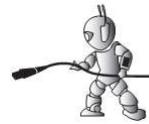


## RT 21

### Administration des Services Réseaux



## Chapitre IV : Service de messagerie et services d'application

### File Transfer Protocol (FTP)

Dr. H. Zerrouki  
<zerrouki.hadj@gmail.com>

UDL-SBA, Faculté de Génie Electrique

DÉPARTEMENT  
Télécommunications



## Plan de cours



File Transfer Protocol (FTP)

Protocoles de transfert de fichiers

Le modèle FTP

Les modes FTP

❖ Mode Actif  
❖ Mode passif

Protocole FTP

## File Transfer Protocol (FTP)

- Pour transférer les fichiers volumineux, nous utilisons un **FTP**, il s'agit d'un *File Transfer Protocol* (protocole de transfert de fichiers),
- Le protocole FTP fait parti de la suite TCP/IP défini par la RFC 959.
- FTP est un protocole de communication destiné à l'échange informatique de fichiers sur un réseau.
- Nous mettons les fichiers en ligne (dans le Cloud ou nuage) et vous pouvez les récupérer...
- Utiliser pour coordonner les transferts de fichiers vers des sites Web distants ou le transfert de fichiers de site Web distant vers des ordinateurs locaux. (Client/serveur et Serveur/client).

## Protocoles de transfert de fichiers

### Plusieurs protocoles

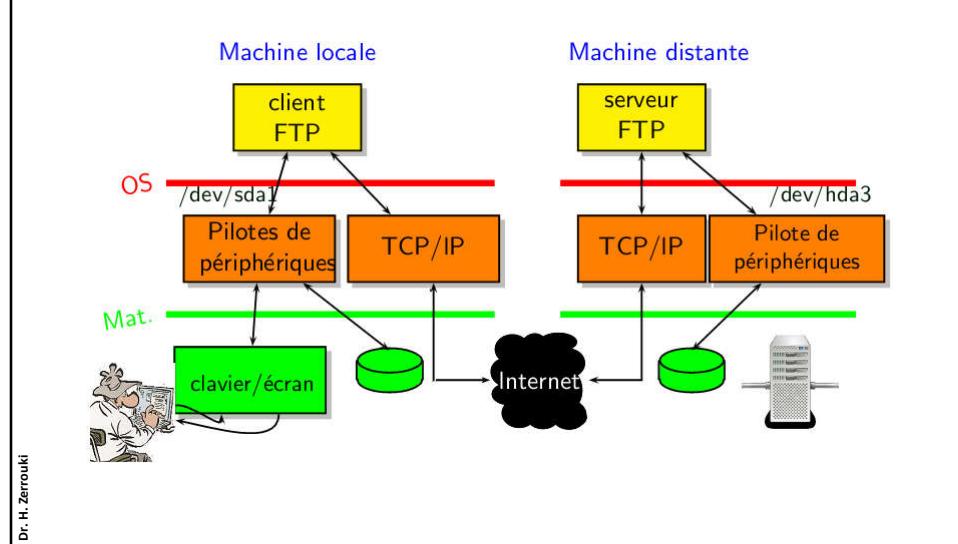
- copie de fichiers à distance : **RCP**, **SCP** (UNIX et Linux);
- protocole de transfert de fichiers avec accès aux systèmes de fichiers local et distant : **FTP**, **TFTP**, **SFTP**.
- copie de fichier sur http : **WebDAV**

### Type client/serveur

- **le client** (initiateur de la connexion) interagit avec l'utilisateur, le système de fichiers local et les protocoles réseau;
- **le serveur** (héberge les fichiers distants) interagit avec les protocoles réseau et le système de fichiers distant.

Ne pas confondre avec les protocoles d'accès aux fichiers distants : NFS (RPC), SMB (Microsoft).

## Protocoles de transfert de fichiers



Dr. H. Zerrouki

## Connexions contrôle et données

- Le protocole FTP s'inscrit dans un modèle **client-serveur**, c'est-à-dire qu'une machine envoie des ordres (le client) et que l'autre attend des requêtes pour effectuer des actions (le 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)
- Les clients FTP contactent le serveur FTP sur le port **TCP / 21**
- Ouverture de 2 connexions TCP parallèles :
  - Contrôle** : échange des commandes et des réponses entre le client et le serveur - << contrôle hors-bande >> (sur le port **TCP/21** côté client).
  - Données** : transfert des fichiers de données vers/depuis l'hôte distant (sur le port **TCP/20** côté client).

Dr. H. Zerrouki

## Connexions contrôle et données

### Connexion contrôle (FTP) :

- échange des requêtes/réponses (dialogue client/serveur);
- permanente, full-duplex, besoin de fiabilité (et faible délai !);
- initiée par le client (port **21** en général).

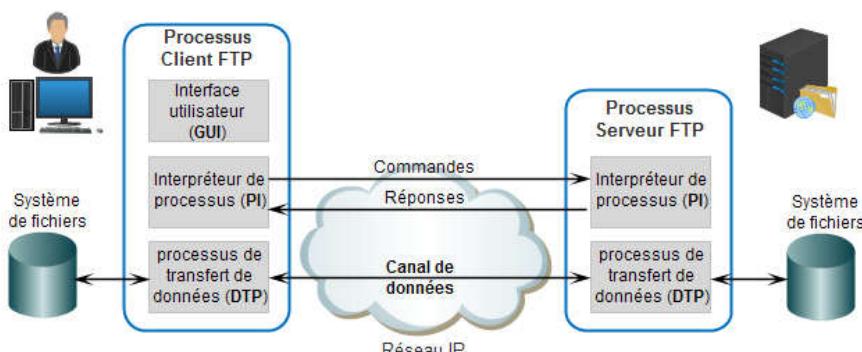
**Rôle** : pour transmettre les commandes de fichiers (transfert, suppression de fichiers, renommage, liste des fichiers...).

### Connexion données (FTP-DATA) :

- envoi de fichiers ou listes de fichiers/répertoires (données);
- temporaire, full-duplex, besoin de débit (et fiabilité !);
- initiée par défaut par le serveur
  - \* ouverture active (connect()) du serveur vers le client (depuis le port **20** vers un port proposé par le client);
- la connexion est fermée dès que le caractère EOF (End Of File) est lu.

## Le modèle FTP

- le client comme le serveur possèdent deux processus permettant de gérer ces deux types d'information :
  - Le processus Client FTP
  - Le processus Serveur FTP



## Le modèle FTP

**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.

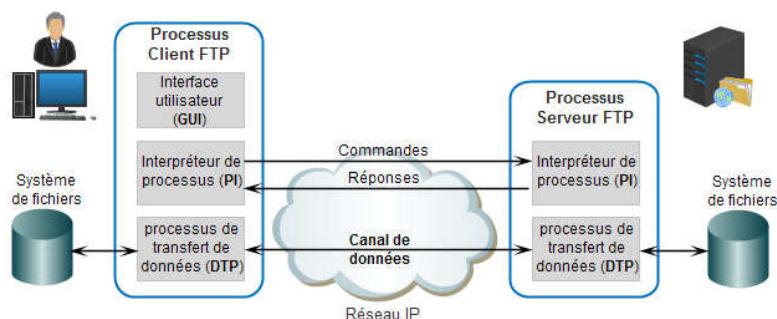
**PI (Protocol Interpreter)** est l'interpréteur de protocole permettant de commander le DTP à l'aide des commandes reçues sur le canal de contrôle. Il est différent sur le client et sur le serveur :

- Le **SERVER-PI** est chargé de :
  - écouter les commandes provenant d'un USER-PI sur le canal de contrôle sur un port donné,
  - établir la connexion pour le canal de contrôle,
  - recevoir sur celui-ci les commandes FTP de l'USER-PI,
  - répondre et de piloter le SERVER-DTP.
- Le **USER-PI** est chargé de :
  - établir la connexion avec le serveur FTP,
  - envoyer les commandes FTP,
  - recevoir les réponses du SERVER-PI
  - contrôler le USER-DTP si besoin.

## Le modèle FTP

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.



## Les modes FTP

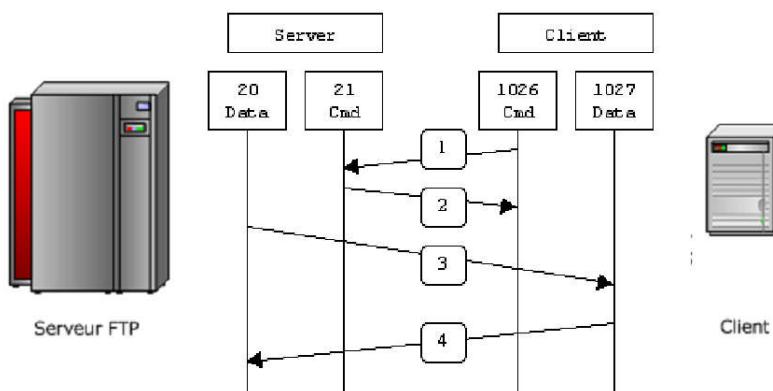
FTP prévoit **deux (2) modes** de fonctionnement :

### Mode actif :

- Le client initie la connexion de contrôle sur le port **21** en spécifiant le mode actif et en indiquant 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 va initier la connexion de son port de données (port **20**) vers le port spécifié par le client. C'est le numéro de port précédent le port de contrôle.
- En mode FTP actif, le client se connecte à partir d'un port aléatoire non privilégié (**N > 1023**) au port de commande du serveur FTP, le port **21**. Ensuite, le client commence à écouter le port **N+1** et envoie la commande **FTP PORT N+1** au serveur FTP. Le serveur se reconnectera ensuite au port de données spécifié du client à partir de son port de données local, qui est le port **20**..

## Les modes FTP

### Mode actif :

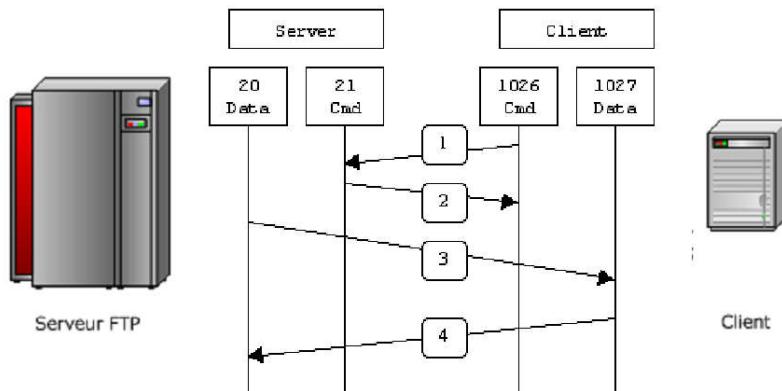


1: Le port de commande du client (**1026**) contacte le port de commande du serveur (**21**) et envoie la commande **PORT 1027**.

2: Le serveur renvoie ensuite un **ACK** au port de commande du client (**1026**).

## Les modes FTP

### Mode actif :



**3:** Le serveur initie une connexion sur son port de données local (**20**) au port de données que le client a spécifié précédemment (**1027**).

**4:** Enfin, le client renvoie un **ACK** au port de données du serveur (**20**).

## Les modes FTP

### 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'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. C'est pour cette raison que ce mode est qualifié de *firewall-friendly*.

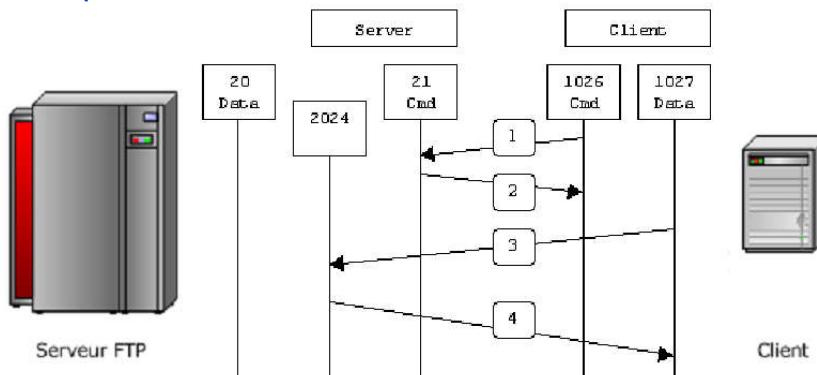
## Les modes FTP

### Mode passif :

- En mode FTP passif, le client initie les deux connexions au serveur, résolvant le problème des pare-feux, filtrant la connexion du port de données et entrant vers le client depuis le serveur.
- Lors de l'ouverture d'une connexion FTP, le client ouvre localement deux ports aléatoires non privilégiés (**N > 1023** et **N+1**). Le premier port contacte le serveur sur le port **21**, mais au lieu d'émettre ensuite une commande **PORT** et de permettre au serveur de se reconnecter à son port de données, le client émettra la commande **PASV**.
- Le résultat est que le serveur ouvre alors un port aléatoire non privilégié (**P > 1023**) et renvoie **P** au client en réponse à la commande **PASV**. Le client initie alors la connexion du port **N+1** au port **P** sur le serveur pour transférer des données.

## Les modes FTP

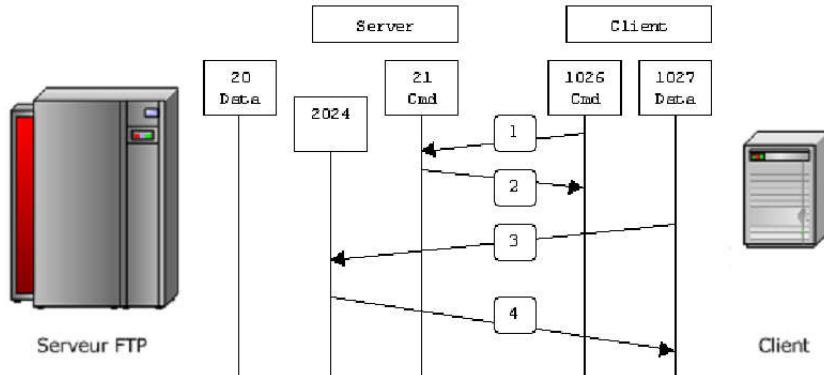
### Mode passif



- 1: Le client contacte le serveur sur le port de commande (**21**) utilisant le port client (**1026**) et émet la commande **PASV**.
- 2: Le serveur répond ensuite avec le **PORT 2024**, indiquant au client sur quel port il écoute la connexion de données.

## Les modes FTP

### Mode passif

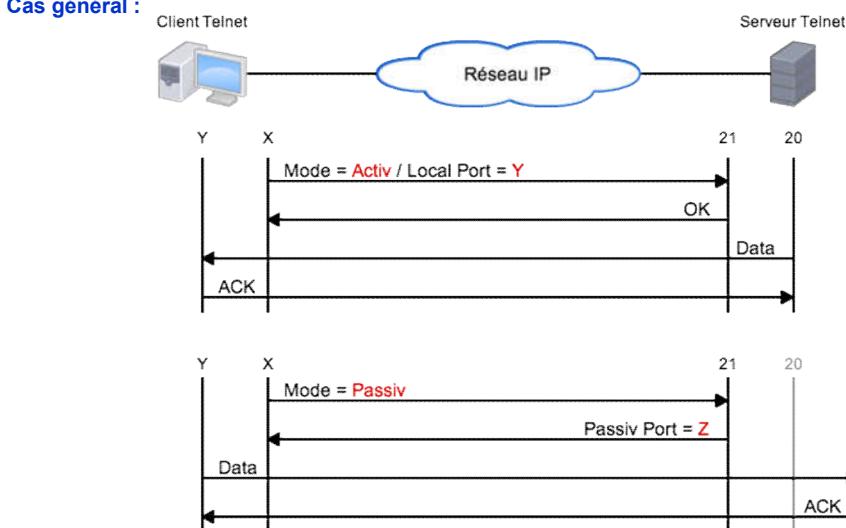


3: Le client initie ensuite la connexion de données de son port de données (1027) au port de données spécifié du serveur (2024).

4: Enfin, le serveur renvoie un ACK au port de données du client (1027).

## Les modes FTP

### Cas général :



## Protocole FTP

### Les commandes FTP :

- 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 **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

## Protocole FTP

### a) Commande de contrôle d'accès :

Commande	Description
<b>USER</b>	Chaîne de caractères permettant d'identifier l'utilisateur. L'identification de l'utilisateur est nécessaire pour établir une communication sur le canal de données
<b>PASS</b>	Chaîne de caractères spécifiant le mot de passe de l'utilisateur. Cette commande doit être immédiatement précédée de la commande <b>USER</b> .
<b>ACCT</b>	Chaîne de caractères représentant le compte (account) de l'utilisateur. Cette commande n'est généralement pas nécessaire. Lors de la réponse à l'acceptation du mot de passe, si la réponse est 230 cette phase n'est pas nécessaire, si la réponse est 332, elle l'est
<b>CWD</b>	<i>Change Working Directory</i> : cette commande permet de changer le répertoire courant. Cette commande nécessite le chemin d'accès au répertoire à atteindre comme argument
<b>CDUP</b>	<i>Change to Parent Directory</i> : cette commande permet de remonter au répertoire parent. Elle a été introduite pour remédier aux problèmes de nommage de répertoire parent selon les systèmes (généralement "..")
<b>SMNT</b>	<i>Structure Mount</i> : cette commande permet à l'utilisateur de monter une structure de données de système de fichiers différente sans modifier ses informations de connexion ou de compte.
<b>REIN</b>	<i>Reinitialize</i> : utilisée pour réinitialiser la connexion FTP.
<b>QUIT</b>	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.

## Protocole FTP

### b) Commande de paramètres de transfert :

Commande	Description
<b>PORT</b>	Chaîne de caractères permettant de préciser le numéro de port à utiliser en mode actif.
<b>PASV</b>	Commande permettant d'indiquer au serveur DTP de se mettre en attente une connexion, en mode passif, 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.
<b>TYPE</b>	Cette commande permet de préciser le type de format dans lequel les données seront envoyées
<b>STRU</b>	Caractère Telnet précisant la structure du fichier (F pour <i>File</i> , R pour <i>Record</i> , P pour <i>Page</i> )
<b>MODE</b>	Caractère Telnet précisant le mode de transfert des données (S pour <i>Stream</i> , B pour <i>Block</i> , C pour <i>Compressed</i> )

## Protocole FTP

### c) Commande de service FTP :

Commande	Description
<b>RETR</b>	Cette commande ( <i>RETRIEVE</i> = Récupérer) demande au serveur DTP une copie du fichier dont le chemin d'accès est passé en paramètre.
<b>STOR</b>	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
<b>STOU</b>	Cette commande est identique à la précédente, si ce n'est qu'elle demande au serveur de créer un fichier dont le nom est unique. Le nom du fichier est retourné dans la réponse
<b>APPE</b>	Grâce à cette commande ( <i>append</i> = ajouter à = attacher à) les données envoyées sont concaténées dans le fichier portant le nom passé en paramètre s'il existe déjà, dans le cas contraire il est créé
<b>ALLO</b>	Cette commande ( <i>allocate</i> ) demande au serveur de prévoir un espace de stockage suffisant pour contenir le fichier dont le nom est passé en argument.
<b>REST</b>	Cette commande ( <i>restart</i> ) permet de reprendre un transfert là où il s'était arrêté. Pour cela cette commande envoie en paramètre le marqueur représentant la position dans le fichier à laquelle le transfert avait été interrompu. Cette commande doit être immédiatement suivie d'une commande de transfert.

## Protocole FTP

### c) Commande de service FTP :

<b>RNFR</b>	Cette commande ( <i>rename from</i> ) permet de renommer un fichier. Elle indique en paramètre le nom du fichier à renommer et doit être immédiatement suivie de la commande <i>RNTO</i>
<b>RNTO</b>	Cette commande ( <i>rename to</i> ) permet de renommer un fichier. Elle indique en paramètre le nom du fichier à renommer et doit être immédiatement précédée de la commande <i>RNFR</i>
<b>ABOR</b>	Cette commande ( <i>abort</i> ) indique au serveur DTP d'abandonner tous les transferts associés à la commande précédente. Si aucune connexion de données n'est ouverte, le serveur DTP ne fait rien, sinon il la ferme. Le canal de contrôle reste par contre ouvert.
<b>DELE</b>	Cette commande ( <i>delete</i> ) permet de supprimer le fichier dont le nom est passé en paramètre. Cette commande est irrémédiable, seule une confirmation au niveau du client peut être faite.
<b>RMD</b>	Cette commande ( <i>remove directory</i> ) permet de supprimer un répertoire. Elle indique en paramètre le nom du répertoire à supprimer
<b>MKD</b>	Cette commande ( <i>make directory</i> ) permet de créer un répertoire. Elle indique en paramètre le nom du répertoire à créer
<b>PWD</b>	Cette commande ( <i>print working directory</i> ) permet de renvoyer le chemin complet du répertoire courant

## Protocole FTP

### c) Commande de service FTP :

<b>LIST</b>	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
<b>NLST</b>	Cette commande ( <i>name liste</i> ) permet d'envoyer la liste des fichiers et répertoires dans le répertoire courant
<b>SITE</b>	Cette commande ( <i>site parameters</i> ) permet au serveur de proposer des services spécifiques, non définis dans le protocole FTP
<b>SYST</b>	Cette commande ( <i>system</i> ) permet d'envoyer des informations sur le serveur distant
<b>STAT</b>	Cette commande ( <i>status</i> ) permet d'émettre l'état du serveur, par exemple pour connaître la progression d'un transfert en cours. Cette commande accepte en argument un chemin d'accès, elle retourne alors les mêmes informations que LIST mais sur le canal de contrôle
<b>HELP</b>	Cette commande permet de connaître l'ensemble des commandes comprises par le serveur. Les informations sont renvoyées sur le canal de contrôle
<b>NOOP</b>	Cette commande ( <i>no operations</i> ) sert uniquement à obtenir une commande OK du serveur. Elle peut servir uniquement pour ne pas être déconnecté après un temps d'inactivité trop élevé

## Protocole FTP

### Les réponses FTP :

- 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 caractères 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** (relative à chaque deuxième chiffre)

## Protocole FTP

### Premier chiffre :

Chiffre	Signification	Description
<b>1yz</b>	Réponse préliminaire positive	L'action demandée est en cours de réalisation, une seconde réponse doit être obtenue avant d'envoyer une <u>deuxième commande</u>
<b>2yz</b>	Réponse positive de réalisation	L'action demandée a été réalisée, une nouvelle commande peut être envoyée
<b>3yz</b>	Réponse intermédiaire positive	L'action demandée est temporairement suspendue. Des informations supplémentaires sont attendues de la part du client
<b>4yz</b>	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
<b>5yz</b>	Réponse négative permanente	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

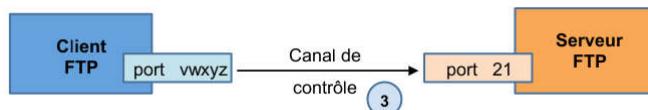
## Protocole FTP

### Second chiffre :

Chiffre	Signification	Description
<b>x0z</b>	Syntaxe	L'action possède une erreur de syntaxe, ou bien il s'agit d'une commande non comprise par le serveur
<b>x1z</b>	Information	Il s'agit d'une réponse renvoyant des informations (par exemple pour une réponse à une commande STAT)
<b>x2z</b>	Connexions	La réponse concerne le canal de données
<b>x3z</b>	Authentification et comptes	La réponse concerne le login (USER/PASS) ou la demande de changement de compte (CPT)
<b>x4z</b>	Non utilisé par le protocole FTP	
<b>x5z</b>	Système de fichiers	La réponse concerne le système de fichiers distant

## Exemples

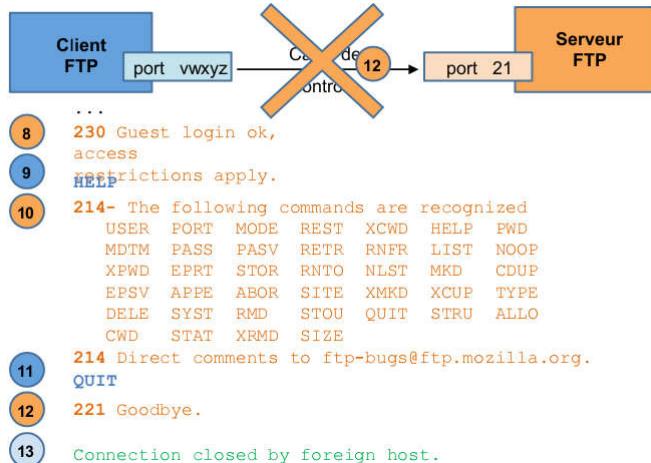
### FTP : canal de contrôle



- 1 Le serveur écoute sur le port 21
- 2 > telnet ftp.mozilla.org 21
- 3 Trying ftp.mozilla.org...  
Connected to ftp.mozilla.org.  
Escape character is '^]'.
- 4 220 mozilla FTP server (...) ready.
- 5 USER anonymous
- 6 331 Guest login ok, send your complete e-mail address as password.
- 7 PASS bonjour@bonjour.com
- 8 230 Guest login ok, access restrictions apply.

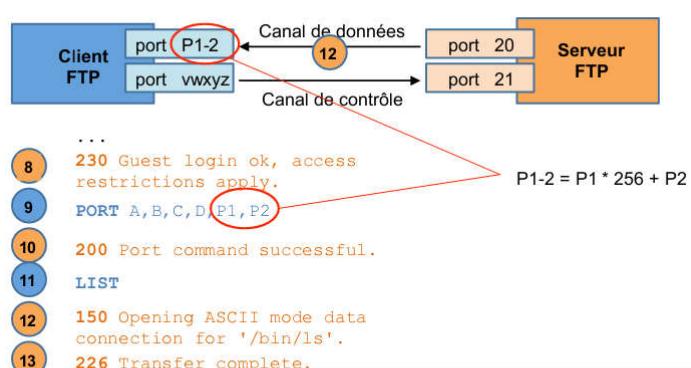
## Exemples

### FTP : commandes



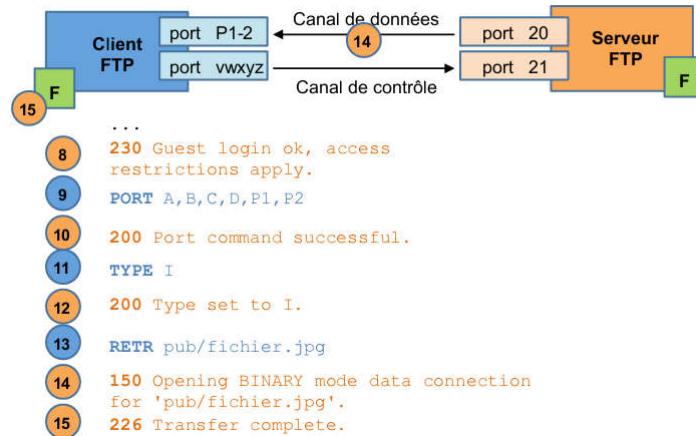
## Exemples

### FTP : lister contenu de dossier



## Exemples

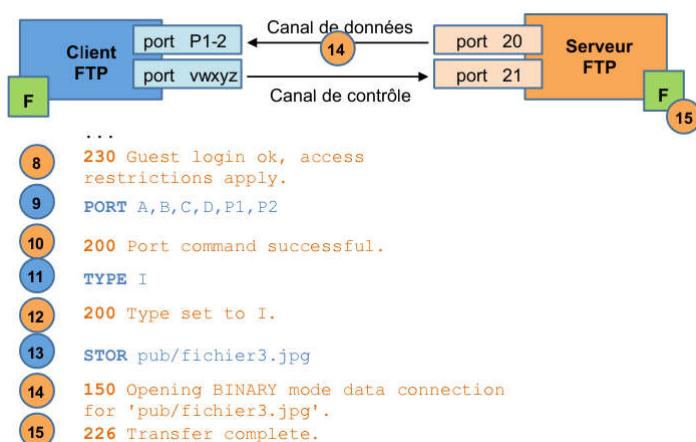
### FTP : réception de fichier



Dr. H. Zerrouki

## Exemples

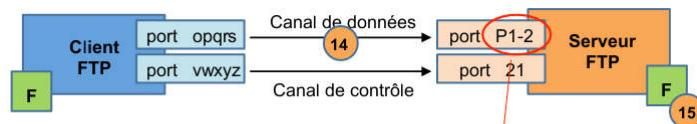
### FTP : envoi de fichier



Dr. H. Zerrouki

## Exemples

### FTP : passive mode



- 8 ...
- 9 230 Guest login ok, access restrictions apply.
- 10 PASV
- 11 227 Entering Passive Mode (A,B,C,D,P1,P2)
- 12 TYPE I
- 13 200 Type set to I.
- 14 STOR pub/fichier3.jpg
- 15 150 Opening BINARY mode data connection for 'pub/fichier3.jpg'.
- 16 226 Transfer complete.

**FIN**