FTP сървъри

Протокол FTP

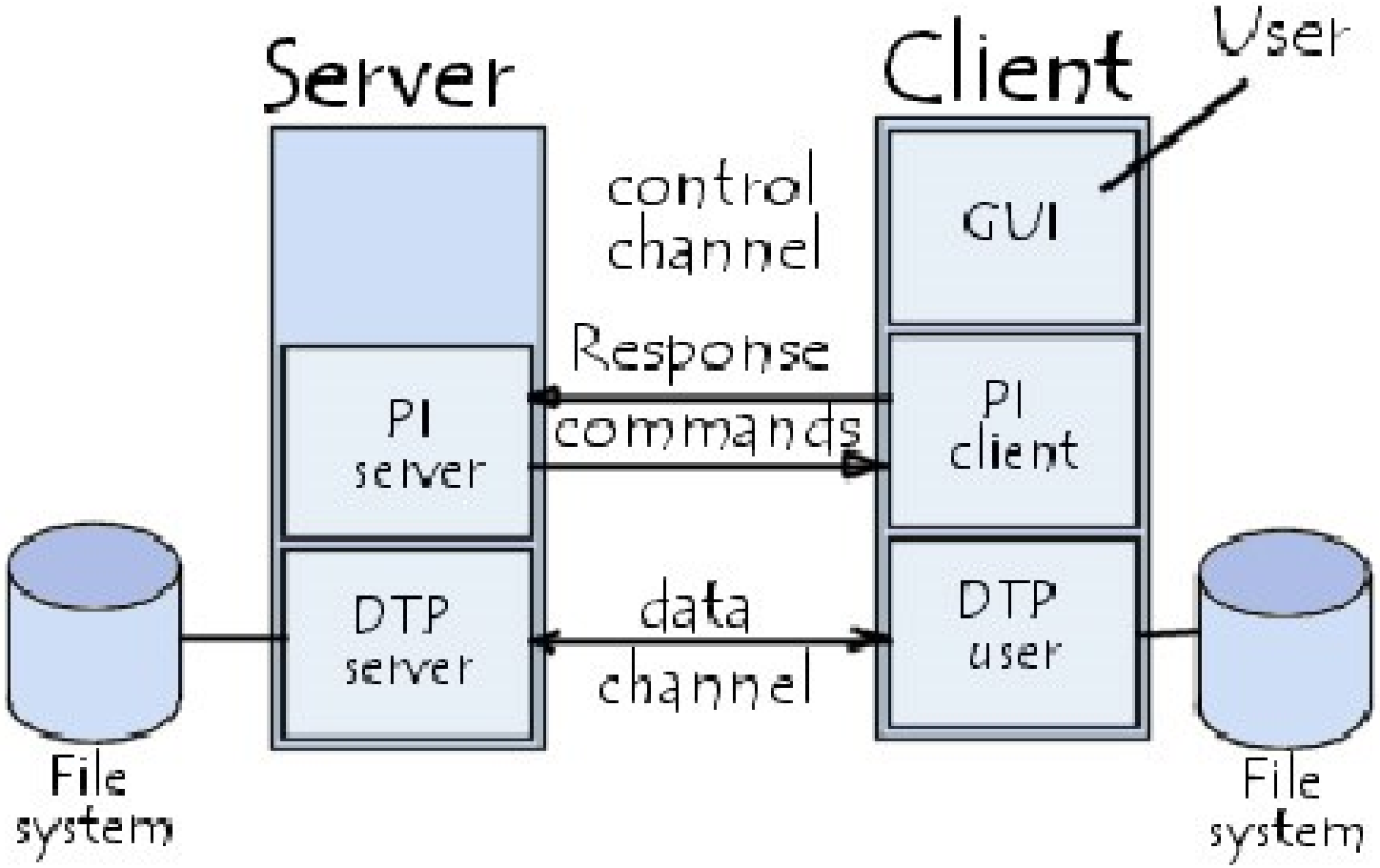
- FTP (File Transfer Protocol) е протокол за обмен на файлове
- Съвременният FTP протокол е дефиниран в RFC959.

Предназначение

FTP протоколът дефинира начина по който данните трябва да се обменят в TCP/IP мрежи. Основните цели на протокола са:

- позволява споделяне на файлове между отдалечени машини;
- позволява независимост между клиентските и сървърните системни файлове;
- позволява ефикасен обмен на данни.

Модел на FTP


Server
Client
User
control
channel
GUI
Response
commands
PI
server
PI
client
DTP
server
data
channel
DTP
user
File
system
File
system

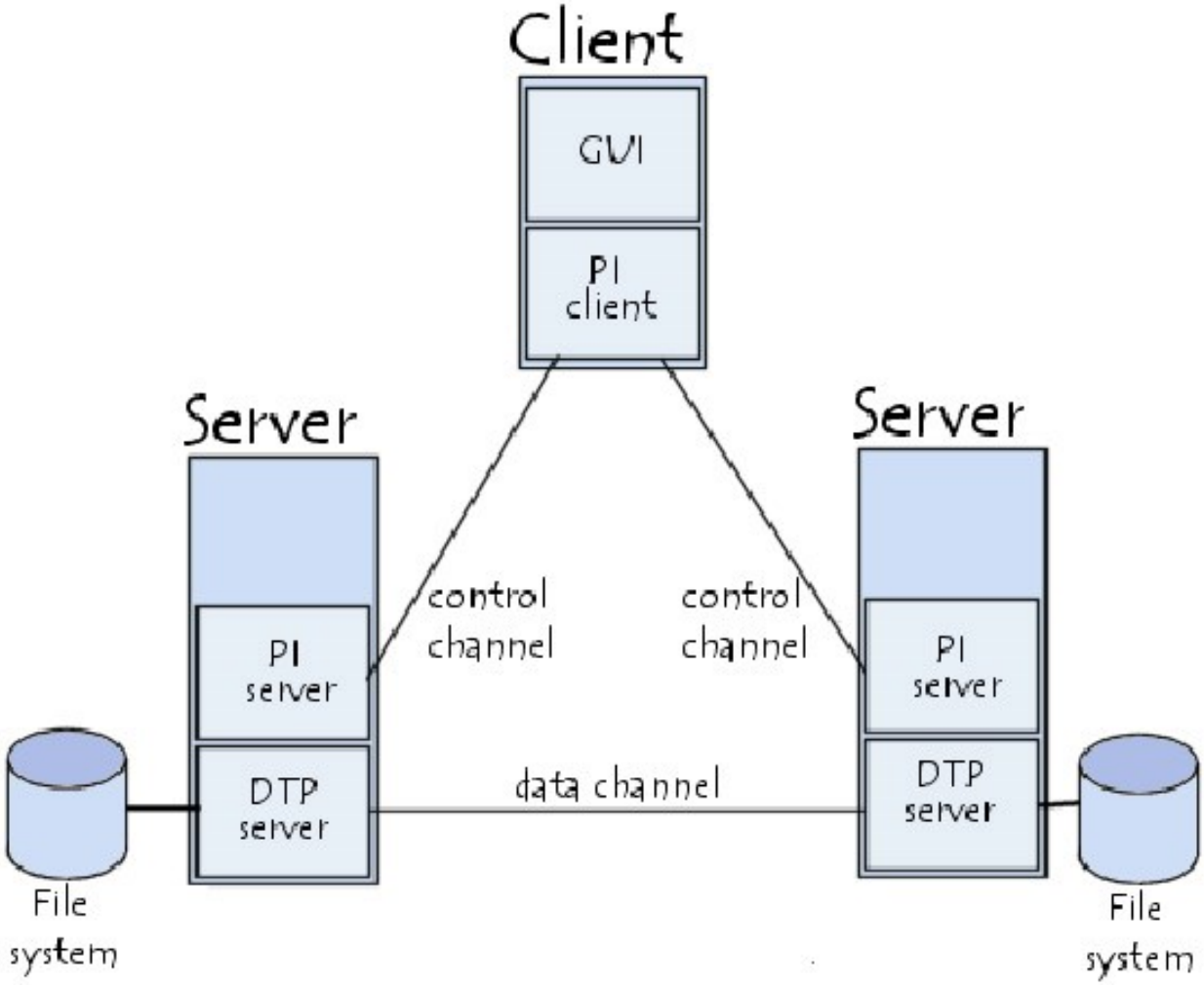
Компоненти

- DTP (Data Transfer Process) е процесът, на базата на който се създава връзката и се поддържа данновия канал. От страната на сървъра се нарича SERVER-DTP, а клиентската страна - USER-DTP;
- PI (Protocol Interpreter) интерпретира протокола, позволявайки DTP да бъде управляван чрез команди, получавани през контролния канал. Той се различава за клиента и сървъра.

Компоненти

- SERVER-PI е отговорен за очакването (Listening) на командите идващи от USER-PI през контролния канал; осъществяване на конекция за контролния канал, получаване на FTP команди от USER-PI през него; отговаряне на тези команди и стартиране на SERVER-DTP;
- USER-PI отговаря за създаването на конекция с FTP сървъра; изпращане на FTP команди; получаване на отговори от SERVER-PI и контролиране на USER-DTP при необходимост.

Между-сървърен обмен



FTP команди

Всички комуникации по контролния канал се базират на изискванията на TELNET протокола.

FTP командите реално са Telnet символни низове, завършващи с Telnet кода за край на ред <CR LF>.

Ако FTP команда има параметър, той се разделя от командата със ‘ ‘

Видове команди

- контролни команди за достъп;
- команди за параметри за обмен;
- команди за FTP услуги

Команди за достъп

Access control commands	
Command	**Description**
USER	Character string allowing the user to be identified. User identification is necessary to establish communication over the data channel.
PASS	Character string specifying the user's password. This command must immediately precede the *USER* command. It falls to the client to hide the display of this command for security reasons.
ACCT	Character string representing the user's account. The command is generally not necessary. During the response accepting the password, if the response is 230 this stage is not necessary, if the response is 332, it is.
CWD	*Change Working Directory*: this command enables the current directory to be changed. This command requires the directory's access path to be fulfilled as an argument.
CDUP	*Change to Parent Directory*: this command allows you to go back to the parent directory. It was introduced to solve problems of naming the parent directory according to the system (generally "...").
SMNT	*Structure Mount*:
REIN	*Reinitialize*:
QUIT	Command enabling the current session to be terminated. The server waits to finish the transfer in progress if the need arises, then supplies a response before closing the connection.

Команди за параметри за обмен

Transfer parameter commands	
Command	**Description**
PORT	Character string allowing the port number used to be specified.
PASV	Command making it possible to indicate to the DTP server to stand by for a connection on a specific port chosen randomly from among the available ports. The response to this command
STRU	Telnet character specifying the file structure (F for *File*, R for *Record*, P for *Page*).
MODE	Telnet character specifying data transfer method (S for *Stream*, B for *Block*, C for *Compressed*).

Команди за услуги

FTP service commands	
Command	**Description**
RETR	This command (*RETRIEVE*) asks the server DTP for a copy of the file whose access path is given in the parameters.
STOR	This command (*store*) asks the server DTP to accept the data sent over the data channel and store them in a file bearing the name given in the parameters. If the file does not exist, the server creates it, if not it overwrites it.
STOU	This command is identical to the previous one, only it asks the sever to create a file where the name is unique. The name of the file is returned in the response.
APPE	Thanks to this command (*append*) the data sent is concatenated into the file bearing the name given in the parameter if it already exists, if not, it is created.
ALLO	This command (*allocate*) asks the server to plan a storage space big enough to hold the file whose name is given in the argument.
REST	This command (*restart*) enables a transfer to be restarted from where it stopped. To do so, the command sends the marker representing the position in the file where the transfer had been interrupted in the parameter. This command must immediately follow a transfer command.
RNFR	This command (*rename from*) enables a file to be renamed. In the parameters it indicates the name of the file to be renamed and must be immediately followed by the *RNTO* command.

Команди за услуги

RNTO	This command (*rename to*) enables a file to be renamed. In the parameters it indicates the name of the file to be renamed and must be immediately followed by the *RNFR* command.
ABOR	This command (*abort*) tells the server DTP to abandon all transfers associated with the previous command. If no data connection is open, the DTP sever does nothing, if not it closes it. The control channel however remains open.
DELE	This command (*delete*) allows a file to be deleted, the name of which is given in the parameters. This command is irreversible, confirmation can only be given at client level.
RMD	This command (*remove directory*) enables a directory to be deleted. The name of the directory to be deleted is indicated in the parameters.
MKD	This command (*make directory*) causes a directory to be created. The name of the directory to be created is indicated in the parameters.
PWD	This command (*print working directory*) makes it possible to resend the complete current directory path.

Команди за услуги

LIST	This command allows the list of files and directories present in the current directory to be resent. This is sent over the passive DTP. It is possible to place a directory name in the parameter of this command, the server DTP will send the list of files in the directory placed in the parameter.
NLST	This command (*name list*) enables the list of files and directories present in the current directory to be sent.
SITE	This command (*site parameters*) causes the server to offer specific services not defined in the FTP protocol.
SYST	This command (*system*) allows information on the remote server to be sent.
STAT	This command (*status*) makes it possible to transmit the status of the server, for example to know the progress of a current transfer. This command accepts an access path in the argument, it then returns the same information as LIST but over the control channel.
HELP	This command gives all the commands understood by the server. The information is returned on the control channel.
NOOP	This command (*no operations*) is only used to obtain an OK command from the server. It can only be used in order not to be disconnected after an excessive period of inactivity.

FTP отговори

Чрез отговорите се осигурява синхронизация между клиента и сървъра. За всяка команда, изпратена от клиента, сървърът потенциално ще изпълни някакво действие и ще върне обратно отговор.

FTP отговори

Отговорите са формирани от 3 цифрен код, указващ начина по който изпратената от клиента команда е изпълнена. Реално се изпраща и текст (Telnet символен низ, разделен с ‘ ‘ от кода).

- първото число показва статуса на отговора (успех или неуспех);
- второто число указва за какво се отнася отговора;
- третото дава повече специфична информация (относно второто).

FTP отговори

First number		
Digit	**Meaning**	**Description**
1yz	Preliminary positive response	The action requested is in progress, a second response must be obtained before sending a second command
2yz	Positive fulfilment response	The action requested has been fulfilled, a new command can be sent
3yz	Intermediary positive response	The action request is temporarily suspended. Additional information is awaited from the client
4yz	Negative fulfilment response	The action requested has not taken place because the command has temporarily not been accepted. The client is requested to try again later
5yz	Permanent negative response	The action requested has not taken place because the command has not been accepted. The client is requested to formulate a different request

FTP отговори

Digit	Meaning	Description
x**0**z	Syntax	The action has a syntax error, or is a command not understood by the server
x**1**z	Information	This is a response sending back information (for example a response to a STAT command)
x**2**z	Connections	The response relates to the data channel
x**3**z	Authentication and accounts	The response relates to the (USER/PASS) login or the request to change the account (CPT)
x**4**z	Not used by the FTP protocol	
x**5**z	File system	The response relates to the remote file system

FTP сесия

Нормално, FTP сървърът очаква конекция на порт 21 (за контролния канал), докато порт 20 е за данновия канал.

След като клиентът се е свързал, сървърът изпраща поздравително съобщение от вида:

```
220  Some FTP server. Please login:
```

FTP сесия

```
220 SpiderMan's FTP server. Please login!
USER SpiderMan
331 Username okay. Send password!
PASS password
230 Password accepted, user logged in.
LIST
150 Opening ASCII mode data connection for /bin/ls
226 Transfer complete
TYPE I
200 Type set to I
PASV
227 Entering passive mode (206,84,161,87,28,46)
RETR datafile.zip
150 Opening BINARY mode data connection for datafile.zip
226 Transfer complete
```

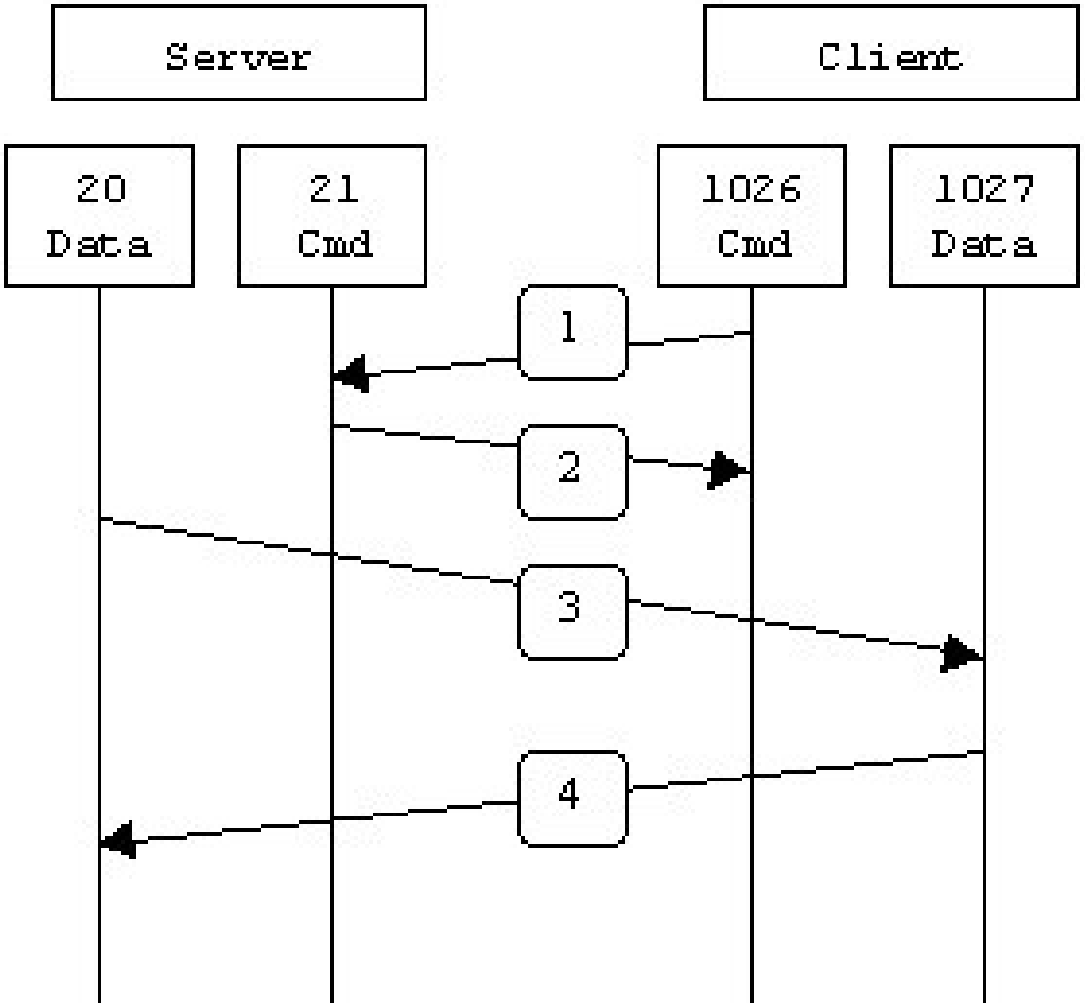
Активен режим на конекция

В активен режим клиентът се свързва от случаен непривилигерован порт (N>1023) към сървърния порт за команди 21.

След това клиентът започва за слуша на порт N+1 и изпраща командата PORT N+1 към сървъра.

Сървърът ще се свърже обратно към указания клиентски порт от неговия си локален порт – 20.

Активен режим на конекция



Firewall при активен режим - клиент

Главният проблем при активен FTP режим е от страна на клиента. Клиентът реално не осъществява актуална конекция към данновия порт на сървъра - той просто указва на сървъра на кой порт очаква конекция и сървърът се свързва към указания от клиента порт.

От гледна точка на firewall от страна на клиента това е конекция отвън към вътрешен клиент- нормално такива конекции се блокират (Windows Firewall).

Firewall при активен режим - сървър

От гледна точка на firewall от страна на сървъра, за поддържането на активен FTP режим е необходимо да бъдат отворени следните комуникационни канали:

- FTP сървърен порт 21 от всякъде (Клиентът инициира конекцията);
- FTP сървърен порт 21 към портове > 1023 (Сървърът отговаря на клиентския контролен порт);
- FTP сървърен порт 20 към портове > 1023 (Сървърът инициира даннова конекция към клиентския даннов порт);
- FTP сървърен порт 20 от портове >1023 (Клиентът изпраща потвърждения към сървърния даннов порт).

Активен режим - пример

```
testbox1: {/home/p-t/slacker/public_html} % ftp -d testbox2
Connected to testbox2.slacksite.com.
220 testbox2.slacksite.com FTP server ready.
Name (testbox2:slacker): slacker
---> USER slacker
331 Password required for slacker.
Password: TmpPass
---> PASS XXXX
230 User slacker logged in.
---> SYST
215 UNIX Type: L8
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
ftp: setsockopt (ignored): Permission denied
---> PORT 192,168,150,80,14,178
200 PORT command successful.
---> LIST
150 Opening ASCII mode data connection for file list.
drwx------   3 slacker    users         104 Jul 27 01:45 public_html
226 Transfer complete.
ftp> quit
---> QUIT
221 Goodbye.
```

Указване на портове

Когато се изпраща командата PORT се указва порта на клиента (192.168.150.80) - първите четири байта, разделени със запетая. Последните два байта формират порта на клиента, по който той ще слуша.

Актуалния порт се формира като:

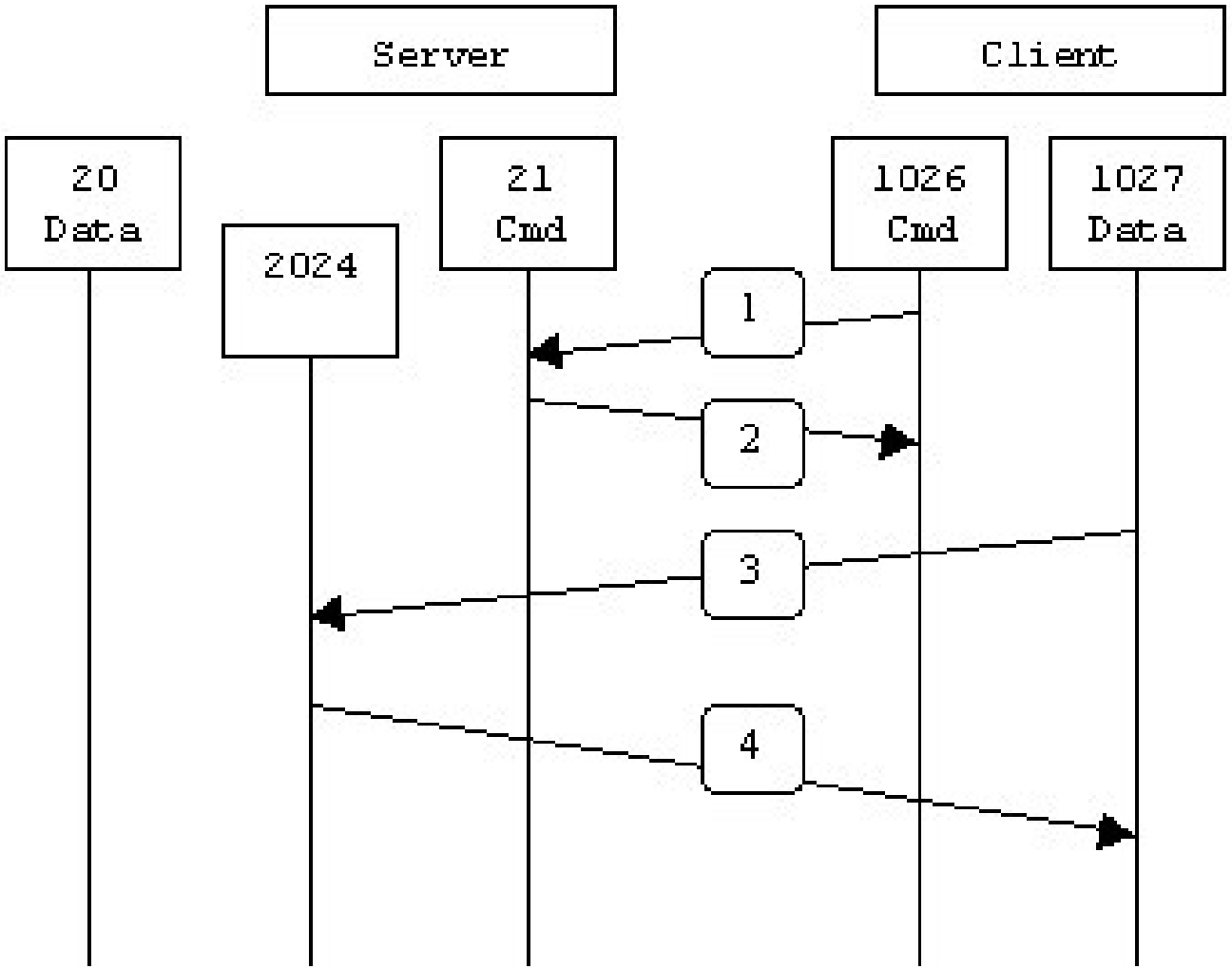
(14 * 256) + 178 = 3762

Пасивен режим на конекции

В този режим клиентът инициира и двете конекции към сървъра. Когато отваря FTP конекция, клиентът отваря 2 случайни непривилегировани порта локално (N>1023 и N+1). Първия порт се използва за свързване към сървъра на порт 21, но вместо команда PORT, клиентът издава команда PASV.

Като резултат сървърът отваря случаен непривилегирован порт (P>1023) и изпраща обратно към клиента команда PORT P. Клиентът инициира конекция от порт N+1 към сървърния порт P за обмен на данните.

Пасивен режим на конекции



Firewall при пасивен режим - сървър

От гледна точка на firewall от страна на сървъра, за поддържане на пасивен режим е необходимо да бъдат отворени следните комуникационни канали:

- FTP сървърен порт 21 от всякъде (Клиентът инициира конекцията);
- FTP сървърен порт 21 към портове > 1023 (Сървърът отговаря към клиентския контролен порт);
- FTP сървърен портове > 1023 от всякъде (Клиентът инициира даннова конекция към случаен порт на сървъра);
- FTP сървърни портове > 1023 към отдалечени портове > 1023 (Сървърът изпраща ACK и данни към клиентските портове).

Пасивен режим - пример

```
testbox1: {/home/p-t/slacker/public_html} % ftp -d testbox2
Connected to testbox2.slacksite.com.
220 testbox2.slacksite.com FTP server ready.
Name (testbox2:slacker): slacker
---> USER slacker
331 Password required for slacker.
Password: TmpPass
---> PASS XXXX
230 User slacker logged in.
---> SYST
215 UNIX Type: L8
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> passive
Passive mode on.
ftp> ls
ftp: setsockopt (ignored): Permission denied
---> PASV
227 Entering Passive Mode (192,168,150,90,195,149).
---> LIST
150 Opening ASCII mode data connection for file list
drwx------   3 slacker    users         104 Jul 27 01:45 public_html
226 Transfer complete.
ftp> quit
---> QUIT
221 Goodbye.
```

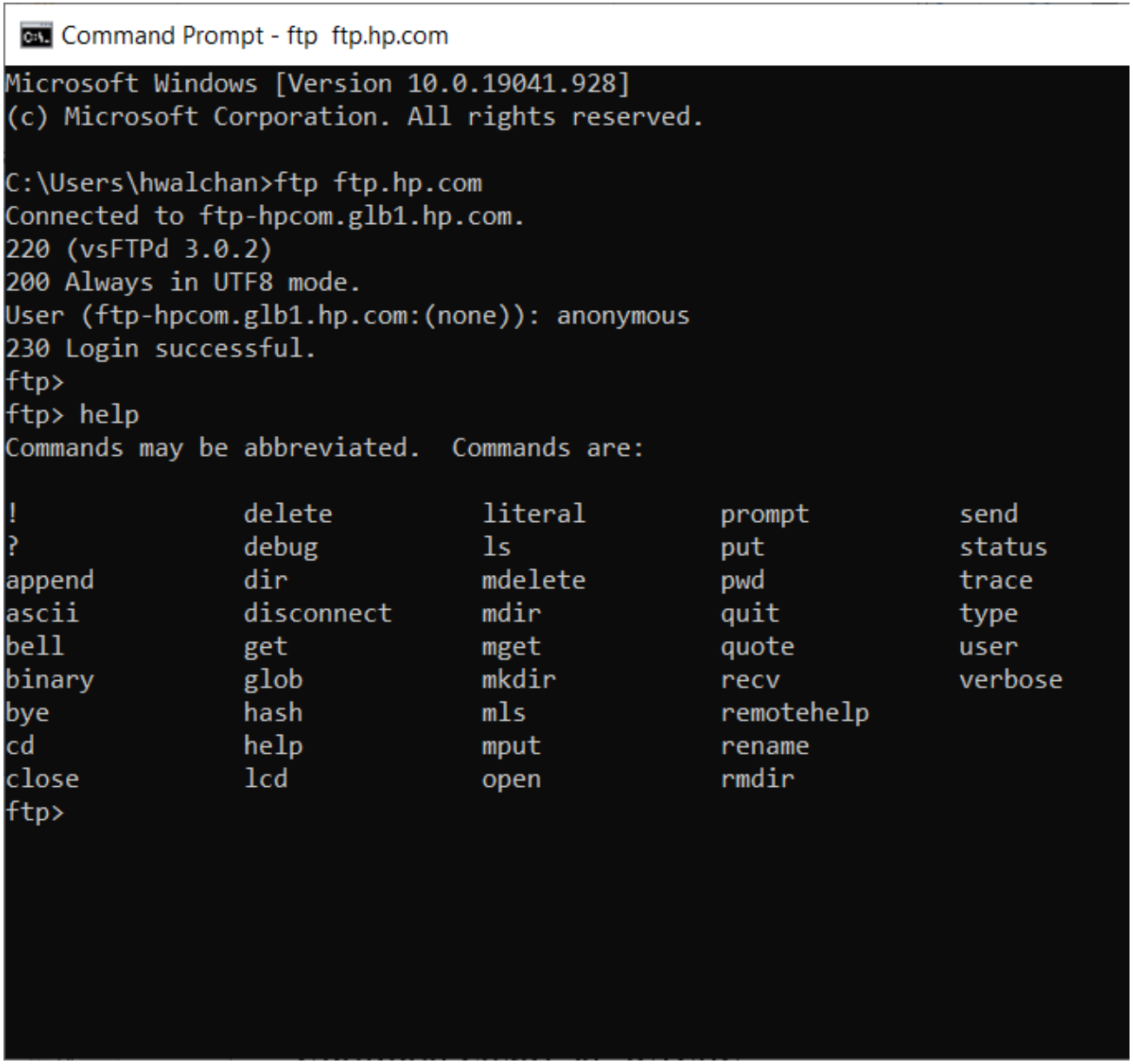
Анонимни FTP сървъри

Позволяват достъп на нерегистрирани потребители до файлове на сървъра.

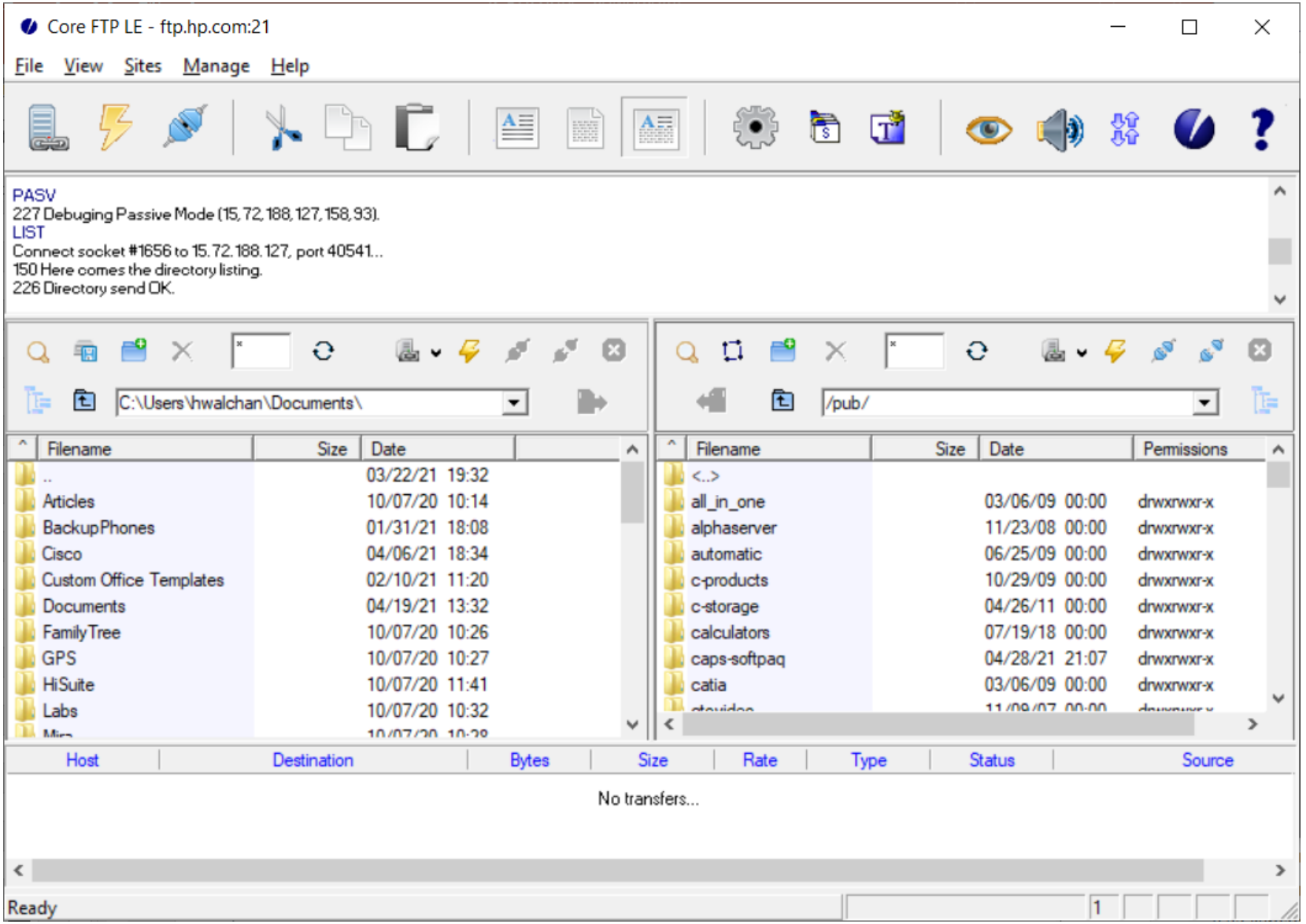
Изисквания:

- Сървърът трябва да е конфигуриран да поддържа анонимен акаунт.
- Клиентът се логва с името **anonymous** или **ftp**.
- Може да не се въвежда парола или да се изисква въвеждане на валиден е-мейл адрес (в зависимост от конфигурирането на сървъра).
- Клиентът има достъп единствено до директория **/pub** на сървъра само за четене.

Анонимни FTP сървъри



Анонимни FTP сървъри



Конфигуриране на FTP сървър

Сървър под Linux - **vsftpd** (Very Secure FTP daemon)

Конфигурационният му файл е в */etc/vsftpd.conf*.

Конфигуриране на FTP сървър

Разрешаване на **vsftpd** в конфигурацията на супер-сървъра.

В конфигурационния файл */etc/inetd.conf* трябва да се махне коментара в реда за **vsftpd**.

Рестартиране на супер-сървъра: **killall –HUP inetd**

```
# discard       dgram   udp     wait    root    internal
# daytime       stream  tcp     nowait  root    internal
# daytime       dgram   udp     wait    root    internal
# chargen       stream  tcp     nowait  root    internal
# chargen       dgram   udp     wait    root    internal
# time          stream  tcp     nowait  root    internal
# time          dgram   udp     wait    root    internal
#
# These are standard services.
#
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  vsftpd
#ftp    stream  tcp     nowait  root    /usr/sbin/tcpd  proftpd
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
#
# telnet stream  tcp     nowait  root    /usr/sbin/tcpd
/usr/sbin/in.telnetd
#
```

Конфигуриране на FTP сървър

1. Разрешаване на **анонимен акаунт.**

В конфигурационния файл */etc/vsftpd.conf* трябва да се промени директивата:

```
anonymous_enable=YES
```

2. Създаване на аномимен потребител

Ако няма **директория** */home/ftp*, трябва да се създаде потребител **ftp** без парола:

```
useradd ftp
```

Директорията */home/ftp* е публичната директория за достъп.

Въпроси ?