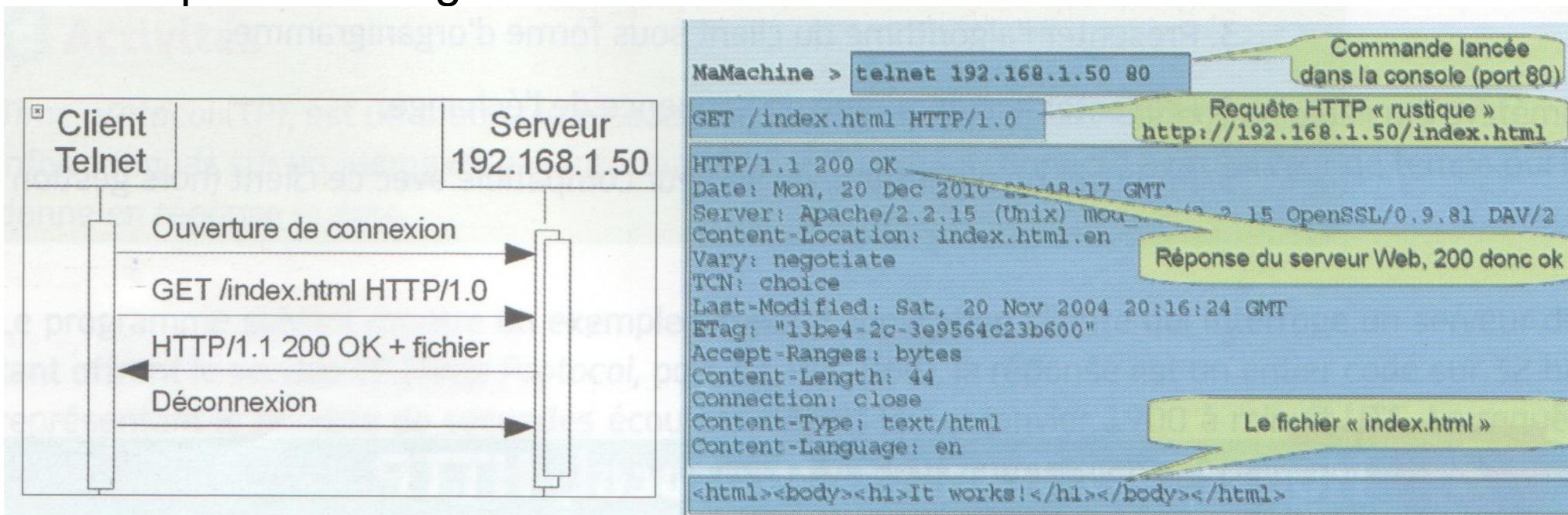


Architecture client - serveur

Le serveur web : le dialogue HTTP

La commande **telnet** exécutée dans un terminal permet de « simuler » un navigateur.

Exemple de dialogue :



Ex : La réponse du serveur contient « Content-Length: 44 ». Que représente la valeur 44 ? Comment pouvez-vous la vérifier ?

Architecture client - serveur

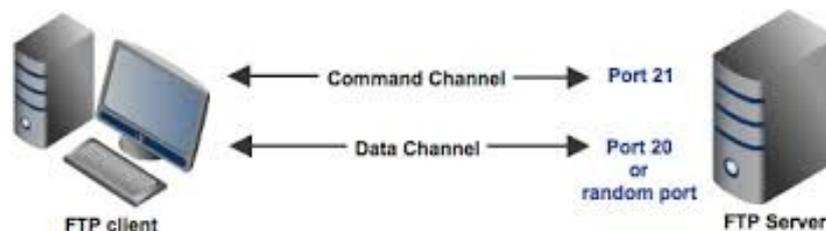
Le serveur de fichiers : le protocole FTP

Le protocole de transfert de fichiers **FTP** pour File Transfer Protocol, est un protocole de communication destiné à l'échange informatique de fichiers sur un réseau TCP/IP. Il permet, depuis un ordinateur, de copier des fichiers vers un autre ordinateur du réseau, d'alimenter un site Web, ou encore de supprimer ou de modifier des fichiers sur cet ordinateur.

FTP obéit à un modèle client-serveur, c'est-à-dire qu'une des deux parties, le client, envoie des requêtes et le serveur répond.

Le protocole FTP appartient à la **couche application** (couche 7 du modèle OSI) et utilise une connexion TCP.

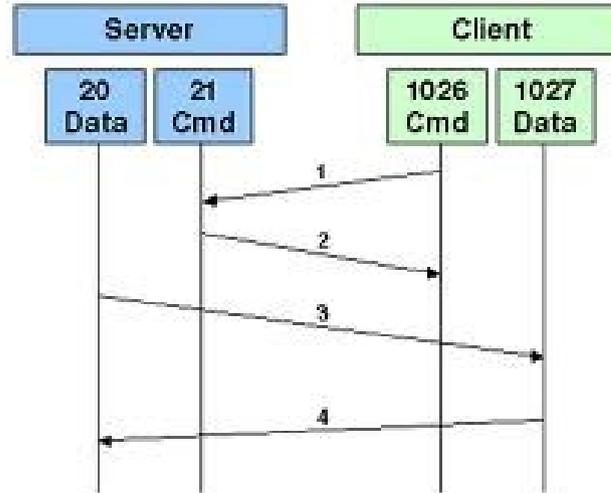
Deux ports sont réservés pour les connexions FTP : le port 21 pour les commandes et le port 20 pour les données.



Architecture client - serveur

Le serveur de fichiers : communications

Le **diagramme d'échange** ci-dessous, illustre l'échange de données entre un client et le serveur.



Architecture client - serveur

Le serveur de fichiers : exemple de communications

Ex : un échange entre un client et un serveur a été intercepté par un logiciel de capture de trame, Wireshark.

En utilisant le tableau ci-dessous répondre aux questions :

- Quelle est l'adresse IP du serveur ?
- Quelle est l'adresse IP du client ?
- Quel utilisateur se connecte au serveur ?
- Quel est son mot de passe ?
- FTP est-il un protocole sécurisé ?

Source	Destination	Protocol	Info
192.168.1.50	192.168.1.45	FTP	Response: 220 192.168.1.50 FTP server (tnftpd 2.0080929) ready.
192.168.1.45	192.168.1.50	FTP	Request: USER test
192.168.1.50	192.168.1.45	FTP	Response: 331 User test accepted, provide password.
192.168.1.45	192.168.1.50	FTP	Request: PASS Chocolat2010
192.168.1.50	192.168.1.45	FTP	Response: 230 User test logged in.
192.168.1.45	192.168.1.50	FTP	Request: SYST
192.168.1.50	192.168.1.45	FTP	Response: 215 UNIX Type: L8 Version: tnftpd 20080929
192.168.1.45	192.168.1.50	FTP	Request: FEAT
192.168.1.50	192.168.1.45	FTP	Response: 211-Features supported
192.168.1.50	192.168.1.45	FTP	Response: MDTM
192.168.1.45	192.168.1.50	FTP	Request: PWD
192.168.1.50	192.168.1.45	FTP	Response: 257 "/Users/test" is the current directory.
192.168.1.45	192.168.1.50	FTP	Request: QUIT
192.168.1.50	192.168.1.45	FTP	Response: 221-
192.168.1.50	192.168.1.45	FTP	Response: Data traffic for this session was 0 bytes in 0 files.

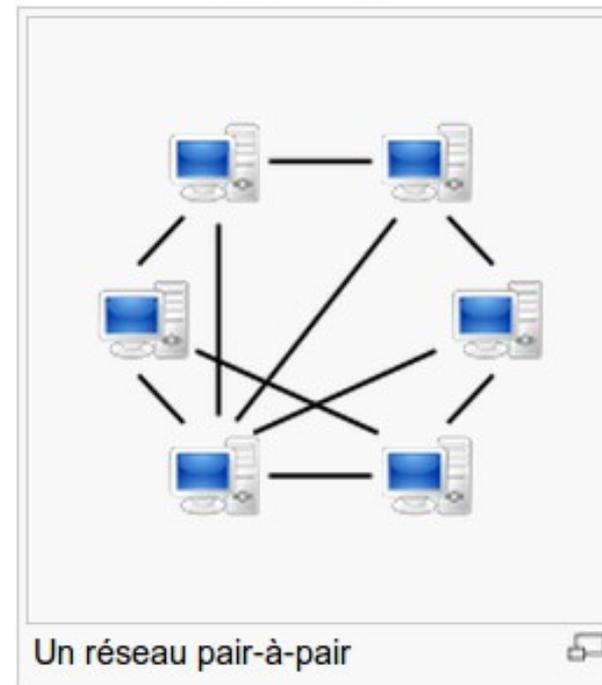
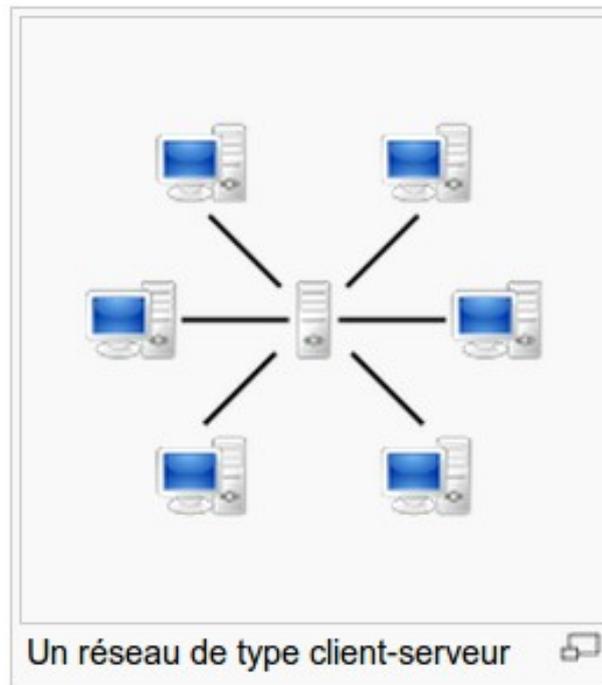
Architecture client - serveur

Quand le client devient serveur : Le peer-to-peer

Le pair-à-pair ou peer-to-peer en anglais, souvent abrégé « P2P » est un modèle de réseau informatique proche du modèle client-serveur mais où chaque client est aussi un serveur.



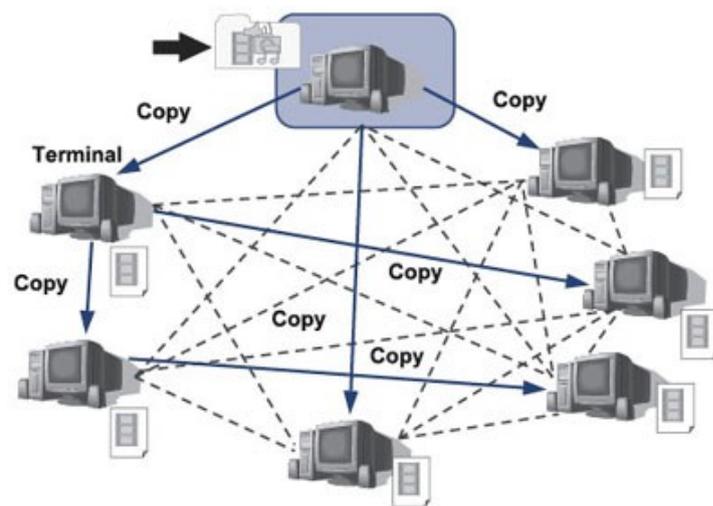
Illustration de réseaux client-serveurs et pair-à-pair.



Le peer-to-peer : avantages

Les systèmes pair-à-pair permettent à plusieurs ordinateurs de communiquer via un réseau, de partager simplement des objets – des fichiers le plus souvent, mais également des flux multimédia continus (streaming), le calcul réparti, un service (comme la téléphonie avec Skype), etc. sur Internet.

Le pair-à-pair a permis une décentralisation des systèmes, auparavant basés sur quelques serveurs exposés à la censure et à l'enregistrement en masse de données privées : il permet à tous les ordinateurs de jouer directement le rôle de client et serveur.



Le peer-to-peer : « le servent »

L'utilisation d'un système pair-à-pair nécessite pour chaque nœud l'utilisation d'un logiciel particulier. Ce logiciel, qui remplit alors à la fois les fonctions de client et de serveur, est parfois appelé servent (de la contraction de « serveur » et de « client »), ou plus communément mais de façon réductrice, « client ».

Exemple de servent :



C'est là l'origine du terme pair (de l'anglais : peer) que l'on trouve dans pair-à-pair : les communications et les échanges se font entre des nœuds qui ont la même responsabilité dans le système.

Architecture client - serveur

Avantages et inconvénients de la communication client-serveur

Avantages	Inconvénients
des ressources centralisées : étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction	risque de surcharge : si trop de clients veulent communiquer avec le serveur au même moment, ce dernier risque de ne pas supporter la charge (alors que les réseaux P2P fonctionnent mieux en ajoutant de nouveaux participants)
une meilleure sécurité : car le nombre de points d'entrée permettant l'accès aux données est moins important	si le serveur n'est plus disponible , plus aucun des clients ne fonctionne (le réseau P2P continue à fonctionner, même si plusieurs participants quittent le réseau).
une administration au niveau serveur : les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés	un coût élevé dû à la technicité du serveur
un réseau évolutif : grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure	un maillon faible : le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui ! Heureusement, le serveur a une grande tolérance aux pannes (notamment grâce au système RAID)