

FTP & SMTP

Deux applications fondamentales
pour le réseau Internet.

File Transfert Protocol

Rapide Historique :

1971 : Première version du protocole définit par le M.I.T.

1973 : Première documentation officielle du protocole FTP.

1975 : Evolution de FTP pour pouvoir fonctionner au dessus de TCP (jusqu'alors FTP utilisé NCP).

1982 : Finalisation de la définition du rôle de FTP : « Le *File Transfert Protocol* est désormais définit comme un protocole de transfert de fichier entre des hôtes d'un ARPANET, afin de profiter de l'utilisation d'une capacité de stockage de données distante »

File Transfert Protocol

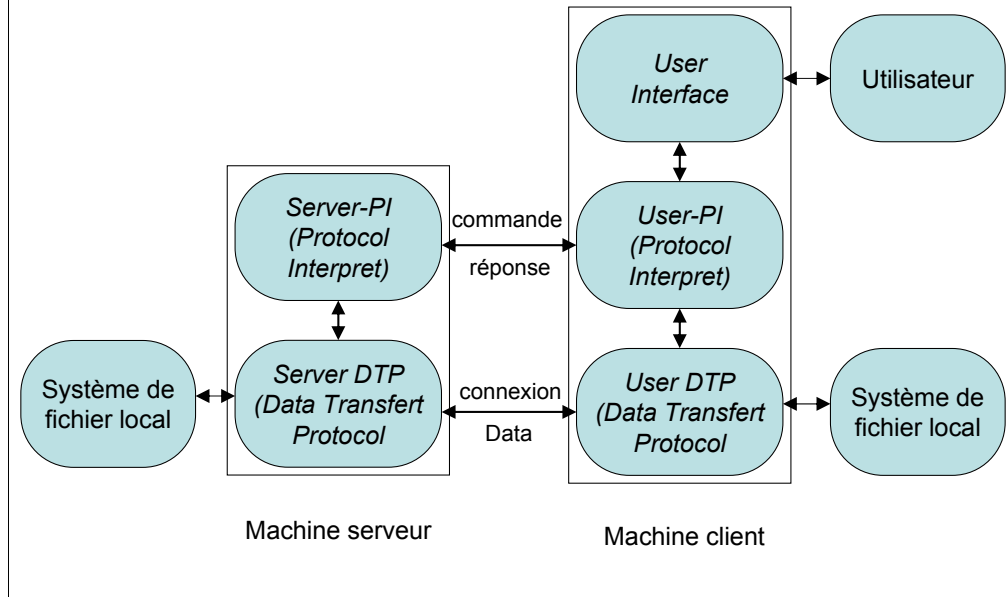
Un protocole d'échange de fichier « au dessus » de TCP :

⇒ Définition d'un client et d'un serveur

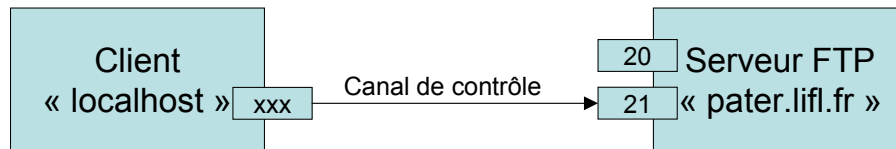
Le client FTP est la machine de l'utilisateur. Le serveur est la machine sur laquelle est placée le système de fichier.

⇒ Prévu pour être exploité par l'intermédiaire de clients dédiés, mais disponible à une exploitation directe (via un client `telnet`).

Principe général de FTP



Etablissement du canal de contrôle



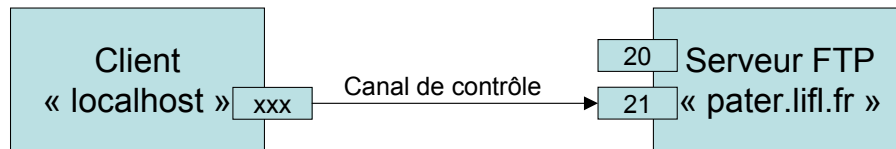
```
> telnet pater.lifl.fr 21
220 G6 FTP Server ready ...
USER toto
331 Password required for toto.
PASS otot
230 User toto logged in.
HELP
214-Supported Commands :
214-PORT STOR APPE RETR CWD
214-PWD XPWD USER PASS LIST
214-NLST TYPE SYST QUIT DELE
214-SIZE REST RNFR RNTD XMKD
214-MKD RMD ABOR PASV NOOP
214-CDUP SITE HELP STAT STOU*
214-MDTM STRU SMNT XCUP ACCT
```

```
214-ALLO REIN MODE XRMD XDEL
214 [End of Help].
```

Fonctionnement du canal de contrôle :

- ⇒ connexion de type telnet ;
- ⇒ le serveur FTP attend des connexions clients ;
- ⇒ les échanges respectent un ordre établi :
 1. le serveur attend des commandes :
format ASCII, 4 caractères + arguments.
 2. le serveur retourne une réponse :
3 chars formant un nombre : code retour
1 char ' ' ou '-' format fin de la réponse ou pas
x chars commentaires « lisibles ».
- ⇒ le client termine les échanges en clôturant la socket,
ou avec une commande LOUGOUT.

Le canal de contrôle



Commandes de gestion du canal de contrôle :

USER <username>

Définir l'utilisateur de la session en cours.

PASS <password>

Envoyer le mot de passe associé à l'utilisateur.

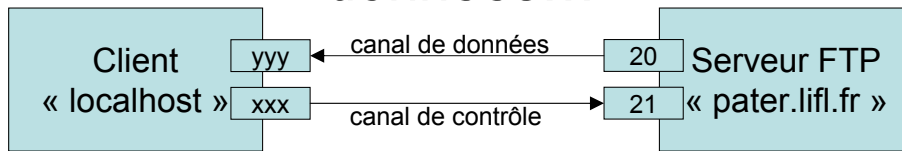
REIN

Réinitialiser la session en cours, sans perdre la connexion.

QUIT

Termine la connexion en cours.
Le serveur rompt la Socket.

Etablissement du canal de données...



Cas par défaut :

Paramétrage du canal de donnée :

PORT 134,206,10,153,20,0

200 Port command successful.

Ouverture du canal de donnée :

Lorsque le canal de donnée doit être ouvert,

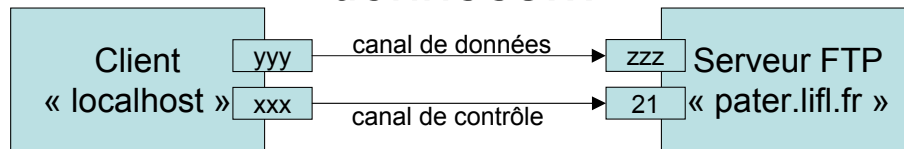
il l'est sous l'initiative du serveur.

Dans ce cas le serveur ouvre le port et l'IP
indiqué par le client via la commande PORT.

Fermeture :

Sur l'initiative du serveur, lorsque les transferts
associés à la commande en cours sont terminés.

Etablissement du canal de données...



Mode passif :

Paramétrage du canal de donnée (demande de mode passif) :

PASV

227 Entering Passive Mode (127,0,0,1,55,12).

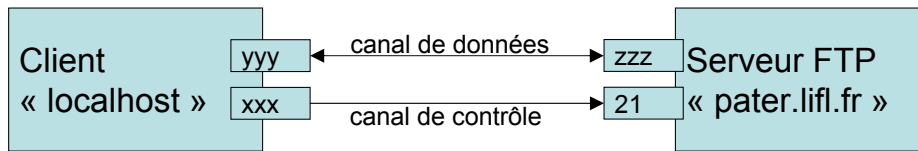
Ouverture du canal de donnée :

Le canal de donnée est ouvert sur l'initiative du client. Le port que le client doit ouvrir est indiqué dans la réponse à la commande PASV. Une fois le canal ouvert, le serveur peut transférer les données associées à la commande suivante (si cette commande déclenche des données sortantes...).

Fermeture :

Sur l'initiative du serveur, lorsque les transferts associés à la commande en cours sont terminés.

Le canal de contrôle



Commandes du paramétrage des transferts :

PORT <IP1>,<IP2>,<IP3>,<IP4>,<PORT1>,<PORT2>

Configurer la cible d'émission pour le canal de contrôle avec :

- IP, les 4 octets de l'adresse IP de la cible ;
- Port, deux nombres entre 0 et 255 qui donne le port sur 16bits.

PASV

Positionne le serveur en mode passif. Il retourne l'adresse et le port sur lequel il attend une connexion du client.

TYPE <param>

Définit le type d'échange réalisé via la socket de donnée :

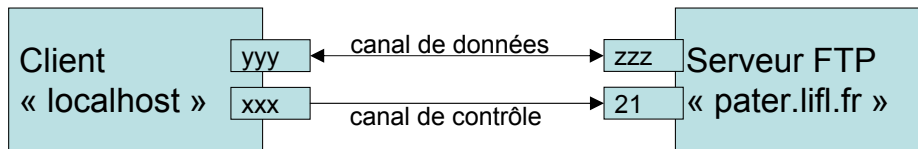
<param> = **A** ASCII, **E** EBCDIC, **I** Image.

MODE <param>

Détermine le mode d'encodage des données.

<param> = **S** Flux, **B** Block, **C** Compressé.

Exploitation d'un serveur FTP



Commandes de transfert FTP :

RETR <filename>

Déclanche la transmission par le serveur du fichier <filename> sur le canal de données.

STOR <filename>

Déclanche la réception d'un fichier qui sera enregistré sur le disque sous le nom <filename>. Si un fichier avec le même nom existe déjà il est remplacé par un nouveau avec les données transmises.

APPE <filename>

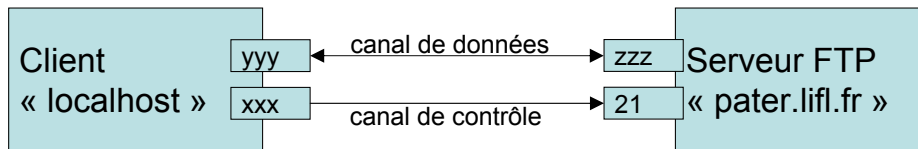
Déclanche la réception d'un fichier qui sera enregistré sur le disque sous le nom <filename>. Si un fichier avec le même nom existe déjà, les nouvelles données lui sont concaténées.

REST <offset>

Redémarrage en cas d'échec d'un transfert précédent. L'offset précise le numéro du dernier octet reçu.

ABOR : abandon d'un transfert en cours.

Exploitation d'un serveur FTP



Commandes de gestion du système de fichier distant et du serveur FTP :

PWD : impression du répertoire courant.

LIST : catalogue du répertoire courant (canal donnée).

NLST : catalogue succinct (canal donnée).

CWD <repname> : changement de répertoire courant pour <repname>.

MKD <repname> : création du nouveau répertoire <repname>.

RMD <repname> : suppression du répertoire <repname>.

DELE <filename> : suppression du fichier <filename>.

RNFR <filename1> : définit le nom actuel d'un fichier à renommer.

RNTO <filename2> : définit le nouveau nom d'un fichier à renommer.

STAT : status courant de la session FTP.

STAT <repname> : équivalent à LIST mais réponse sur le canal de contrôle.

HELP : affiche l'aide sur les opérations du site.

NOOP : no operation.

Code de retour

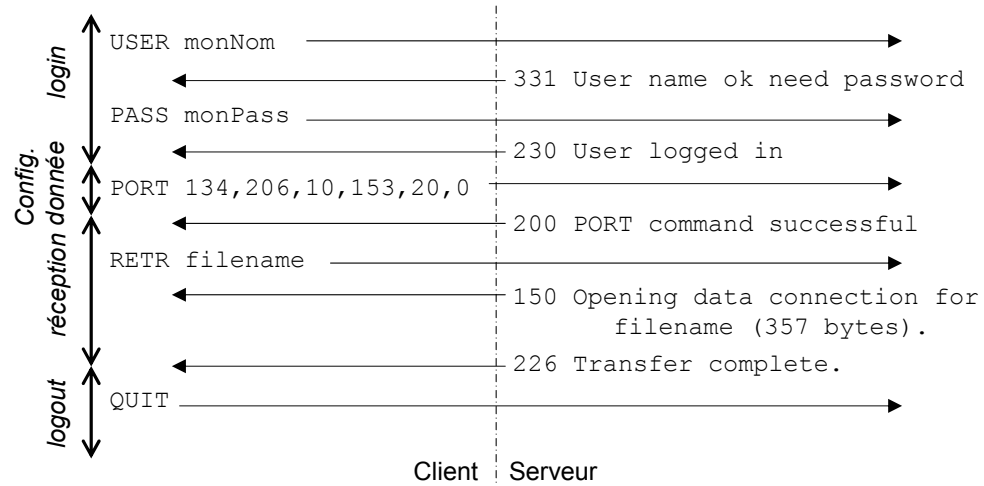
Les codes de retour :

1yz : réponse positive préliminaire	x0z : erreur de syntaxe
2yz : réponse positive définitive	x1z : réponse contenant des informations
3yz : réponse positive intermédiaire	x2z : réponse vis à vis de la connexion
4yz : réponse négative transitoire	x3z : identification et authentification
5yz : réponse négative définitive	x4z : non spécifié
	x5z : système de fichier

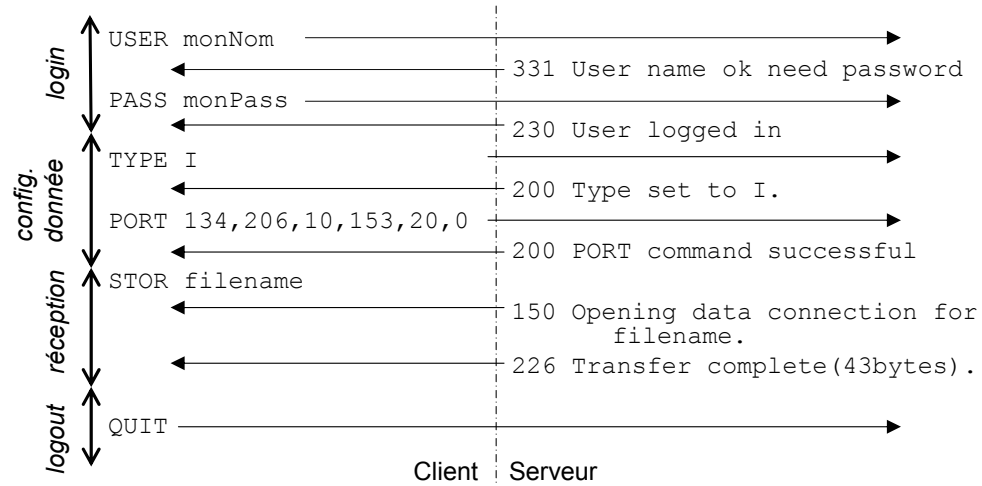
La troisième digit raffine la description d'erreur, exemple :

- 501 Erreur de syntaxe, commande non connue.
- 502 Erreur de syntaxe dans les paramètres/arguments.
- 503 Erreur de syntaxe, commande non implémentée.
- 504 Erreur de syntaxe, mauvaise séquence de commandes.
- 505 Erreur de syntaxe, commande non implémentée pour ce param.

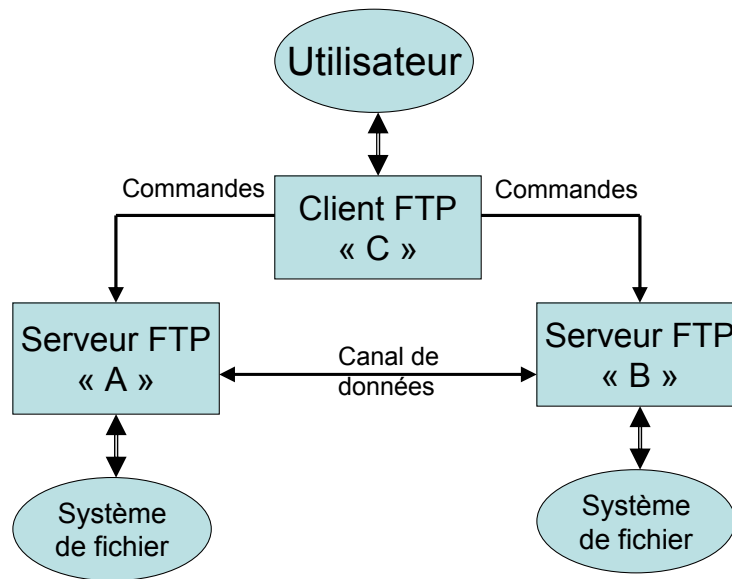
Scénario 1 : réception d'un fichier



Scénario 2 : émission d'un fichier



Fonctionnement alternatif



Simple Mail Transfert Protocol

Rapide Historique :

1971 : FTP utilisé en temps que support d'échange de mail.

1982 : Première documentation officielle du protocole
SMTP : RFC 821 & RFC 822.

1986 : Normalisation de X400/CCITT devenu MOTIS/ISO

Simple Mail Transfert Protocol

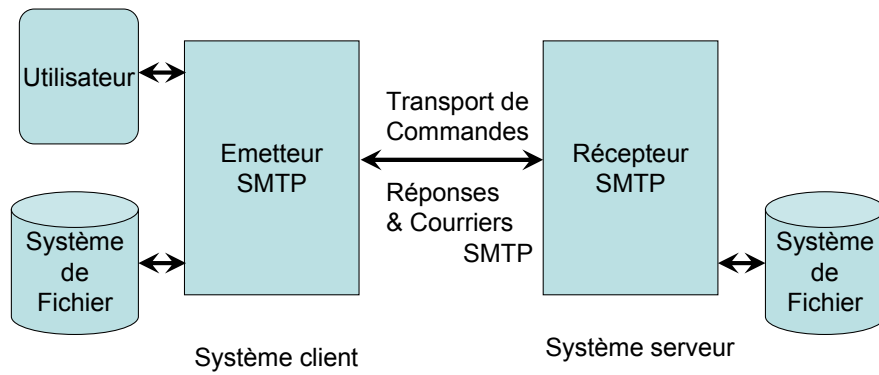
Protocole d'échange de messages électroniques indépendant du protocole de transport sous-jacent.

N.B. : Pour les réseaux IP, SMTP est implanté au dessus de TCP, port 25.

⇒ Un **serveur SMTP** est une machine « cible » qui se présente comme un « bureau de poste » vis à vis de **clients SMTP**.

⇒ Protocole directement accessible vis `telnet`.

SMTP : principes fondateurs



Récepteur SMTP :
⇒ bureau de poste ;
⇒ centre de tri.

SMTP : les commandes de base

HELO <domaine> : Initialisation de la session SMTP
MAIL FROM:<route-inverse> : déclaration de l'émetteur du mail
RCPT TO:<route-directe> : déclaration du destinataire du mail
DATA : initialisation de la séquence de saisie des données
RSET : initialisation de la séquence de saisie des données
SEND FROM:<route-inverse> : message direct plutôt que postage.
SOML FROM:<route-inverse> : message direct OU postage.
SAML FROM:<route-inverse> : message direct ET postage.
VRFY <chaîne> : vérification de l'existence d'un destinataire
EXPN <chaîne> : extraction des destinataires inscrits dans une liste de diffusion
HELP [<chaîne>] : demande d'aide (éventuellement sur une commande)
NOOP : aucune opération
QUIT : clôture de la session SMTP
TURN : demande d'inversion des rôles d'émetteur et de récepteur

Les réponses respectent un format comparable à celui de FTP.

Scénario 1 : validation d'une adresse mail

>telnet monserveursmtp 25	←	220 lifl.lifl.fr ESMTS Sendmail 8.9.3/jtpda-5.3.3 ready at Mon 22 Apr 2002 11:36:22 +0200<EOL>	→
HELO pater.lifl.fr<EOL>	←	250 lifl.lifl.fr Hello pater [134.206.10.153], pleased to meet you<EOL>	→
VRFY gigrimaud<EOL>	←	550 gigrimaud... User unknown<EOL>	→
VRFY grimaud<EOL>	←	250 Gilles.Grimaud <grimaud@lifl.lifl.fr><EOL>	→
QUIT<EOL>	←	221 lifl.lifl.fr closing connection<EOL>	→
	Client	Serveur	

Scénario 2 : envoi d'un message

```
← 220 lifl.lifl.fr ESMTS Sendmail
8.9.3/jtpda-5.3.3 ready at Mon
22 Apr 2002 11:36:22 +0200 <EOL>
HELO pater.lifl.fr<EOL> →
← 250 lifl.lifl.fr Hello pater
[134.206.10.153], pleased to
meet you <EOL>
MAIL FROM:grimaud@lifl.fr<EOL> →
← 250 grimaud@lifl.fr... Sender ok <EOL>
RCPT TO: grimaud@lifl.fr<EOL> →
← 250 grimaud@lifl.fr... Recipient ok <EOL>
RCPT TO:jean@lifl.fr<EOL> →
← 250 jean@lifl.fr... Recipient ok <EOL>
DATA<EOL>
← 354 Enter mail, end with "." on a
line by itself <EOL>
Bla bla bla bla<EOL> →
Bla bla bla bla<EOL> →
.<EOL> →
← 250 LAA21250 Message accepted for
delivery <EOL>
QUIT<EOL> →
← 221 lifl.lifl.fr closing connection <EOL>
```

Corps d'un message SMTP

Chemin SMTP
 entête
 corps
 1ère Partie
 1ème Partie

Return-Path: <pieter@cc.uttente.nl>
 Received: from malonne.lifl.fr by lifl.lifl.fr for <grimaud@lifl.fr>;
 Received: from netlx010.civ.uttente.nl by malonne.lifl.fr for <grimaud@lifl.fr>;
 Received: from utiw32.cc.uttente.nl by netlx010.civ.uttente.nl; <grimaud@lifl.fr>;
 Message-ID: <00bd01c1b621\$7d2b99f0\$830f5982@utiw32>
 From: "Pieter Hartel" <pieter@cc.uttente.nl>
 To: "Gilles Grimaud" <grimaud@lifl.fr>,
 Subject: Re: Strategic Roadmap for Smart Card Research
 Date: Fri, 15 Feb 2002 14:05:48 +0100
 MIME-Version: 1.0
 Content-Type: multipart/mixed;
 boundary="-----_NextPart_000_0344_01C1D021.24302E00"
 Status:
 This is a multi-part message in MIME format.
 -----_NextPart_000_0344_01C1D021.24302E00
 Content-Type: text/plain;
 charset="iso-8859-1"
 Content-Transfer-Encoding: 7bit
 Dear RESET partner,
 Can I please have your comments on the table, no later than monday morning?
 Please don't forget to send your A3 to ERCIM.
 I attach the latest draft of the proposal for your information.
 --pieter
 -----_NextPart_000_0344_01C1D021.24302E00
 Content-Type: application/msword;
 name="RESET Part B-v0.4.doc"
 Content-Transfer-Encoding: base64
 Content-Disposition: attachment;
 filename="RESET Part B-v0.4.doc"
 OM8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAAAEAAAAhgEAAAAAAAAA
 EAAAIaEAAAEAAAD+////AAAAH4BAAB/AQAAjAEAAATcBAAD////////////////////////////////////
 AAAJBByAnFYBAKLYAACTWAAakpMAAGAFADJAAAAAAAAAAAAAAAAAAAAAwAAAAAAAAAD//w8AAAAA

Supports de références sur les protocoles de l'Internet : RFC¹

Disponibles sur le Web :

`http://www.ietf.org/rfc`

Tr. français `http://abcdrfc.free.fr`

RFC relatifs à FTP :

RFC 765, dernière version : **RFC 959**

RFC relatifs à SMTP :

RFC 772, RFC 780, RFC 788,
dernière version : **RFC 821 & RFC 822**

1 - RFC : *Request For Comments*