

Le service de transfert de fichiers (FTP)

1) Présentation du protocole FTP

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

Parmi les applications les plus fréquentes du FTP :

- Download ou Téléchargement, lorsque l'on récupère des fichiers depuis le serveur.
- Upload ou Téléversement, lorsque l'on dépose des fichiers sur le serveur
- transferts de fichiers entre deux serveurs distants. Bien que cette possibilité soit peu intéressante pour l'internaute moyen, ça reste une fonctionnalité importante pour les administrateurs de sites distants

2) Le modèle FTP

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*

FTP utilise deux canaux de transmissions pour transférer un fichier :

- Une **connexion de contrôle** est utilisée pour acheminer les commandes (ou requêtes) du client vers le serveur et les réponses (ou résultats) du serveur vers le client.
- Une **connexion de transfert de données** qui est créée à chaque fois qu'un fichier est transféré entre le client et le serveur.

Pour la connexion de contrôle, le numéro de port utilisé par le serveur ftp est 21. C'est un port réservé (*well-known port*) qui est décrit dans le fichier */etc/services*. Le client, quand à lui obtient un numéro de port dynamiquement.

Pour la connexion de transfert de données, le numéro de port utilisé par le serveur ftp est 20. Le client, quand à lui, obtient un numéro de port dynamiquement.

De nombreux choix de représentation des données sont fournis par la spécification du protocole FTP. Les seuls qui sont encore utilisés aujourd'hui sont les types ASCII et binaires (BINARY ou IMAGE).

On utilise le mode ASCII lorsque les fichiers échangés ne contiennent que des caractères éditables. Dans ce cas, il y a prise en compte des différences entre les machines (traduction des séquences de retour à la ligne notamment). Les fichiers exécutables sont en mode binaire.

En général, c'est l'utilisateur qui doit choisir le mode de transfert (par défaut c'est en général le mode ASCII qui est utilisé).

3) FTP actif -FTP passif

FTP peut s'utiliser de deux façons différentes : en mode actif ou en mode passif

En 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.

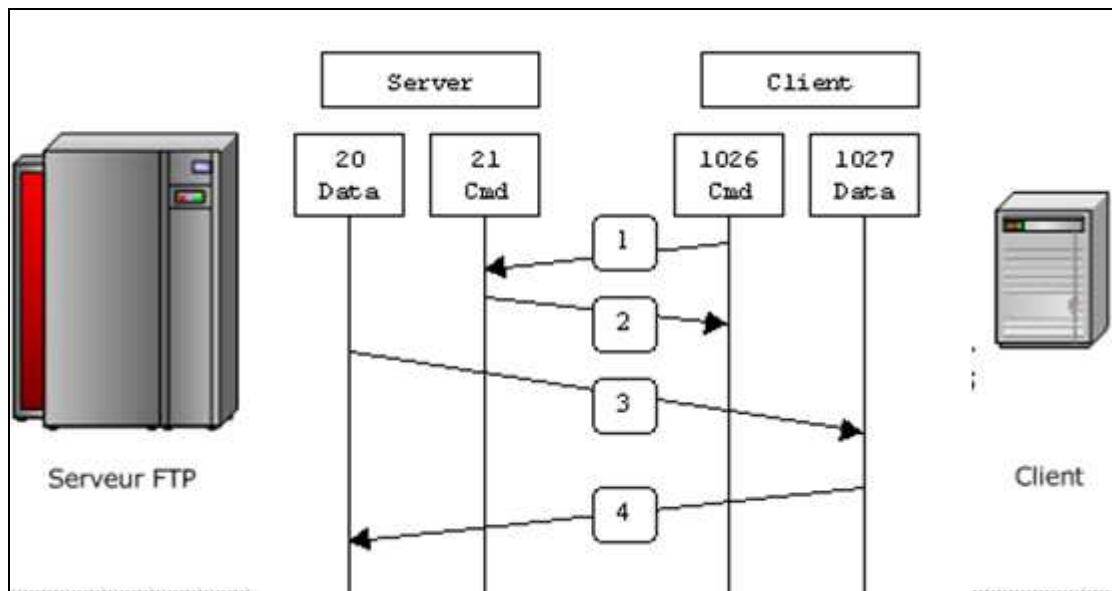


Diagramme FTP en mode actif

Les serveurs FTP qui acceptent le mode passif n'utilisent pas forcément le port 20 pour le transfert de données. En effet, le mode passif permet aux hôtes d'effectuer le transfert de

données sur un ou plusieurs autres couples de ports (port d'écoute > 1024 défini par le serveur et autre port > 1024 pour le client).

Le mécanisme habituel est le suivant :

- début de l'échange à l'initiative du client vers le port 21 du serveur,
- demande de passage en mode passif de la part du client, en vue du transfert de données,
- acceptation par le serveur et proposition d'un port d'écoute > 1024 (ou refus du mode passif),
- poursuite des échanges : commandes par le port 21 et données par un port >1024 (ou port 20 si refus du mode passif).

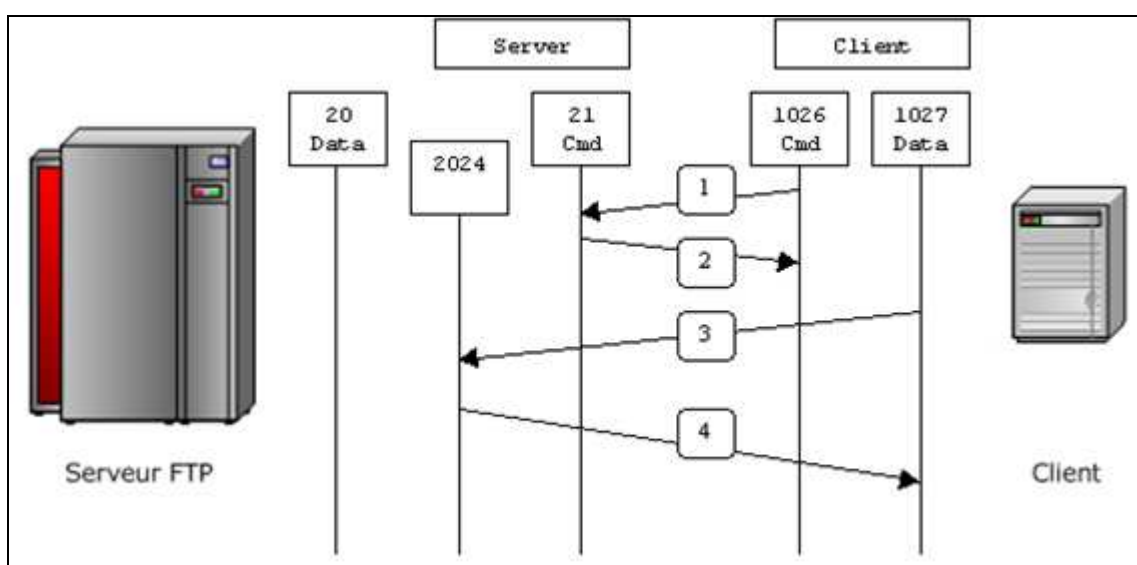


Diagramme FTP en mode passif

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'initie aucune connexion. Dans le cas des clients FTP sur un réseau local, ce mode est beaucoup plus sécurisé que le FTP en mode actif, car le pare-feu ne devra laisser passer que les flux sortant vers internet pour permettre aux clients d'échanger des données avec le serveur.

4) Logiciels de transfert de fichiers

Les logiciels clients sont nombreux, la plupart sont en ligne de commande (par exemple la commande ftp, implémentée en standard sous de nombreux systèmes d'exploitation), mais certains permettent de manipuler des fichiers avec une interface graphique (Filezilla, gftp, etc.).

Il est par ailleurs possible de se connecter à un serveur ftp depuis un navigateur web. Dans ce cas, l'URL spécifié sera de type <ftp://nomDuServeur/public> pour une connexion anonyme.

Il existe également plusieurs logiciels serveurs ftp comme VsFTPd, ProFTPd, Pure-FTPd. Ce sont des logiciels libres qui tournent sous des plateformes dérivées d'Unix.

5) sessions FTP

Une session FTP commence quand le client FTP démarre, un serveur FTP doit être spécifié et la connexion initialisée.

Dès que l'utilisateur s'est identifié et son mot de passe accepté par le serveur, l'utilisateur essaye généralement de localiser les fichiers qui l'intéressent et les récupère depuis le serveur sur le poste local. Dès que tous les transferts sont terminés, l'utilisateur termine sa connexion.

Le circuit de contrôle FTP reste actif pendant toute la session, en revanche, le circuit de transfert n'existe lui que le temps de l'envoi des données. Cette séparation en 2 circuits garantie qu'un canal restera ouvert pour transmettre les messages d'erreur si le circuit de transfert est perdu. Une autre raison de ce partitionnement est l'utilisation du transfert de fichiers en mode "stream", plutôt que d'envoyer une balise de fin de fichier, le transfert est considéré comme complet quand le circuit est fermé. Le résultat est donc que chaque fois qu'un fichier a été envoyé, le circuit de transfert se ferme et le circuit de contrôle en ouvre un autre. Si une erreur provoque la fermeture du circuit de transfert, le module de contrôle considérera le fichier comme envoyé et un fragment du fichier restera dans le système de fichiers du destinataire.

La commande pour initier une session FTP est généralement la suivante:

ftp nom_serveur où `nom_serveur` représente le nom (ou l'adresse IP) du serveur FTP auquel on désire se connecter.

6) Commandes FTP utilisable sous Linux :

- `help` (ou `?`) : Affiche l'ensemble des commandes supportées par le serveur FTP
- `status` : Permet de connaître certains paramètres de la machine cliente

- `binary` : fait basculer du mode ASCII (envoi de documents textes) au mode binary (envoi de fichiers en mode binaire, c'est-à-dire pour les fichiers non texte, comme des images ou des programmes)
- `ascii` : bascule du mode binary au mode ascii. Ce mode est le mode par défaut.
- `ls` : identique à la commande UNIX, mais exécutée sur le serveur FTP.
- `pwd` : Affiche le nom répertoire courant sur le serveur FTP.
- `cd` : identique à la commande UNIX mais exécutée sur le serveur FTP.
- `lcd` : identique à la commande `cd` mais exécutée sur la machine cliente.
- `get nom1` : permet de récupérer le fichier `nom1` présent sur le serveur FTP.
- `put nom1` : permet d'envoyer le fichier local `nom1` sur le serveur FTP.
- `user` : lance une nouvelle session pour un nouvel utilisateur
- `mget` : identique à GET mais permet de récupérer plusieurs fichiers dont le nom est donné en paramètre.
- `quit` : déconnecte le logiciel client du serveur FTP et le met en état inactif
- `open` : ferme la session en cours et ouvre une nouvelle session sur un autre serveur FTP
- `close` : ferme la session en cours, en laissant le logiciel FTP client actif
- `!commande` : Quand `!` précède une commande, la commande est exécutée localement.
- `mkdir nom_rep` : crée le répertoire `nom_rep` dans le répertoire courant du serveur.
- `rmdir nom_rep` : supprime le répertoire `nom_rep` (s'il est vide) du répertoire courant du serveur.
- `Ren nom1 nom2` : permet de renommer le fichier `nom1` en `nom2` sur le serveur.
- `bye` : se déconnecte du serveur FTP et le met l'application FTP client en état inactif
- `mput` : identique à PUT mais permet d'envoyer plusieurs fichiers dont le nom est donné en paramètre.

7) Installation et configuration d'un serveur FTP

Un exemple de mise en place du service FTP s'appuyant sur *proftpd* et *gftp* sera vu en détails dans la séance du TP. Cependant, ci-dessous sont explicitées quelques *directives* du *fichier de configuration proftpd.conf* :

```
ServerName "nom_de_mon_serveur_FTP"
```

description = Indique le nom du serveur qui s'affichera sur les clients

```
AccessGrantMsg "message"
```

description = Message de bienvenue. Le message peut contenir des jokers comme %u (ici le nom de l'utilisateur)

```
<Limit ...>...</Limit>
```

description = Autorise ou refuse l'utilisation de certaines commandes FTP.

Par exemple la section suivante n'autorise la commande MKDIR qu'aux utilisateurs user1 et user2 :

```
<Limit MKDIR>
```

```
    Allow user1 user2
```

```
    Deny All
```

```
</Limit>
```

```
ServerType "type"
```

description = Détermine la manière dont le serveur reçoit les connexions réseau. Le type est "standalone" indique que le serveur est autonome, et que le service tourne en permanence. Le type est "inet", indique que le serveur doit être lancé par inetd

```
MaxInstances 30
```

description = Limite le nombre de processus simultanés autorisés

```
AllowOverwrite on
```

description = autorise un utilisateur à écraser un fichier qui lui appartient.

AllowUser 'liste-d'utilisateurs'

description = à placer dans un contexte <Limit ...>...</Limit> définit qui est autorisé à exécuter la commande du bloc "Limit" courant.

DenyUser 'liste-d'utilisateurs'

description = à placer dans un contexte <Limit ...>...</Limit> définit qui n'est pas autorisé à exécuter la commande du bloc Limit courant.

AllowStoreRestart

description = autorise les clients à reprendre les uploads vers le serveur.

DefaultChdir /var/ftp

description = Indique le répertoire par défaut du serveur.

commentaires = Les utilisateurs se trouvent placés dans ce répertoire lors de la connexion.

DefaultRoot /var/ftp

description = déclare ce répertoire comme la racine du système de fichiers.

UserRatio

description = permet la gestion des ratios.

UserRatio "user" 2 10 5 4096 indique que l'utilisateur "user" a le droit de récupérer 2 fichiers sur le serveur à chaque fois qu'il en déposera un.

On lui octroie pour commencer un crédit de 10 fichiers.

Par ailleurs, pour 1 octet déposé, il pourra recevoir 5 octets et il détient un crédit de 4ko. À la place d'un nom on peut aussi utiliser * qui définit des ratios par défaut.

<Directory "répertoire"> ... </Directory>

description = Cette section indique les droits sur le répertoire et sur tout ce qu'il contient. Par exemple :

```
<Directory /var/ftp/docs>
  <Limit ALL>
    Deny ALL
  </Limit>
  HideNoAccess on
</Directory>
```

Ici, nous interdisons toute opération sur le répertoire docs grâce à la section <Limit ALL>.

HideNoAccess

description = Cache tous les éléments inaccessibles aux utilisateurs.

```
<Anonymous "répertoire">...</Anonymous>
```

description = Configure l'accès anonyme

Exemple :

```
<Anonymous /home/ftp>
  # Après la connexion anonyme, passe sous l'utilisateur/groupe ftp.
  User ftp
  Group ftp

  # Fait correspondre le login "anonymous" au compte unix "ftp"
  UserAlias anonymous ftp

  # Interdit l'écriture partout
  <Directory *>
    <Limit WRITE>
      DenyAll
    </Limit>
  </Directory>
```