

© Copyright 2008-2010 tv <thierry.vaira@orange.fr>

Permission is granted to copy, distribute and/or modify this document under the terms of the **GNU Free Documentation License**, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover.

You can obtain a copy of the GNU General Public License : write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Permission est accordée de copier, distribuer et/ou modifier ce document selon les termes de la **Licence de Documentation Libre GNU** (GNU Free Documentation License), version 1.1 ou toute version ultérieure publiée par la Free Software Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture, et sans Texte de Quatrième de Couverture.

Vous pouvez obtenir une copie de la GNU General Public License : écrire à la Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Table des matières

Séquence 1 - Client FTP.....	2
Séquence 2 - Serveur FTP.....	7
Annexe: les modèles de référence.....	8
Annexe: le protocole FTP (File Transfer Protocol).....	9
Annexe : clients ftp.....	14

En ligne

protocoles internet (http, smtp, pop, imap, ftp, mime, nntp, irc...) : www.salemioche.com/

protocoles internet (ip, arp, tcp, udp, ...) : www.frameip.com/

rfc en français : <http://abcdrfc.free.fr/> et <http://www.eisti.fr/res/norme/rfc959/959tm.htm>

Séquence 1 - Client FTP

Préparation de la manipulation : les besoins

a . Le client de base : outil capable d'ouvrir une connexion TCP sur un port distant et d'échanger des informations en mode texte

Réponse : **telnet** ou **netcat**

```
ping adresse_ip_pc
```

b . Identifier le port d'écoute distant à partir de la liste officielle. Dans quel fichier trouve-t-on cette information ?

Réponse : **ftp -> 21 (dans le fichier /etc/services)**

c . Choisir un serveur FTP pour l'échange.

Réponse: **ftp.free.fr** ou **adresse IP locale (fournie)**

d. La commande de connexion :

Manipulation

1 . Envoyer les requêtes FTP ci-dessous. **Lister** et **commenter** les codes de retour du serveur pendant l'échange (utiliser les informations fournies sur le protocole FTP).

Réponses:

Réponse du serveur : 220 Welcome to ProXad FTP server

PWD

Réponse du serveur :

USER anonymous

Réponse du serveur :

PASS

Réponse du serveur :

PWD

Réponse du serveur :

TYPE I

Réponse du serveur :

LIST -aL

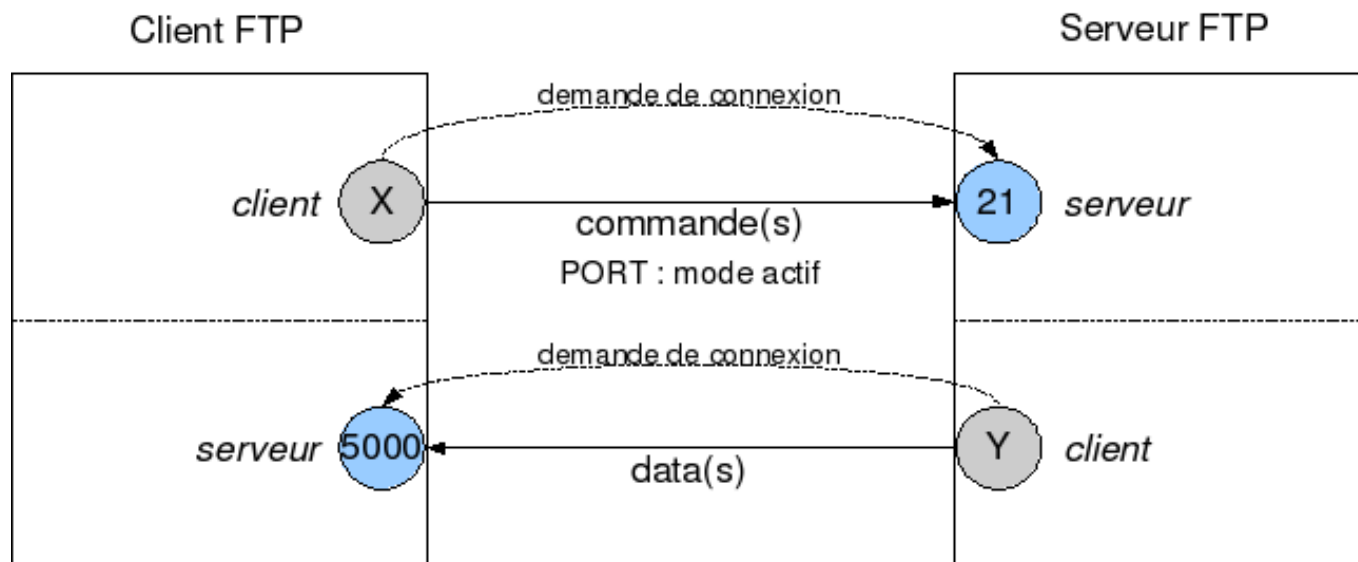
Réponse du serveur :

QUIT

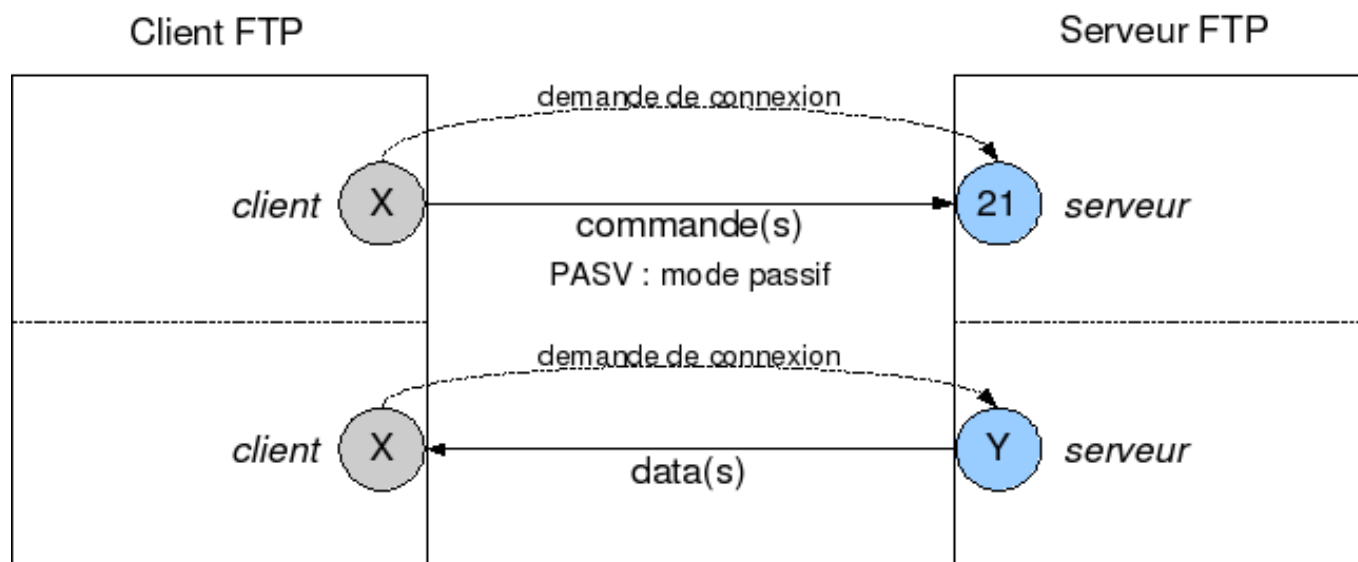
Réponse du serveur : 221 Goodbye.

La commande **LIST -aL** (ainsi que les autres commandes comme **STOR** ou **RETR**) ne fonctionne pas car les données ne sont pas envoyées par le serveur sur le canal ouvert (port 21). Le canal ouvert sur le port 21 du serveur est destiné à l'échanges de commandes du protocole FTP. Les données échangées entre le client et le serveur doivent se faire sur un autre canal. L'échange de données fonctionnant suivant le modèle client/serveur (le client initie la connexion), il existe donc deux possibilités :

- le **mode actif** : le client FTP (en utilisant la commande **PORT**) détermine le port d'écoute et joue le rôle de serveur pour ce canal



- le **mode passif** : le client FTP (en utilisant la commande **PASV**) choisit le mode passif et le serveur FTP détermine le port d'écoute et joue le rôle de serveur pour ce canal



2 . On va tout d'abord utiliser la commande **PASV** pour passer en mode passif :

La réponse du serveur à la commande "PASV" est une chaîne de caractère du type :
"xxx (IP1 , IP2 , IP3 , IP4 , PORT1 , PORT2) "

Il faut extraire les informations suivantes (attention les serveurs FTP gèrent un *timeout* !):

- reconstituer l'adresse IP -> IP1.IP2.IP3.IP4
- déterminer le numéro de port -> PORT1 x 256 + PORT2

Assurer le bon fonctionnement de la commande suivante "LIST -aL" ...

```
$ telnet 192.168.52.7 21
Trying 192.168.52.7...
Connected to 192.168.52.7
(192.168.52.7).
Escape character is '^]'.
220 mandrivalinux.local FTP server
(Version wu-2.6.2(1) Sun Aug 3 13:51:18
EDT 2008) ready.
USER anonymous
331 Guest login ok, send your complete
e-mail address as password.
PASS tv@free.fr
230 Guest login ok, access restrictions
apply.
PASV
227 Entering Passive Mode
(192,168,52,7,205,179)
```

```
LIST -al
150 Opening ASCII mode data connection
for directory listing.
226 Transfer complete.

QUIT
221-You have transferred 0 bytes in 0
files.
221-Total traffic for this session was
433 bytes in 0 transfers.
221-Thank you for using the FTP service
on mandrivalinux.local.
221 Goodbye.
Connection closed by foreign host.
```

```
// calcul : 205*256 + 179 = 52659
$ telnet 192.168.52.7 52659
Trying 192.168.52.7...
Connected to 192.168.52.7 (192.168.52.7).
Escape character is '^]'.
total 48
drwxr-xr-x 6 root root 4096 Jun  1
16:12 .
drwxr-xr-x 6 root root 4096 Jun  1
16:12 ..
d--x--x--x 2 root root 4096 Jun  1 16:12
bin
d--x--x--x 2 root root 4096 Jun  1 16:12
etc
drwxr-xr-x 2 root root 4096 Jun  1 16:12
lib
drwxr-xr-x 2 root 79 4096 Jun  1 16:15
pub
Connection closed by foreign host.
```

- 3 . Réaliser une capture de l'échange précédent et déterminer le protocole de transport utilisé par FTP.
- 4 . On va maintenant utiliser la commande **PORT** pour passer en mode actif :

La commande "PORT" envoyée au serveur est une chaîne de caractère du type :

"PORT IP1 , IP2 , IP3 , IP4 , PORT1 , PORT2 "

Il faut déterminer les informations suivantes pour réaliser l'échange de données :

- fournir l'**adresse IP** où se fera l'échange -> IP1.IP2.IP3.IP4
- choisir un **numéro de port** libre où se fera l'échange -> PORT1 x 256 + PORT2

Le client FTP a donc la responsabilité d'ouvrir un port (libre) et d'attendre la demande de connexion. Il faut donc utiliser un outil capable de fonctionner en serveur TCP. Le choix se porte sur l'utilitaire **netcat**. et le numéro de port 5000 qui est réservé

```
$ netcat -h
[v1.10]
connect to somewhere:  nc [-options] hostname port[s] [ports] ...
listen for inbound:   nc -l -p port [-options] [hostname] [port]
options:
    -l                    listen mode, for inbound connects
    ...
    -p port              local port number
    -s addr              local source address

$ cat /etc/services

$ netstat -l -t -v -n
```

```
$ telnet 192.168.52.7 21
Trying 192.168.52.7...
Connected to 192.168.52.7
(192.168.52.7).
Escape character is '^]'.
220 mandrivalinux.local FTP server
(Version wu-2.6.2(1) Sun Aug 3 13:51:18
EDT 2008) ready.
USER anonymous
331 Guest login ok, send your complete
e-mail address as password.
PASS tv@free.fr
230 Guest login ok, access restrictions
apply.
```

PORT 192,168,52,2,19,136

```
200 PORT command successful.
```

```
LIST -al
150 Opening BINARY mode data connection
for directory listing.
226 Transfer complete.
```

```
QUIT
221-You have transferred 0 bytes in 0
files.
221-Total traffic for this session was
593 bytes in 0 transfers.
221-Thank you for using the FTP service
on mandrivalinux.local.
221 Goodbye.
Connection closed by foreign host.
```

```
// vérification : 19*256 + 136 = 5000
$ netcat -l -s 192.168.52.2 -p 5000
total 48
drwxr-xr-x 6 root root 4096 Jun  1
16:12 .
drwxr-xr-x 6 root root 4096 Jun  1
16:12 ..
d--x--x--x 2 root root 4096 Jun  1 16:12
bin
d--x--x--x 2 root root 4096 Jun  1 16:12
etc
drwxr-xr-x 2 root root 4096 Jun  1 16:12
lib
drwxr-xr-x 2 root 79  4096 Jun  1 16:15
pub
```

Séquence 2 - Serveur FTP

Préparation de la manipulation : les besoins

a . Le serveur de base : outil capable d'accepter une connexion TCP sur un port local et d'échanger des informations en mode texte

Réponse : **netcat**

b . Identifier le port local à partir de la liste officielle du protocole **ftp**. Dans quel fichier trouve-t-on cette information ?

Réponse : **ftp -> 21 (dans le fichier /etc/services)**

c . Choisir un client HTTP pour l'échange.

Réponse : **firefox ou filezilla**

d . La commande pour lancer le serveur (sous root pour accéder au port 21)

Réponse : **nc -l -p 21**

Manipulation

1 . Envoyer les réponses FTP ci-dessous.

Réponses:

```
# nc -l -p 21
```

```
220 mandrivalinux.local FTP server
```

```
USER anonymous
```

```
331 Guest login ok
```

```
PASS mozilla@example.com
```

```
230 Guest login ok
```

```
SYST
```

```
215 UNIX emulated by FileZilla
```

```
PWD
```

```
257 "/"
```

```
TYPE I
```

```
200 Switching to Binary mode.
```

```
PASV
```

```
227 Entering Passive Mode (192,168,52,2,159,240)
```

```
SIZE /
```

```
212 600
```

```
MDTM /
```

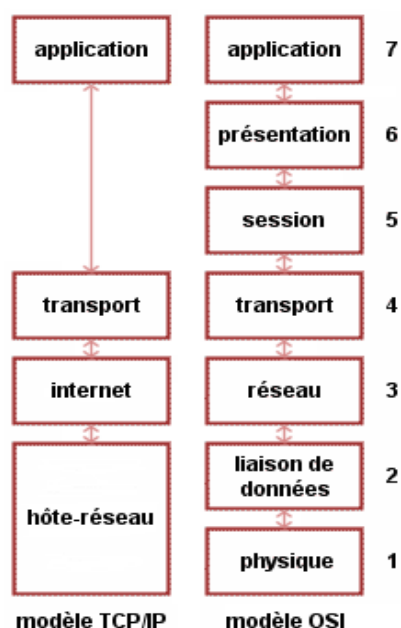
```
213 19980615100045.014
```

```
RETR /
```

```
^C
```

```
#
```

Annexe: les modèles de référence



- **La couche Application** : cette couche est l'**interface** entre l'application utilisateur et le réseau. Elle va apporter à l'utilisateur les services de base offerts par le réseau, comme par exemple le transfert de fichier, la messagerie ... Elle contient donc tous les protocoles de haut niveau, comme par exemple Telnet, TFTP (*Trivial File Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), HTTP (*HyperText Transfer Protocol*) ... Le point important pour cette couche est le choix du protocole de transport à utiliser. Par exemple, TFTP (surtout utilisé sur réseaux locaux) utilisera UDP, car on part du principe que les liaisons physiques sont suffisamment fiables et les temps de transmission suffisamment courts pour qu'il n'y ait pas d'inversion de paquets à l'arrivée. Ce choix rend TFTP plus rapide que le protocole FTP qui utilise TCP. A l'inverse, SMTP utilise TCP, car pour la remise du courrier électronique, on veut que tous les messages parviennent intégralement et sans erreurs.
- **La couche Session** du modèle OSI établit une communication entre émetteur et récepteur en assurant l'ouverture et la fermeture des sessions (des communications) entre usagers, définit les règles d'organisation et de synchronisation du dialogue entre les abonnés.
Exemple TCP/IP : RPC (*Remote Procedure Call*).
- **La couche Présentation** met en forme les informations échangées pour les rendre compatibles avec l'application destinatrice, dans le cas de dialogue entre systèmes hétérogènes. Elle peut comporter des fonctions de traduction, de compression, d'encryptage, ... etc.
Exemple TCP/IP : XDR (*eXternal Data Representation*).

Remarque: même s'il semble que les couches Session et Présentation du modèle OSI semblent inutiles dans le modèle TCP/IP, des protocoles équivalents existent et sont empilés dans la couche Application.

Annexe: le protocole FTP (*File Transfer Protocol*)

Introduction

Le protocole FTP est un protocole de transfert de fichier.

Sa mise en place date de 1971, date à laquelle un mécanisme de transfert de fichiers (décrit dans le RFC 141) entre les machines du MIT (*Massachusetts Institute of Technology*) avait été mis au point. De nombreux RFC ont ensuite apporté des améliorations au protocole de base, mais les plus grandes innovations datent de juillet 1973.

Le protocole FTP est actuellement défini par le RFC 959.

Le protocole FTP a pour objectifs de :

- permettre un partage de fichiers entre machine distante
- permettre une indépendance aux systèmes de fichiers des machines clientes et serveur
- permettre de transférer des données de manière efficace

Le protocole FTP s'utilise de façon standard sur le **port 21 du serveur en mode TCP**. Par contre le **FTP ne fonctionne que sur du TCP**.

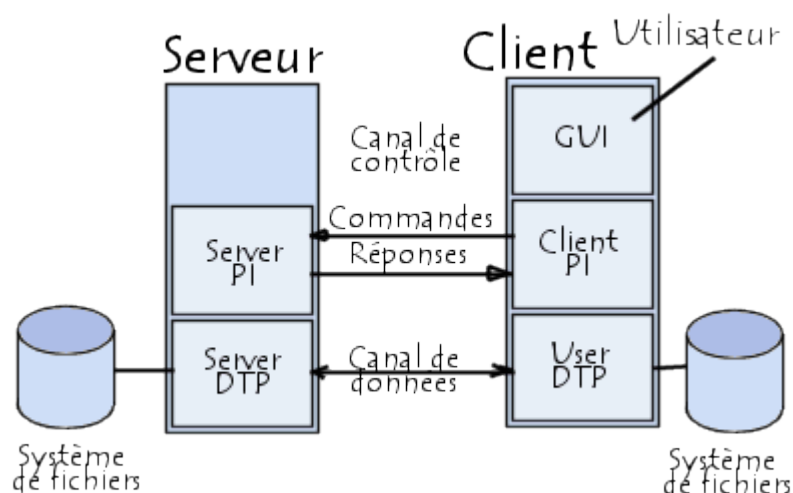
Remarque: il existe un protocole TFTP (*Trivial FTP*) qui est basé sur UDP.

Fonctionnement

Le protocole FTP est basé un modèle client-serveur.

Lors d'une connexion FTP, deux canaux de transmission sont ouverts :

- Un canal pour les commandes (canal de contrôle)
- Un canal pour les données



Ainsi, le client comme le serveur possède deux processus permettant de gérer ces deux types d'information :

- le **DTP** (*Data Transfer Process*) est le processus chargé d'établir la connexion et de gérer le canal de données. Le DTP côté serveur est appelé *SERVER-DTP*, le DTP côté client est appelé *USER-DTP*
- le **PI** (*Protocol Interpreter*) est l'interpréteur de protocole permettant de commander le DTP à l'aide des commandes reçues sur le canal de contrôle. Il est différent sur le client et sur le serveur :
 - Le *SERVER-PI* est chargé d'écouter les commandes provenant d'un *USER-PI* sur le canal de contrôle sur un port donné, d'établir la connexion pour le canal de contrôle, de recevoir sur celui-ci les commandes FTP de l'*USER-PI*, d'y répondre et de piloter le *SERVER-DTP*
 - Le *USER-PI* est chargé d'établir la connexion avec le serveur FTP, d'envoyer les commandes FTP, de recevoir les réponses du *SERVER-PI* et de contrôler le *USER-DTP* si besoin

Lors de la connexion d'un client FTP à un serveur FTP, le *USER-PI* initie la connexion au serveur selon le protocole Telnet. Le client envoie des commandes FTP au serveur, ce dernier les interprète, pilote son DTP, puis renvoie une réponse standard. Lorsque la connexion est établie, le serveur-PI donne le port sur lequel les données seront envoyées au Client DTP. Le client DTP écoute alors sur le port spécifié les données en provenance du serveur.

Remarque:

Il est important de remarquer que, les ports de contrôle et de données étant des canaux séparés, il est possible d'envoyer les commandes à partir d'une machine et de recevoir les données sur une autre. Ainsi, il est par exemple possible de transférer des données entre deux serveurs FTP en passant par un client pour envoyer les instructions de contrôle et en transférant les informations entre deux processus serveurs connectés sur le bon port.

Les commandes

Toutes les communications effectuées sur le canal de contrôle suivent les recommandations du protocole Telnet. Ainsi les commandes FTP sont des chaînes de caractères Telnet (en code NVT-ASCII) terminées par le code de fin de ligne Telnet (c'est-à-dire la séquence <CR>+<LF>, **Carriage Return** (retour chariot) suivi du caractère **Line Feed**, notée <CRLF>).

Si la commande FTP admet un paramètre, celui-ci est séparé de la commande par un espace (<SP>).

Les commandes FTP permettent de préciser :

- Le port utilisé
- Le mode de transfert des données
- La structure des données
- La nature de l'action à effectuer (Retrieve, List, Store, ...)

On distingue trois types de commandes FTP :

- Les commandes de contrôle d'accès :
- Les commandes du paramétrage de transfert
- Les commandes de service FTP

En voici, quelques unes :

Commandes de contrôle d'accès	
USER	Chaîne de caractère permettant d'identifier l'utilisateur. L'identification de l'utilisateur est nécessaire pour établir une communication sur le canal de données
PASS	Chaîne de caractère spécifiant le mot de passe de l'utilisateur. Cette commande doit être immédiatement précédée de la commande <i>USER</i> . Il revient au client de masquer l'affichage de cette commande pour des raisons de sécurité
QUIT	Commande permettant de terminer la session en cours. Le serveur attend de finir le transfert en cours le cas échéant, puis de fournir une réponse avant de fermer la connexion
Commandes du paramétrage de transfert	
PORT	Chaîne de caractère permettant de préciser le numéro de port à utiliser
PASV	Commande permettant d'indiquer au serveur DTP de se mettre en attente une connexion sur un port spécifique choisi aléatoirement parmi les ports disponibles. La réponse à cette commande est l'adresse IP de la machine et le port.
TYPE	Cette commande permet de préciser le type de format dans lequel les données seront envoyées
Commandes de service	
PWD	Cette commande (<i>Print Working Directory</i>) permet de renvoyer le chemin complet du répertoire courant
LIST	Cette commande permet de renvoyer la liste des fichiers et répertoires présents dans le répertoire courant. Cette liste est envoyée sur le DTP passif. Il est possible de passer en paramètre de cette commande un nom de répertoire, le serveur DTP enverra la liste des fichiers dans le répertoire passé en paramètre
RETR	Cette commande (<i>RETRIEVE</i>) demande au serveur DTP une copie du fichier dont le chemin d'accès est passé en paramètre.
STOR	Cette commande (<i>STORE</i>) demande au serveur DTP d'accepter les données envoyées sur le canal de données et de les stocker dans le fichier portant le nom passé en paramètre. Si le fichier n'existe pas, le serveur le crée, sinon il l'écrase

Les réponses

Les réponses FTP permettent d'assurer la synchronisation entre client et serveur FTP. Ainsi à chaque commande envoyée par le client, le serveur effectuera éventuellement une action et renverra systématiquement une réponse.

Les réponses sont constituées d'un code à 3 chiffres indiquant la façon suivant laquelle la commande envoyée par le client a été traitée. Toutefois, ce code à 3 chiffres étant difficilement lisible par un humain, il est accompagné d'un texte (chaîne de caractère Telnet séparée du code numérique par un espace).

Les codes de réponse sont constitués de 3 chiffres dont voici les significations :

- Le premier chiffre indique le statut de la réponse (succès ou échec)
- Le second chiffre indique ce à quoi la réponse fait référence
- Le troisième chiffre donne une signification plus spécifique

Premier chiffre		
Chiffre	Signification	Description
1yz	Réponse préliminaire positive	L'action demandée est en cours de réalisation, une seconde réponse doit être obtenue avant d'envoyer une deuxième commande
2yz	Réponse positive de réalisation	L'action demandée a été réalisée, une nouvelle commande peut être envoyée
3yz	Réponse intermédiaire positive	L'action demandée est temporairement suspendue. Des informations supplémentaires sont attendues de la part du client
4yz	Réponse négative de réalisation	L'action demandée n'a pas eu lieu car la commande n'a temporairement pas été acceptée. Le client est prié de réessayer ultérieurement
5yz	Réponse négative permanent	L'action demandée n'a pas eu lieu car la commande n'a pas été acceptée. Le client est prié de formuler une requête différente
Second chiffre		
Chiffre	Signification	Description
x0z	Syntaxe	L'action possède une erreur de syntaxe, ou bien il s'agit d'une commande non comprise par le serveur
x1z	Information	Il s'agit d'une réponse renvoyant des informations (par exemple pour une réponse à une commande STAT)
x2z	connexions	La réponse concerne le canal de données
x3z	Authentification et comptes	La réponse concerne le login (USER/PASS) ou la demande de changement de compte (CPT)
x4z	Non utilisé par le protocole FTP	
x5z	Système de fichiers	La réponse concerne le système de fichiers distant

Modes

Mode Passif : PASV

- Le client FTP envoie sur le canal de contrôle la commande **PASV**
- Le Serveur répond "xxx (IP1,IP2,IP3,IP4,PORT1,PORT2)"
- Le client décode la phrase, calcule le N° de Port = $PORT1 * 256 + PORT2$
- **Le client se connecte** à l'adresse IP donnée sur le N° de Port donné et ouvre le port de transfert
- Une fois la connexion ouverte, le client envoie sa commande sur le port de contrôle (LIST, RETR, ...)
- Les données sont reçues sur le port data, et dès l'envoi terminé, le serveur envoie sur le port contrôle une phrase de fin de transfert, et ferme sa connexion de port de transfert. Attention le message de fin sur le port contrôle arrive souvent avant la fin de la réception des données sur le port transfert.

Mode Actif : PORT IP1,IP2,IP3,IP4,PORT1,PORT2

- Le client FTP envoie sa commande "**PORT IP1,IP2,IP3,IP4,PORT1,PORT2** "
- Le serveur répond qu'il est d'accord
- Le client envoie sa commande (LIST, RETR, ...)
- **Le serveur se connecte** sur le client sur le port spécifié par celui ci.
- Le transfert se déroule de la même manière qu'en mode passif.

Le port de transfert de données est spécifié soit par le client qui indique au serveur son N° et attend sa connexion (mode standard ou actif, déclenché par la commande PORT), soit par le serveur qui indique au client le N° du port auquel il doit se connecter (mode passif, déclenché par la commande PASV).

Cette notion est extrêmement importante pour l'utilisation de FTP avec des pare-feux (*firewall*) :

- Si le client a un firewall, le mode standard ne fonctionnera pas car le serveur n'arrivera jamais à se connecter au client pour transférer les données, et par exemple la commande LIST ne fonctionnera pas puisqu'elle a besoin d'un canal de données.
- Si le serveur a un firewall, il faut configurer celui ci pour qu'il laisse passer le port du serveur (21) et une plage de ports pour les transferts si le serveur accepte le mode passif.

Remarque:

Le serveur doit avoir une plage de ports (entre X et Y) pour les transferts même s'il n'a qu'un seul client, car chaque commande LIST utilise un port de données, et un port de données ne peut plus être utilisé pendant presque une minute dans certains cas. Donc pour éviter tout problème à chaque fois que le client ou le serveur se mettent en attente, ils changent de port sans essayer de réutiliser le port précédent.

Annexe : clients ftp

FileZilla est un client FTP, FTPS et SFTP, multiplate-forme depuis la version 3. C'est un logiciel libre développé sous la licence publique générale GPL/GNU. Il permet de transférer des fichiers.

Lien : <http://filezilla-project.org/>

Les paramètres de connexion :

L'**adresse IP** du serveur

Le compte de connexion : identifiant + le mot de passe

21 (le numéro de port du service FTP)

Une fois les paramètres saisis, on clique sur **Connexion rapide** pour se connecter sur le serveur FTP

The screenshot shows the FileZilla interface with the following annotations:

- Host:** 192.168.52.254 (indicated as the server IP address)
- Identifiant:** root (indicated as the username)
- Mot de passe:** **** (indicated as the password)
- Port:** 21 (indicated as the FTP service port)
- Connexion rapide** button (indicated as the connection button)
- le protocole FTP en action !** (pointing to the terminal window showing commands like PASV, LIST, and directory listings)
- permet d'accéder à l'historique des connexions et de rappeler rapidement les paramètres d'une connexion** (pointing to the 'Connexion rapide' button)
- transfert de fichier par glisser/déposer** (pointing to the file lists in both local and remote panes)
- Sens descendant (client ← serveur) : download** (indicated by a blue arrow pointing from the remote pane to the local pane)
- Sens montant (client → serveur) : upload** (indicated by a blue arrow pointing from the local pane to the remote pane)
- Cadre Client (local)** (label for the left pane)
- Cadre Serveur (distant)** (label for the right pane)

Terminal output (FTP protocol in action):

```

Réponse: 200 Type set to binary
Commande: PASV
Réponse: 227 Entering Passive Mode (192,168,52,254,226,147)
Commande: LIST
Réponse: 150 Opening ASCII mode data connection for file list.
Réponse: 226 Closing data connection.
Statut: Succès de la lecture du contenu du dossier
    
```

Nom de fichier	Taille de fic...	Type de fichier	Dernière modification
..			
ModulePanneau...	1 772	CLASS-fichier	26/03/2009 08:38:51
CPanneau.class	341	CLASS-fichier	26/03/2009 08:38:51
panneau_v1.tini	8 542	TINI-fichier	26/03/2009 08:38:52
TGestionPanneau...	1 050	CLASS-fichier	26/03/2009 08:38:51

Nom de fichier	Taille de fi...	Type de fic...	Dernière modifi...	Permissions	Propriétaire...
..					
etc		Dossier de ...	12/03/2009 12:...	drwx--r-x	root admin
panneau.tini	8 530	TINI-fichier	21/03/2009 13:...	-rwx--r--	root admin
panneau_v1.tini	8 542	TINI-fichier	26/03/2009 09:...	-rwx--r--	root admin
panneau_v2.tini	13 594	TINI-fichier	26/03/2009 10:...	-rwx--r--	root admin

Les systèmes d'exploitation classique fournissent généralement un utilitaire **ftp** en mode commande.

Exemple sous Linux :

```
$ ftp
ftp> help
Commands may be abbreviated.  Commands are:
!                cr                mdir                proxy                send
$                delete                mget                sendport                site
account          debug                mkdir                put                size
append          dir                mls                pwd                status
ascii           disconnect          mode                quit                struct
bell            form                modtime            quote                system
binary          get                mput                recv                sunique
bye             glob                newer                reget                tenex
case            hash                nmap                rstatus                trace
ccc            help                nlist                rhelp                type
cd             idle                ntrans             rename                user
cdup           image                open                reset                umask
chmod          lcd                passive            restart                verbose
clear          ls                 private            rmdir                ?
close         macdef             prompt             runique
cprotect      mdelete           protect            safe
```

```
ftp> open ftp.free.fr
Connected to ftp.proxad.net.
220 Welcome to ProXad FTP server
530 Please login with USER and PASS.
Name (ftp.free.fr:tv): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
lrwxrwxrwx    1 ftp      ftp                28 Sep 27  2007 MPlayer ->
mirrors/mplayerhq.hu/MPlayer
drwxr-xr-x    2 ftp      ftp                4096 May 07  2008 awstats
drwx-----  2 ftp      ftp               16384 Mar 08  2006 lost+found
drwxr-xr-x    3 ftp      ftp                4096 Feb 04  09:34 mirrors
drwxr-xr-x    2 ftp      ftp                4096 Dec 24  2008 nzb
drwxr-xr-x   10 ftp      ftp                4096 Nov 23  18:20 pub
drwxr-xr-x    2 ftp      ftp               655360 Mar 18  23:30 stats
drwxr-xr-x    2 ftp      ftp                4096 Mar 19  09:39 tmp
226 Directory send OK.
```

```
ftp> bye
221 Goodbye.
```