

S1-TP2- RE1 : Service de transfert de fichier

Le **File Transfer Protocol** (*protocole de transfert de fichiers*), ou **FTP**, est un protocole de communication dédié à l'échange informatique de fichiers sur un réseau TCP/IP. Il permet, depuis un ordinateur, de copier des fichiers depuis ou vers un autre ordinateur du réseau, d'administrer un site web, ou encore de supprimer ou modifier des fichiers sur cet ordinateur. La variante sécurisée de FTP avec les protocoles SSL ou TLS s'appelle FTPS.

FTP obéit à un modèle client-serveur, c'est-à-dire qu'une des deux parties, le *client*, envoie des requêtes auxquelles réagit l'autre, appelé *serveur*. En pratique, le serveur est un ordinateur sur lequel fonctionne un logiciel lui-même appelé serveur FTP, qui rend publique une arborescence de fichiers similaire à un système de fichiers Unix. **Pour accéder à un serveur FTP, on utilise un logiciel client FTP (possédant une interface graphique ou en ligne de commande).**

Le protocole, qui appartient à la couche session du modèle OSI et à la couche application du modèle ARPA, utilise une connexion TCP. Il peut s'utiliser de deux façons différentes :

1. **Mode actif**: c'est le client FTP qui détermine le port de connexion à utiliser pour permettre le transfert des données. Ainsi, pour que l'échange des données puisse se faire, le serveur FTP initialisera la connexion de son port de données (port 20) vers le port spécifié par le client. Le client devra alors configurer son pare-feu pour autoriser les nouvelles connexions entrantes afin que l'échange des données se fasse. De plus il peut s'avérer problématique pour les utilisateurs essayant d'accéder à des serveurs FTP lorsqu'ils sont derrière une passerelle NAT. Vu la façon dont fonctionne le NAT, le serveur FTP initie la connexion de données en se connectant à l'adresse externe de la passerelle NAT sur le port choisi. Certaines passerelles NAT n'ayant pas de correspondance pour le paquet reçu dans la table d'état le paquet sera ignoré et ne sera pas délivré au client.
2. **Mode passif**, le serveur FTP détermine lui-même le port de connexion à utiliser pour permettre le transfert des données (data connexion) et le communique au client. Dans le cas de l'existence d'un pare-feu devant le serveur FTP celui-ci devra être configuré pour autoriser la connexion de données. L'avantage de ce mode, est que le serveur FTP n'initialise aucune connexion. Ce mode fonctionne sans problèmes avec une passerelle NAT.

Deux ports sont standardisés (*well known ports*) pour les connexions FTP : le port 21 pour les commandes et le port 20 pour les données.

1. Utilisation

Pour accéder à un serveur FTP, on utilise un client ftp, en ligne de commande ou avec une interface graphique.

Les utilisateurs de [GNU-Linux](#) ou d'un [Unix](#) peuvent consulter une documentation (la plupart du temps installée par défaut), en tapant « man ftp »

La plupart des navigateurs récents autorisent les connexions FTP en utilisant une [URL](#) de type :

`ftp://nom_d'utilisateur:mot_de_passe@nom_ou_adresse_du_serveur:port_ftp`

2. Logiciels de FTP

La plupart des logiciels possèdent désormais une interface graphique permettant de mettre en place un serveur FTP ou de se connecter à celui-ci pour y copier des données (client FTP). Certains logiciels tels que [CuteFTP](#) (Windows) sont payants ; d'autres, tels que [FileZilla](#) et [KamzyFTP](#) pour (Windows), [Cyberduck](#) pour (MacOSX) ou [gftp](#) pour (Linux), tout aussi pratiques et efficaces, sont gratuits et [libres](#). On peut également citer [FTP explorer](#) (Shareware pour Windows), aussi pratique à utiliser que l'explorateur de fichiers, ou encore FireFtp, plugin de Firefox. Pour des utilisations plus poussées (commandes spéciales etc.) un logiciel tels que FlashFXP (payant) est aussi utilisé.

3. Le protocole

Le protocole utilise deux types de connexions TCP :

- Une connexion de *contrôle* initialisée par le client, vers le serveur (port 21 en général), pour transmettre les commandes de fichiers (transfert, suppression de fichiers, renommage, liste des fichiers...).
- Une connexion de *données* initialisée par le client ou le serveur pour transférer les données requises (contenu des fichiers, liste de fichiers).

3.1 Connexion de contrôle

Le client reçoit une ou plusieurs réponses du serveur. Chaque réponse est précédée d'un code décimal permettant au client FTP de traiter la réponse qui peut comporter une ou plusieurs lignes de texte.

Pour l'exemple précédent, si le serveur trouve le fichier demandé, il envoie au client :

150 File status okay; about to open data connection.

Selon ce que le client et le serveur ont convenu, l'un des deux écoute sur le port TCP convenu, et l'autre s'y connecte pour établir la connexion de données. Puis le serveur envoie au client le contenu du fichier demandé, ferme la connexion de données, et envoie la réponse suivante sur la connexion de contrôle :

226 Closing data connection.

3.2 Connexion de données

La connexion de données est établie pour la durée de transmission de données (contenu de fichiers, ou liste de fichiers). En général, elle est établie pour le transfert de données d'une seule commande, à moins qu'un autre mode de transmission soit sélectionné et supporté par le serveur.

La commande PASV indique au serveur qu'il doit attendre passivement la connexion en écoutant un port TCP (en général, le port 20). Le port écouté par le serveur est indiqué dans la réponse :

227 Entering Passive Mode (*b1,b2,b3,b4,p1,p2*).

Où h1 à h4 sont 4 nombres entiers entre 0 et 255 représentant l'adresse IP du serveur, et p1 et p2 représentent le port TCP où le serveur attend la connexion, sous la forme de deux entiers entre 0 et 255 ($\text{port_TCP} = p1 * 256 + p2$).

Dans le cas contraire où le client attend la connexion sur un port TCP, il indique sous la même forme le port écouté en envoyant la commande PORT :

PORT h1,h2,h3,h4,p1,p2

Si tout se passe bien, le serveur répond :

200 Command okay.

3.3 Mode de transfert

Lors du transfert de fichier sur la connexion de données, 2 modes peuvent être utilisés :

- Le mode Binaire : le fichier est transmis tel quel.
- Le mode ASCII : uniquement destiné aux fichiers texte. Le fichier est examiné et des transformations apportées pour conserver un format correct. Par exemple, la fin de ligne est représentée par le caractère <LF> sur un système UNIX, et par la paire <CR><LF> sous Windows.

4. TRAVAIL PRATIQUE

4.1 Afficher le man de la commande ftp ou (help sous le prompt ftp) pour réaliser un sommaire des principales commandes (ascii, binary, cd, close, delete, get, put, mget, mput, ls, lcd, !, mkdir, open, quit, rmdir, passive...). Ces commandes sont-elles identiques à celles vues en Tp de SYS ?

4.2 Réaliser une connexion FTP en mode commande

- a) entre un client ftp (linux) et un serveur Unix (Onyx)
- b) entre un client (Windows) et un serveur Unix (Onyx)

pour transférer et récupérer le fichier de votre choix (ex : germinal.txt), vous noterez la durée du transfert et calculerez le débit associé. Que déduire par rapport au débit théorique Ethernet 100Mb/s

4.3 Réaliser en mode commande une connexion ftp anonyme vers le serveur ftp.cict.fr (login : anonymous, password : guest).

- a) le serveur est-il directement accessible ? pourquoi ?
- b) quelle est l'utilité de la commande ftp « pasv », et que déduire du mode de connexion TCP ?
- c) après vous être déplacé dans l'arborescence de serveur, transférer le fichier de votre choix.
- d) est-il possible de déposer un fichier sur ftp.cict.fr ?

4.4 Tout en réalisant un transfert de fichier (étape par étape), observer (netstat) les connexions réseaux générées côté client et côté serveur avant la phase de connexion, après, avant le transfert de données de fichier, après, avant le dépôt d'un fichier, après. Notez les résultats sous la forme d'un tableau (@IP source, destination, n° de ports source, destination) et vos conclusions.

4.5 Réaliser une connexion ftp avec l'utilitaire de transfert (Filezilla), entre un client Windows et le serveur Onyx