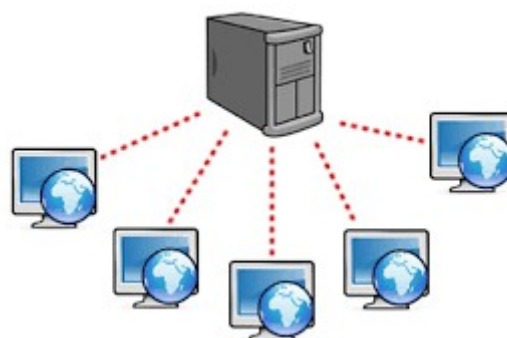


# Architecture client/serveur

## Table des matières

1. Principe du client/serveur.....	2
2. Communication client/serveur.....	3
2.1. Avantages.....	3
2.2. Inconvénients.....	3
3. HTTP (HyperText Transfer Protocol).....	3
3.1. Les méthodes.....	4
3.2. Dialogue HTTP.....	4
4. FTP (File Transfer Protocol).....	4
4.1. Mode actif.....	5
4.2. Mode passif.....	5
5. Exercices d'application.....	6
5.1. Énoncé.....	6
5.2. Correction.....	7

L'architecture client/serveur désigne un mode de communication entre plusieurs composants d'un réseau. Chaque entité est considérée comme un client ou un serveur. Chaque logiciel client peut envoyer des requêtes à un serveur. Un serveur peut être spécialisé en serveur d'applications, de fichiers, de terminaux, ou encore de messagerie électronique.



## 1. Principe du client/serveur

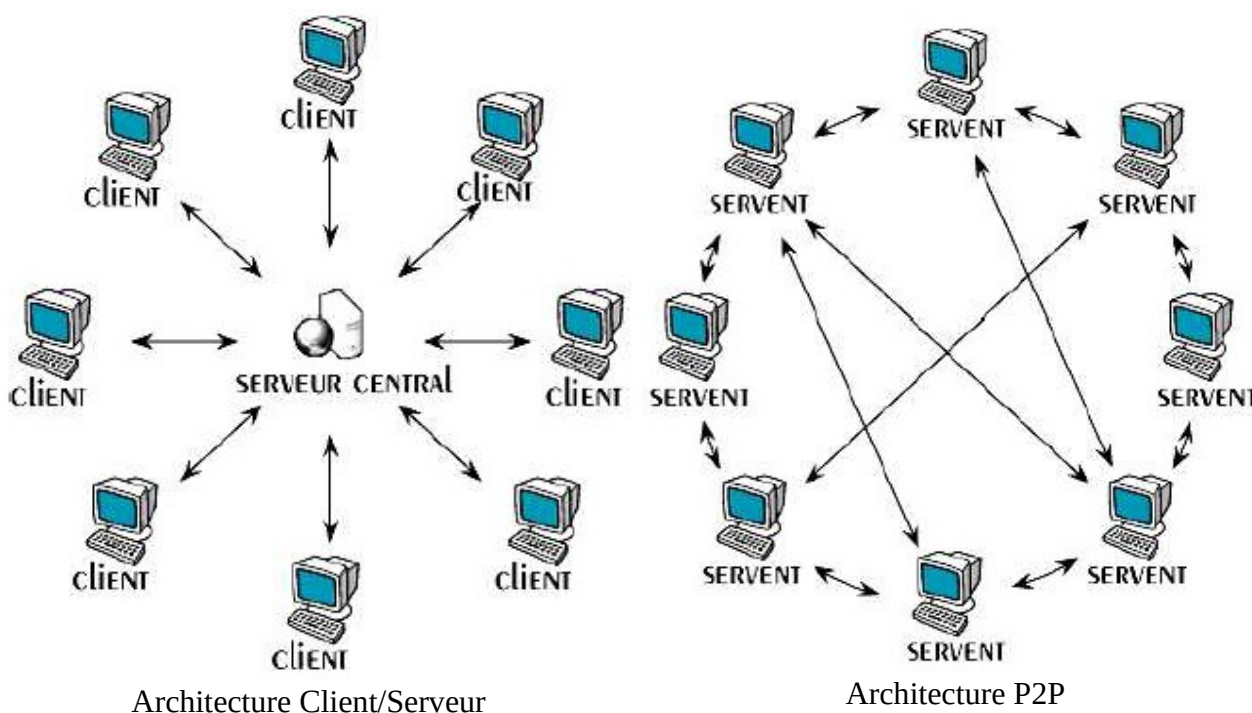
Un **client** : Les caractéristiques d'un client sont les suivantes : il est d'abord actif (ou maître), il envoie des requêtes au serveur, il attend et reçoit les réponses du serveur.

Un **serveur** : Un serveur est initialement passif, il attend, il est à l'écoute, prêt à répondre aux requêtes envoyées par des clients. Dès qu'une requête lui parvient, il la traite et envoie une réponse.

Le dialogue : Le client et le serveur doivent bien sûr utiliser le même protocole de communication. Un serveur est généralement capable de servir plusieurs clients simultanément.

Remarques : Une fois le client traité, le serveur peut en traiter un autre. Il existe des serveurs multiclients comme les serveurs Web/http qui sont capables de traiter plusieurs clients en même temps. Il existe aussi des serveurs « non connectés », dans ce cas il n'y a pas de connexion ou de déconnexion.

Un autre type d'architecture réseau est le pair à pair (peer to peer en anglais ou P2P), dans lequel chaque ordinateur ou logiciel est à la fois client et serveur.



Les systèmes pair-à-pair permettent à plusieurs ordinateurs de communiquer via un réseau, de partager simplement des fichiers le plus souvent, mais également des flux multimédia ou encore un service (comme la téléphonie avec Skype par exemple), ... sur internet.

L'utilisation d'un système pair-à-pair nécessite pour chaque noeud 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 ».

Une des applications la plus connue est le partage de fichiers par le biais de logiciel à la fois client et serveur comme eDonkey, eMule, FastTrack (utilisé par KaZaA) ou BitTorrent...

Les serveurs pair-à-pair fonctionnent dans la quasi-totalité des cas en mode synchrone : le transfert

d'information est limité aux éléments connectés en même temps au réseau. Ils peuvent utiliser le protocole TCP (Transmission Control Protocol ou protocole de contrôle de transmissions) comme couche de transport des données (il fonctionne en duplex, la réception des données est donc confirmée et leur intégrité est assurée).

En revanche, certaines utilisations comme le continu (streaming) nécessitent l'emploi d'un protocole plus léger et plus rapide, comme UDP (User Datagram Protocol ou protocole de datagramme utilisateur), bien que moins fiable, l'UDP est aussi le protocole le plus utilisé pour transmettre des messages entre serveurs dans les systèmes en partie centralisés.

## **2. Communication client/serveur**

Le mode client/serveur n'est pas le modèle de communication parfait, il n'y en a pas ! Connaissant les avantages et les inconvénients par rapport au mode distribué (par exemple pair à pair), il est possible de choisir celui qui convient le mieux.

### **2.1. Avantages**

Toutes les données sont centralisées sur un seul serveur, on a donc « un contrôle de sécurité simplifié ».

Les technologies supportant l'architecture client/serveur sont plus matures que les autres (et plus anciennes).

L'administration se porte au niveau serveur. Toute la complexité/puissance peut être déportée sur le(s) serveur(s), les utilisateurs utilisant simplement un client léger.

Les serveurs étant centralisés, cette architecture est particulièrement adaptée et véloce pour retrouver et comparer de vastes quantités d'information (moteur de recherche sur le web).

### **2.2. Inconvénients**

Si trop de client veulent communiquer sur le serveur en même temps, ce dernier risque de ne pas supporter la charge (alors que les réseaux pair à pair fonctionnent mieux en ajoutant de nouveaux participants).

Si le serveur n'est plus disponible, plus aucun des clients ne fonctionne (le réseau pair à pair continue à fonctionner, même si plusieurs participants quittent le réseau).

Les coûts de mise en place et de maintenance sont élevés.

En aucun cas les clients ne peuvent communiquer entre eux, entraînant une asymétrie de l'information au profit des serveurs.

## **3. HTTP (HyperText Transfer Protocol)**

La consultation des 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 (HyperText Transfer Protocol) qui est utilisé.

Les navigateurs sont les clients (Firefox, Internet Explorer, ...). Ces clients se connectent à des serveurs http tels qu'Apache http Server ou IIS (Internet Information Services).

Remarque : HTTPS (avec S pour Secured ou Sécurisé) est la variante de http sécurisée par l'usage des protocoles SSL (Secure Sockets Layer) ou nouvellement nommé TLS (Transport Layer Security) qui est un protocole de sécurisation des échanges sur internet.

### 3.1. Les méthodes

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. En général, l'action concerne une ressource identifiée par l'URL qui suit le nom de la méthode.

Les méthodes les plus utilisées sont GET et POST.

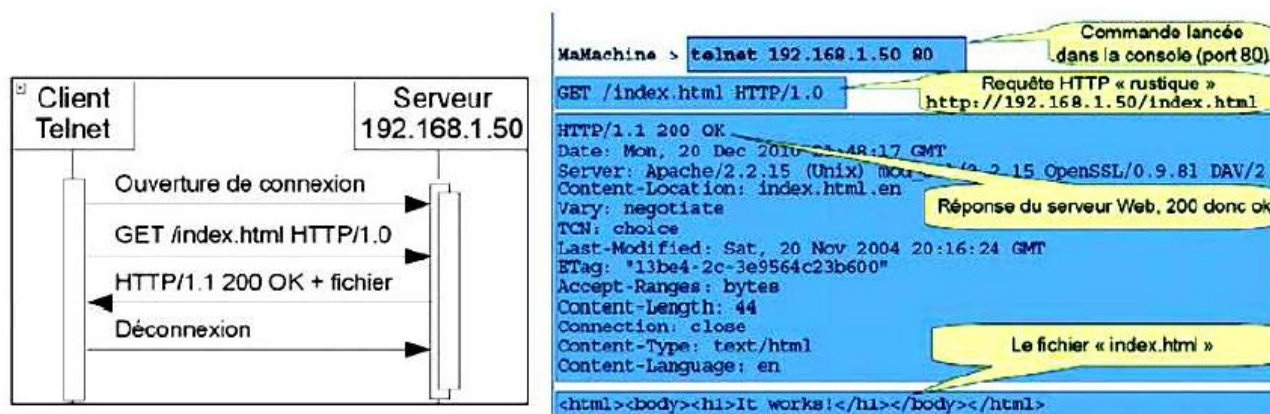
**GET** : c'est la méthode la plus courante pour demander une ressource. Une requête GET est sans effet sur la ressource.

**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.

Les autres méthodes sont : HEAD, OPTIONS, CONNECT, TRACE, PUT, DELETE. Certains serveurs autorisent des méthodes supplémentaires permettant la gestion des ressources du serveur (par exemple WebDAV ou CALDAV).

### 3.2. Dialogue HTTP

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



Lien vers les codes HTTP : <http://www.codeshttp.com/>

La réponse du serveur « Content-Lengh: 44 » représente la longueur des données utiles en nombre d'octets (et donc de caractères en UTF-8).

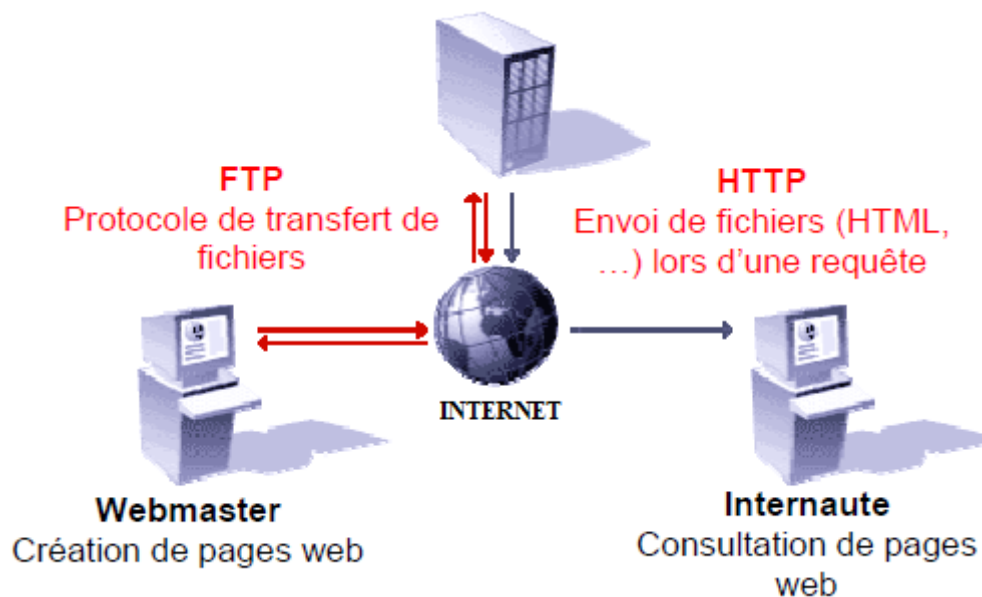
## 4. FTP (File Transfer Protocol)

Le protocole de transfert de fichiers, ou FTP (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.

En pratique, le serveur est un ordinateur sur lequel fonctionne un logiciel lui-même appelé serveur

FTP. Pour accéder à un serveur FTP, on utilise un logiciel client FTP (possédant une interface graphique comme FileZilla par exemple ou en ligne de commande).

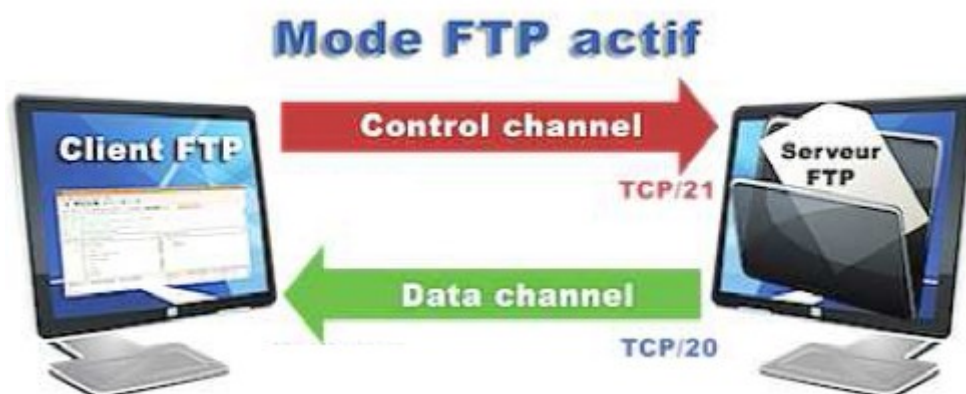


#### Rôle de FTP dans la publication WEB

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

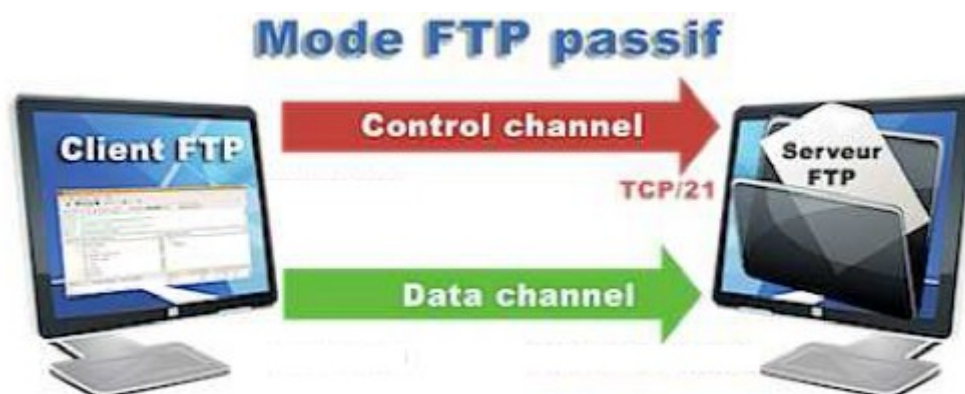
### 4.1. Mode actif

Le client devient serveur pour recevoir un fichier, cela permet de télécharger le « serveur » FTP. Ce mode ne fonctionne pas très bien avec les « BOX » internet.



### 4.2. Mode passif

Le serveur FTP reste serveur même pour les transferts de fichiers.



## 5. Exercices d'application

### 5.1. Énoncé

Question 1 : Surligner les bonnes réponses.

- |  |      |      |
|--|------|------|
| Un serveur web écoute sur le port 80...  | Vrai | Faux |
| Pour déposer un fichier sur un serveur, Quel protocole utiliser ?  | FTP  | HTTP |
| Pour obtenir la liste de fichiers disponibles sur un serveur, quel protocole utiliser ?  | FTP  | HTTP |
| Vous vous connectez sur un compte FTP, votre FAI (Fournisseur d'Accès Internet) peut-il avoir connaissance de votre mot de passe ? | Oui  | Non  |

Question 2 : Capture de trame avec wireshark (un utilitaire permettant de capturer des trames réseau).

Un échange entre un client et un serveur a été intercepté :

source	destination	protocole	information
192.168.1.50	192.168.1.45	FTP	Response: 220 192.168.1.50 FTP server (tnftpd 20080929) 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.

Question 2.1 : Quelle est l'adresse IP du serveur ?

Question 2.2 : Quelle est l'adresse IP du client ?

Question 2.3 : Quel utilisateur se connecte au serveur ?

Question 2.4 : Quel est le mot de passe de l'utilisateur connecté au serveur ?

Question 2.5 : FTP est-il un protocole sécurisé ?

Question 3 : échange HTTP

Dans l'échange HTTP suivant :

```
$> telnet www.llfnet.fr 80
POST /cgi-bin/montest.pl HTTP/1.1
Host: 192.168.1.1
Content-Length: 33
envoiest=testdenvoiedeparametre
HTTP/1.1 200 OK
Date: Mon, 03 Dec 2012 22:45:12 GMT
Server: Apache/2.2.15 (Unix) mod_ssl/2.2.15 OpenSSL/0.9.8l DAV/2
Content-Length: 85
Connection: close
Content-Type: text/html; charset=ISO-8859-1
<HTML>
<BODY>
contenu HTML donnée = envoiest=testdenvoiedeparametre
</BODY>
</HTML>
```

Question 3.1 : Lister l'adresse IP du serveur, la version et le type de serveur web ?

Question 3.2 : Surligner les données transférées par la requête POST.

Question 3.3 : Quel est le type des données envoyées par le serveur web ?

## 5.2. Correction

Question 1 :

Un serveur web écoute sur le port 80...

Vrai

Faux

Pour déposer un fichier sur un serveur, Quel protocole utiliser ?

FTP

HTTP

Pour obtenir la liste de fichiers disponibles sur un serveur, quel protocole utiliser ?

FTP

HTTP

Vous vous connectez sur un compte FTP, votre FAI (Fournisseur d'Accès Internet) peut-il avoir connaissance de votre mot de passe ?

Oui

Non

Question 2.1 : 192.168.1.50

Question 2.2 : 192.168.1.45

Question 2.3 : Utilisateur connecté au serveur = test

Question 2.4 : Mot de passe = Chocolat2010

Question 2.5 : Non

Question 3.1 :

- Adresse IP du serveur : 192.168.1.1
- Nom du serveur : www.llfnet.fr
- Le serveur est de type Apache version 2.2.15

Question 3.2 :

```
HTTP/1.1 200 OK
Date: Mon, 03 Dec 2012 22:45:12 GMT
Server: Apache/2.2.15 (Unix) mod_ssl/2.2.15 OpenSSL/0.9.8l DAV/2
Content-Length: 85
Connection: close
Content-Type: text/html; charset=ISO-8859-1
<HTML>
<BODY>
contenu HTML donnée = envoiepost=testdenvoiedeparametre
</BODY>
</HTML>
```

Question 3.3 : Le champ Content-Type: text/html, nous renseigne sur le contenu. Les données sont donc de type texte en langage HTML.