

Protocollo FTP (File Transfer Protocol)

Il protocollo FTP viene utilizzato per il trasferimento dei file, sfruttando la classica connessione client-server. Le caratteristiche di questo protocollo, ormai molto datato, sono definite dalla RFC959.

Una piccola parentesi

L'**IETF** (Internet Engineering Task Force) è un organismo internazionale che si occupa di redigere delle norme tecniche per standardizzare il mondo delle telecomunicazioni e di Internet. L'ente composto da tecnici ed esperti programmatori, lavora suddiviso in gruppo preparando dei documenti definiti RFC (**R**equ**E**st **F**or **C**omments) validati poi da un altro ente certificatore denominato **IESG** (Internet Engineering Steering Group) quest'ultimo provvedere a promuovere questi documenti come standard ufficiali a livello mondiale.

Tutti gli standard ed i documenti prodotti sono open source e disponibili gratuitamente.

Per effettuare la connessione tra client e server, occorre conoscere il suo indirizzo IP o il suo URL (che verrà poi comunque risolto da un DNS). La connessione può avvenire tramite programmi eseguiti da CLI o tramite interfacce grafiche facilmente utilizzabili come ad esempio Filezilla, uno dei client (anche server) più diffusi liberamente scaricabile ed utilizzabile.

A prescindere dal tipo di client utilizzato, la connessione può avvenire in due modalità:

- FTP Anonymous
- FTP con account

FTP Anonymous

Questa modalità viene utilizzata per prelevare dei file ad accesso pubblico da servers che li mettono a disposizione liberamente, come ad esempio Università, enti o società varie. In questo modo si accede al server che contiene i file con un client, senza dover inserire nome utente e password. Per fare questo è sufficiente digitare la parola **Anonymous** quando viene richiesto lo username, come password possiamo non inserire nulla, ma è consuetudine indicare nel campo password il proprio indirizzo di posta elettronica, solo a titolo documentativo.

FTP con account

In questo caso oltre che conoscere l'indirizzo o il nome del server FTP, occorre disporre delle credenziali valide per accedere ai file condivisi (nome utente e password). Una volta entrato l'utente potrà scaricare liberamente i file del server o aggiungerne altri come se fosse una directory del proprio hard disk.

Protocollo TFTP (Trivial File Transfer Protocol)

A pochi anni dalla nascita dell'FTP, viene rilasciata la **RFC1350** per definire il nuovo protocollo TFTP come versione più leggera del precedente. In questo caso infatti il livello trasporto utilizza il protocollo **UDP** anziché il **TCP**, ed utilizza la porta **69**. Il TFTP viene utilizzato soprattutto per il trasferimento dei file nelle reti locali in quanto non prevede né autenticazione né cifratura.

Connessione FTP

Il protocollo FTP utilizza come mezzo di trasporto il protocollo TCP, il trasferimento dei file avviene instaurando due connessioni TCP su due canali differenti, uno definito "canale di controllo" ed uno definito "canale dati". Sono due processi differenti definiti **PI (Protocol Interpreter)** utilizzato per trasferire i comandi dal client al server utilizzando la porta 21, e **DTP (Data Transfer Protocol)** utilizzato per trasferire i dati utilizzando la porta 20.



Esistono due modalità di collegamento, la modalità **normale** (normal mode) e quella **passiva** (passive mode).

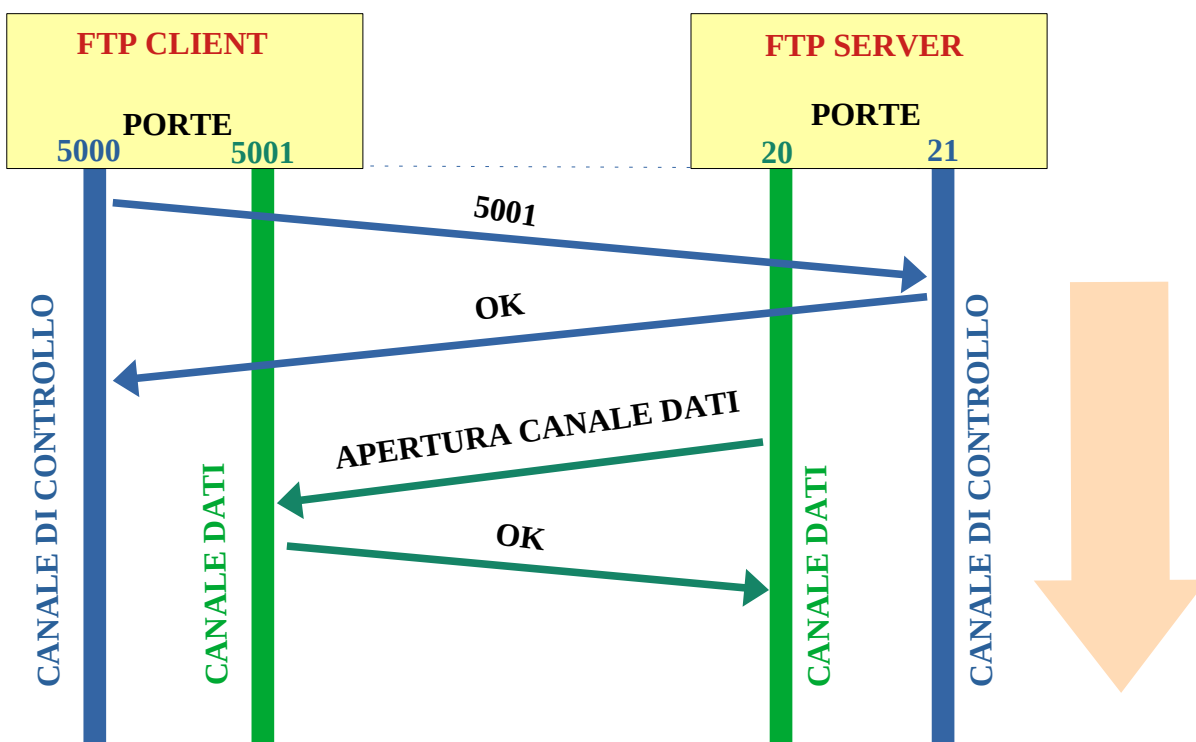
Nel primo caso il canale di controllo viene aperto dal client e quello dati dal server, nel secondo caso invece il client ha la possibilità di aprire sia il canale di controllo che quello dati.

NORMAL MODE

Il client apre il canale di controllo tra una propria porta ad esempio la 5000 e la 21 del server, indicando la porta per il trasferimento dati ad esempio la 5001. Il server risponde sul canale di controllo con un OK ed apre la connessione tra la sua porta 20 e quella indicata dal server 5001. Il client risponde inviando al server un segnale di OK. La connessione ora è pronta per il trasferimento dei file.

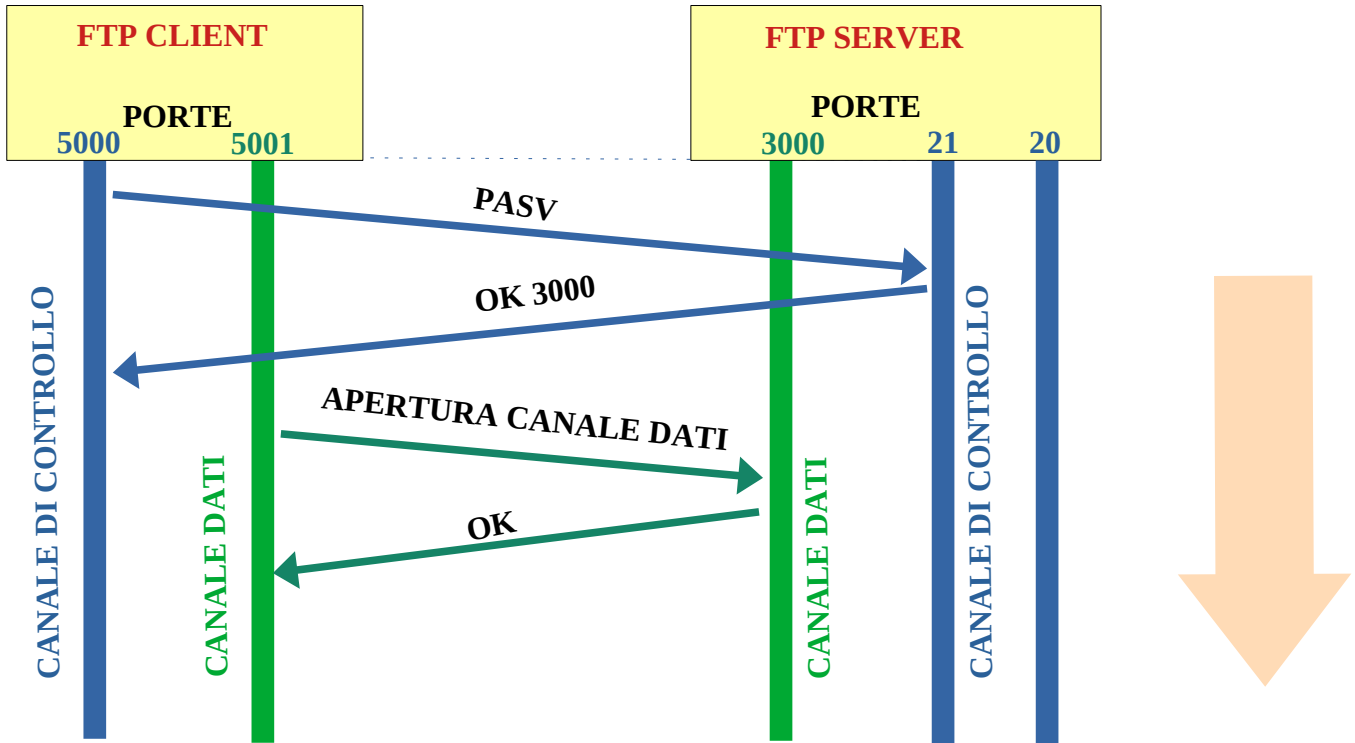
Una piccola parentesi

Le porte fino a 1023 sono utilizzate per convenzione dai server. Questa regola è stata introdotta dallo Unix e rispettata poi da altri sistemi operativi come Windows.



PASSIVE MODE

Il client riserva due porte (numero maggiore di 1023) ad esempio 5000 per il controllo e 5001 per i dati ed invia il comando PASV sul canale di controllo. Alla ricezione del comando, il server invia un segnale di OK con il quale indica una porta diversa dalla 20 su cui aprire il canale (sempre maggiore di 1023) ad esempio la porta 3000. Il client a questo punto apre il canale dati tra la porta 5001 e la 3000, il server risponde con OK, a questo punto la connessione è pronta.



Il protocollo FTP non prevede alcuna cifratura, pertanto i dati scambiati tra client e server possono essere intercettati e letti. Ad esempio possiamo provare ad utilizzare **Wireshark** mettendo come filtro di visualizzazione **ftp || ftp-data** (ftp oppure ftp-data). In questo modo potremmo catturare i pacchetti scambiati tra il nostro PC ed un server FTP libero come ad esempio ftp.gnu.org.

Per fare questo è sufficiente aprire il prompt dei comandi e digitare il comando **ftp**, successivamente aprire il canale con il comando **open ftp.gnu.org**.

```
ftp> open ftp.gnu.org
Connesso a ftp.gnu.org.
220 GNU FTP server ready.
200 Always in UTF8 mode.
Utente (ftp.gnu.org:(none)): Anonymous
230-NOTICE (Updated October 15 2021):
```

successivamente dopo aver digitato il nome dell'utente **Anonymous**, apparirà la scritta **Login successful**

da questo momento in poi è possibile digitare i comandi, come ad esempio il comando **ls**, utilizzato per vedere l'elenco dei file e delle cartelle presenti sul server

```
230 Login successful.
ftp> ls
```

Nel frattempo sulla finestra di cattura di Wireshark, appariranno tutti i pacchetti ftp che sono transistati in uscita ed in ingresso dal nostro Pc.

Time	Source	Destination	Protocol	Length	Info
1878	11.313890	209.51.188.20	192.168.1.25	FTP	81 Response: 220 GNU FTP server ready.
1879	11.325476	192.168.1.25	209.51.188.20	FTP	68 Request: OPTS UTF8 ON
1884	11.444391	209.51.188.20	192.168.1.25	FTP	80 Response: 200 Always in UTF8 mode.
3685	23.249843	192.168.1.25	209.51.188.20	FTP	70 Request: USER Anonymous
3687	23.468701	209.51.188.20	192.168.1.25	FTP	93 Response: 230-NOTICE (Updated October 15 2021):
3688	23.469238	209.51.188.20	192.168.1.25	FTP	60 Response: 230-
3690	23.469520	209.51.188.20	192.168.1.25	FTP	120 Response: 230-If you maintain scripts used to access ftp.gnu.org over FTP,
3691	23.469611	209.51.188.20	192.168.1.25	FTP	122 Response: 230-we strongly encourage you to change them to use HTTPS instead.
3693	23.470278	209.51.188.20	192.168.1.25	FTP	60 Response: 230-
3694	23.470278	209.51.188.20	192.168.1.25	FTP	121 Response: 230-Eventually we hope to shut down FTP protocol access, but plan
3696	23.470555	209.51.188.20	192.168.1.25	FTP	121 Response: 230-to give notice here and other places for several months ahead
3697	23.471607	209.51.188.20	192.168.1.25	FTP	68 Response: 230-of time.
3698	23.471607	209.51.188.20	192.168.1.25	FTP	60 Response: 230-
3699	23.471607	209.51.188.20	192.168.1.25	FTP	63 Response: 230----
3705	23.586447	209.51.188.20	192.168.1.25	FTP	723 Response: 230-
4096	30.851012	192.168.1.25	209.51.188.20	FTP	80 Request: PORT 192,168,1,25,195,49
4102	30.969310	209.51.188.20	192.168.1.25	FTP	105 Response: 200 PORT command successful. Consider using PASV.
4103	30.978530	192.168.1.25	209.51.188.20	FTP	60 Request: NLST
4108	31.214896	209.51.188.20	192.168.1.25	FTP	93 Response: 150 Here comes the directory listing.
4109	31.215061	209.51.188.20	192.168.1.25	FTP-DATA	292 FTP Data: 238 bytes (PORT) (NLST)
4114	31.333189	209.51.188.20	192.168.1.25	FTP	78 Response: 226 Directory send OK.
8320	331.333373	209.51.188.20	192.168.1.25	FTP	68 Response: 421 Timeout.

Possiamo aprirne qualcuno, ad esempio i due indicati in figura e vedere che le informazioni sono in chiaro. Nel primo pacchetto c'è il nome utente digitato Anonymous e nel secondo c'è il risultato del comando **ls**, cioè la lista dei file presenti sul server. Il problema della cifratura è stato risolto con la **RFC-4217** con cui si introduce il protocollo **FTPS** basato sulla cifratura **SSL** o **TSL**, operante sulle porte **989**(dati) e **990**(controllo).

Programmi client e server per FTP

E' possibile instaurare una connessione con client ftp in due modi, il primo da CLI (prompt dei comandi) digitando semplicemente ftp, e successivamente indicando i comandi per aprire la connessione e trasferire i file come visto sopra. Oppure in maniera molto più semplice si possono installare programmi come il noto filezilla, che presentano un interfaccia grafica di facile utilizzo.

The screenshot shows the FileZilla client interface. At the top, the status bar indicates the connection: 'sftp://pi@192.168.1.21 - FileZilla'. Below this, the connection details are visible: Host: 192.168.1.21, Nome utente: pi, Password: [masked], Porta: 22. The interface is split into two main panes. The left pane shows the local site 'C:\Users\daniele\Desktop\'. The right pane shows the remote site '/home/pi'. Below these panes, there are two lists of files. The left list shows local files like 'Apocalypso', 'FTP', 'prova', etc. The right list shows remote files like '.cache', '.config', '.gnupg', etc. The bottom status bar indicates '25 file e 7 cartelle. Dimensione totale: 31.890.390.509 byte' for the local site and '14 file e 32 cartelle. Dimensione totale: 152.558.797 byte' for the remote site.

In questo caso tramite filezilla, ci siamo connessi alla scheda Raspberry digitando l'indirizzo, il nome utente e la password, ed inserendo la porta 22 utilizzata per la cifratura SSL. In basso a sinistra abbiamo le cartelle del nostro PC invece a sinistra quelle della scheda Raspberry.

Utilizzando invece il client da CLI, dovremo digitare i comandi previsti dal client FTP, di seguito l'elenco dei comandi possibili:

COMANDI DI ACCESSO	
open <host>	<i>Apri una connessione verso un host</i>
user <nomeutente>	Identifica il nome utente
pass <password>	Invia la password
quit	Chiude la sessione ftp
comandi di trasferimento	
port <indirizzo porta>	Definisce la porta del client per il canale dati
pasv	Attiva e disattiva la modalità passiva
type	Indica il tipo di trasferimento dei dati (binario o ascii)
bin	Imposta la trasmissione in modalità binaria, per trasferire tutti i file che non contengono caratteri ascii (es. programmi, immagini, documenti word, ecc..)
ascii	Imposta la trasmissione in modalità ascii
get <file da remoto> < file in locale>	Trasferisce il file dal server remoto alla posizione indicata nel pc
get <nome file>	Visualizza un file a video
mget <file da remoto> < file in locale>	Trasferisce più file dal server remoto alla posizione indicata nel pc. Ad esempio mget *.txt trasferisce tutti i file che hanno estensione txt
put <file da remoto> < file in locale>	Trasferisce il file dal proprio pc al server remoto.
mput <file da remoto> < file in locale>	Trasferisce più file dal proprio pc al server remoto.
prompt	Abilita e disabilita la conferma per ogni file durante i trasferimenti multipli
comandi di gestione dei file	
delete <nome file>	Cancella un file sul server FTP
mdelete <nome files>	Cancella più file sul server FTP
cd <nome directory>	Cambia directory
cpdu oppure cd ..	Passa alla directory di livello superiore
pwd	Visualizza la directory attiva
mkdir <nome directory>	Crea una nuova directory
rmdir <dir1> <dir2>	Rinomina una directory
ls o dir	Visualizza il contenuto della directory
rename <file1> <file2>	Rinomina il file 1 in file 2
comandi di aiuto	
help	Visualizza i comandi possibili
help <nome comando>	Fornisce un help sul comando singolo
!	Esce dalla shell mantenendo attivo il client FTP
status	Visualizza informazioni sulla sessione FTP corrente

IMPORTANTE !!!

I comandi sopra indicati sono quelli utilizzati dal client ftp su prompt dei comandi, ma non corrispondono ai comandi previsti nella RFC959 che definisce il funzionamento del protocollo FTP. Ad esempio il comando **GET** che trasferisce il file in realtà viene tradotto nel comando previsto dalla RFC che è **RETR**. Catturando i pacchetti con wireshark potremo vedere i comandi grezzi previsti dal protocollo FTP e definiti nella **RFC959**.

A questo link troviamo la RFC con tutti i comandi: <https://datatracker.ietf.org/doc/html/rfc959>

I codici di risposta sono invece composti da 3 cifre **xyz**, ognuna delle quali fornisce indicazioni delle operazioni in modo sempre più dettagliato.

- 1yz risposta positiva preliminare, il comando è stato accettato e ci sarà un'ulteriore risposta.
- 2yz comando terminato con successo.
- 3yz risposta intermedia positiva, in attesa di ulteriori informazioni per completare l'operazione.
- 4yz comando non eseguito correttamente
- 5yz comando non eseguito dal server

ESERCITAZIONE N.1

La prima esercitazione che possiamo fare è quella di utilizzare la **CLI** per comunicare con uno dei seguenti server ftp: ftp.gnu.org ftp.pureftpd.org

I due server sopracitati, consentono l'accesso Anonymous, pertanto il primo esercizio è quello di connettersi dal proprio PC e provare a scaricare un file testo dal server, elencando poi i comandi necessari per effettuare questa operazione analizzando i pacchetti catturati con wireshark.

es.

trasferimento del file welcome.msg dal server ftp.gnu.org e relativa cattura su wireshark del pacchetto dati

210	22.421423	192.168.1.25	209.51.188.20	FTP	81 Request: PORT 192,168,1,25,228,232
212	22.542654	209.51.188.20	192.168.1.25	FTP	105 Response: 200 PORT command successful. Consider using PASV.
213	22.553005	192.168.1.25	209.51.188.20	FTP	72 Request: RETR welcome.msg
218	22.792555	209.51.188.20	192.168.1.25	FTP-DA...	1146 FTP Data 1092 bytes (PORT) (RETR welcome.msg)
221	22.795370	209.51.188.20	192.168.1.25	FTP	125 Response: 150 Opening BINARY mode data connection for welcome.msg (1092 bytes).
223	22.912992	209.51.188.20	192.168.1.25	FTP	78 Response: 226 Transfer complete.

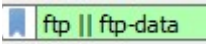
in questo caso si può vedere il comando FTP corrispondente al comando GET e cioè RETR.

Inoltre si può leggere il contenuto del file richiesto.

```
> Transmission Control Protocol, Src Port: 20, Dst Port: 58600, Seq: 1, Ack: 1, Len: 1092
  FTP Data (1092 bytes data)
    [Setup frame: 210]
    [Setup method: PORT]
    [Command: RETR welcome.msg]
    Command frame: 213
    [Current working directory: ]
  Line-based text data (29 lines)
    NOTICE (Updated October 15 2021):\n
    \n
    If you maintain scripts used to access ftp.gnu.org over FTP,\n
    we strongly encourage you to change them to use HTTPS instead.\n
    \n
0000  68 f7 28 b1 98 96 14 14 59 0c 57 b0 08 00 45 00  h·(·····Y·W···E·
0010  04 6c be 09 40 00 2f 06 3a 79 d1 33 bc 14 c0 a8  ·l·@·/·:·y·3····
0020  01 19 00 14 e4 e8 e6 13 e3 cb c6 89 ac 91 50 18  ······P·
0030  01 f6 a9 83 00 00 4e 4f 54 49 43 45 20 28 55 70  ······NO TICE (Up
0040  64 61 74 65 64 20 4f 63 74 6f 62 65 72 20 31 35  dated Oc tober 15
0050  20 32 30 32 31 29 3a 0a 0a 49 66 20 79 6f 75 20  2021):·If you
0060  6d 61 69 6e 74 61 69 6e 20 73 63 72 69 70 74 73  maintain scripts
0070  20 75 73 65 64 20 74 6f 20 61 63 63 65 73 73 2 used to access
0080  66 74 70 2e 67 6e 75 2e 6f 72 67 20 6f 76 65 72  ftp.gnu. org over
0090  20 46 54 50 2c 0a 77 65 20 73 74 72 6f 6e 67 6c  FTP, we strongl
00a0  79 20 65 6e 63 6f 75 72 61 67 65 20 79 6f 75 20  y encour age you
00b0  74 6f 20 63 68 61 6e 67 65 20 74 68 65 6d 20 74  to chang e them t
00c0  6f 20 75 73 65 20 48 54 54 50 53 20 69 6e 73 74  o use HT TPS inst
00d0  65 61 64 2e 0a 0a 45 76 65 6e 74 75 61 6c 6c 79  ead.·Ev entually
00e0  20 77 65 20 68 6f 70 65 20 74 6f 20 73 68 75 74  we hope to shut
00f0  20 64 6f 77 6e 20 46 54 50 20 70 72 6f 74 6f 63  down FT P protoc
0100  6f 6c 20 61 63 63 65 73 73 2c 20 62 75 74 20 70  ol acces s, but p
```


L'esercizio da realizzare nel dettaglio è il seguente:

1. connettersi al server ftp.gnu.org da CLI, catturando i pacchetti relativi a questa operazione con wireshark.
2. Fare lo screenshot del pacchetto dati di richiesta di connessione da parte del client identificando la porta utilizzata dal client e dal server, indicare nello screenshot le due porte.
3. fare lo screenshot del pacchetto contenente il comando di autenticazione in modo anonymous, ed evidenziare il comando FTP utilizzato per l'autenticazione dell'utente.
4. Leggere il contenuto del server ed identificare il pacchetto su wireshark dove viene inviato il comando relativo a questa richiesta, effettuare lo screenshot ed indicare il comando FTP.
5. Cambiare directory nel server ed identificare il pacchetto su wireshark dove viene inviato il comando relativo a questa richiesta, effettuare lo screenshot ed indicare il comando FTP.
6. Tornare alla directory precedente ed identificare il pacchetto su wireshark dove viene inviato il comando relativo a questa richiesta, effettuare lo screenshot ed indicare il comando FTP.
7. Copiare un file testo dal server al proprio pc ed identificare il pacchetto su wireshark dove viene inviato il comando relativo a questa richiesta, effettuare lo screenshot ed indicare il comando FTP.

N.B. Inserendo sul filtro di visualizzazione ftp e ftp-data  sarà possibile catturare i pacchetti che transitano sul canale dati e sul canale di controllo.

Identificare inoltre la porta utilizzata dal client e quella utilizzata dal server per il canale dati, effettuare lo screenshot delle due porte.

8. Effettuare lo screenshot su wireshark del pacchetto dati contenente il file richiesto.
9. Utilizzando wireshark identificare i comandi FTP corrispondenti ai seguenti comandi del client:
 - ascii
 - bin
 - get
 - mget
 - put (*non verrà consentito il trasferimento in modalità anonymous, ma il comando si può vedere*)
 - cd
 - cd ..
 - pwd
 - ls
 - quit

Inserendo sul filtro di visualizzazione ftp e ftp-data  sarà possibile catturare i pacchetti che transitano sul canale dati e sul canale di controllo.

Inserire quanto richiesto in un documento pdf.

Quanto sopra effettuato, ci ha consentito di acquisire dimestichezza con i comandi e di vedere lo scambio dei pacchetti su **wireshark**, di seguito invece utilizzeremo il programma **filezilla** per realizzare un server ed accedere da remoto con il client.

PROGRAMMI PER FTP

Uno dei programmi più utilizzati per il trasferimento dei file con il protocollo FTP, si chiama Filezilla ed è liberamente scaricabile dal sito: <https://filezilla-project.org/>

A questo link si può scaricare la versione client e server.

Con la versione client ci si può collegare in maniera estremamente semplice ad un server FTP, è sufficiente indicare l'indirizzo o l'url del server, il nome utente e l'eventuale password e la porta. Premendo il pulsante "connessione rapida" sulla finestra del programma apparirà a destra il contenuto del server, come visto precedentemente.

Ma oltre che nella versione client, filezilla ci consente di far diventare il nostro PC un server, attivando così il relativo demone in ascolto sulla porta 21 per rispondere alle richieste dei vari client.

Una volta installato filezilla in versione server, ci viene chiesta la porta e la password per accedere al server locale in modalità amministrazione collegandosi all'indirizzo 127.0.0.1 (localhost).

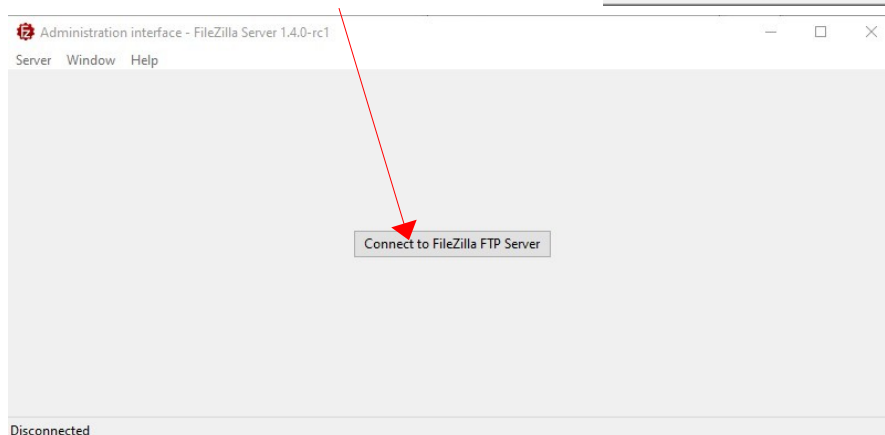
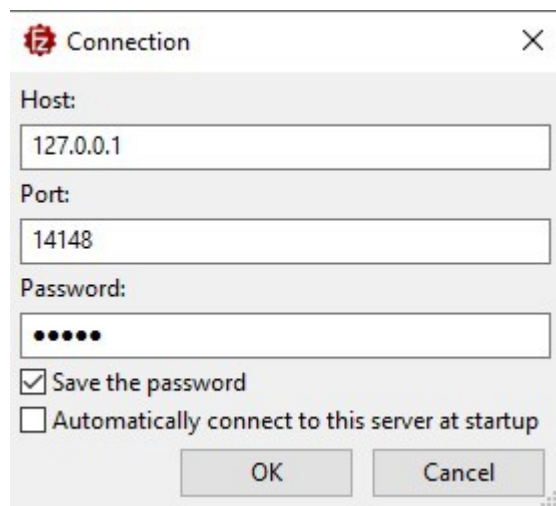
Una piccola parentesi

Con il termine LOCALHOST, si intende l'interfaccia di loopback, e cioè l'indirizzo che ci consente di dialogare con la propria macchina, come fosse una macchina connessa in rete e raggiungibile da remoto tramite l'indirizzo 127.0.0.1. I demoni dei vari servizi sono in ascolto su questo indirizzo oltre che sull'indirizzo fisico della macchina, in questo modo si possono eseguire dei test o, come nel caso di filezilla server, amministrare un server locale. La porta in questo caso non è quella prevista dal protocollo FTP ma una porta diversa in quanto l'amministrazione del server non utilizza il protocollo FTP.

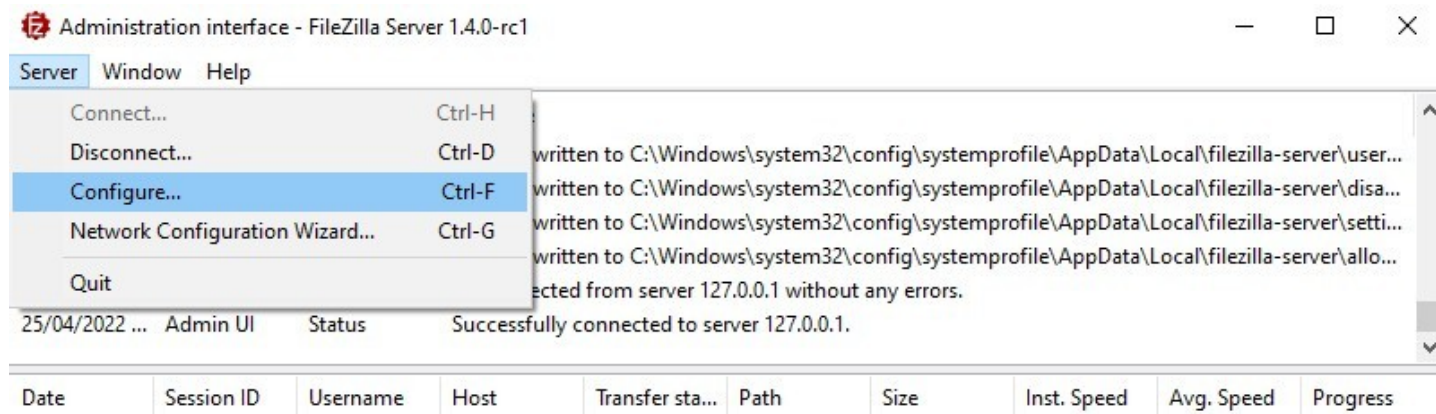
A destra la schermata che appare dopo l'installazione del server filezilla.

Dopo l'installazione avremo la possibilità di avviare, stoppare o amministrare il server andando nei programmi di windows nella sezione filezilla server.

Se il server è avviato allora potremo connettervi al server per l'amministrazione cliccando su pulsante connetti.

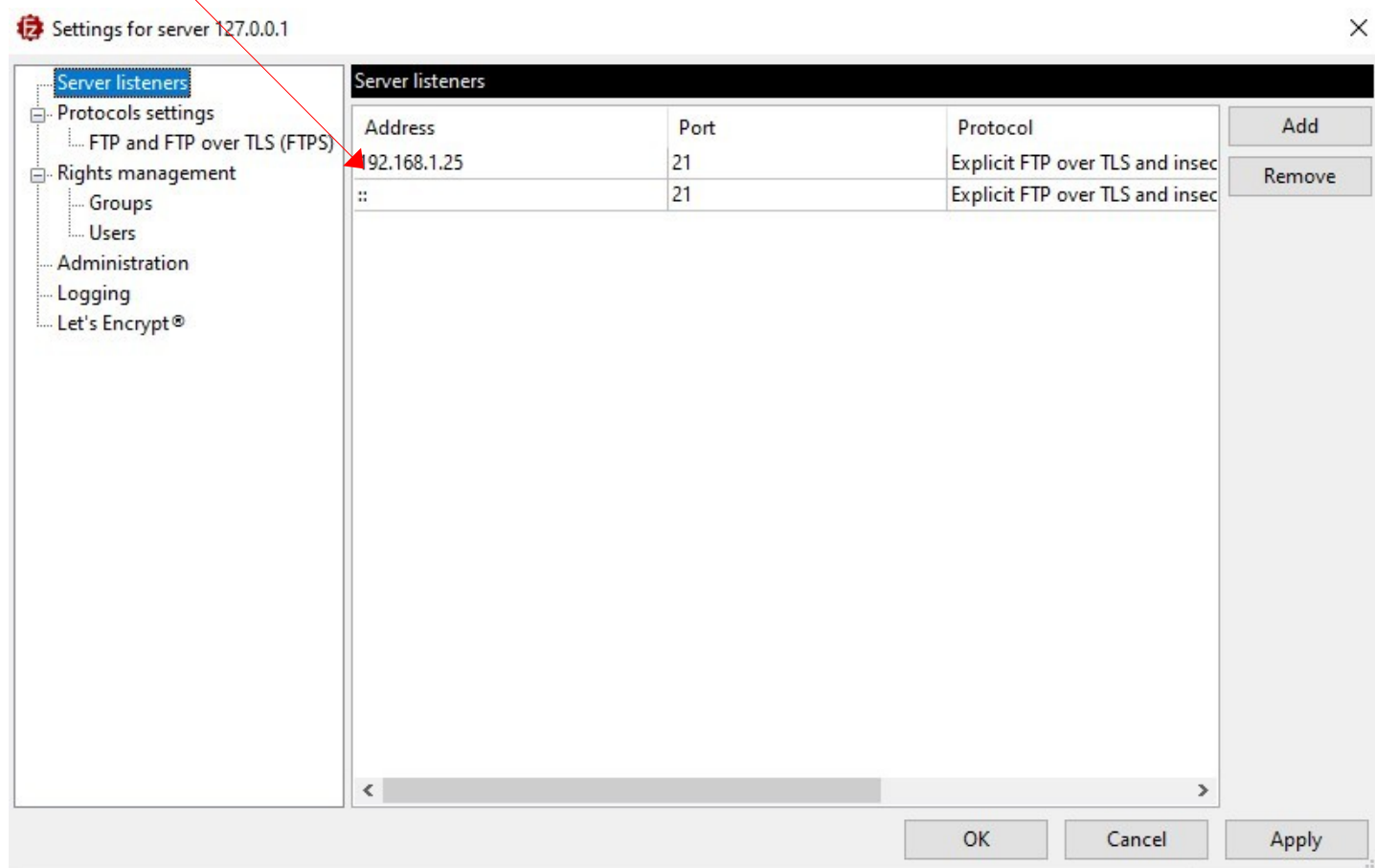


A questo punto se non ci sono problemi saremo connessi al server e tramite “configure” potremo accedere alla configurazione del server.



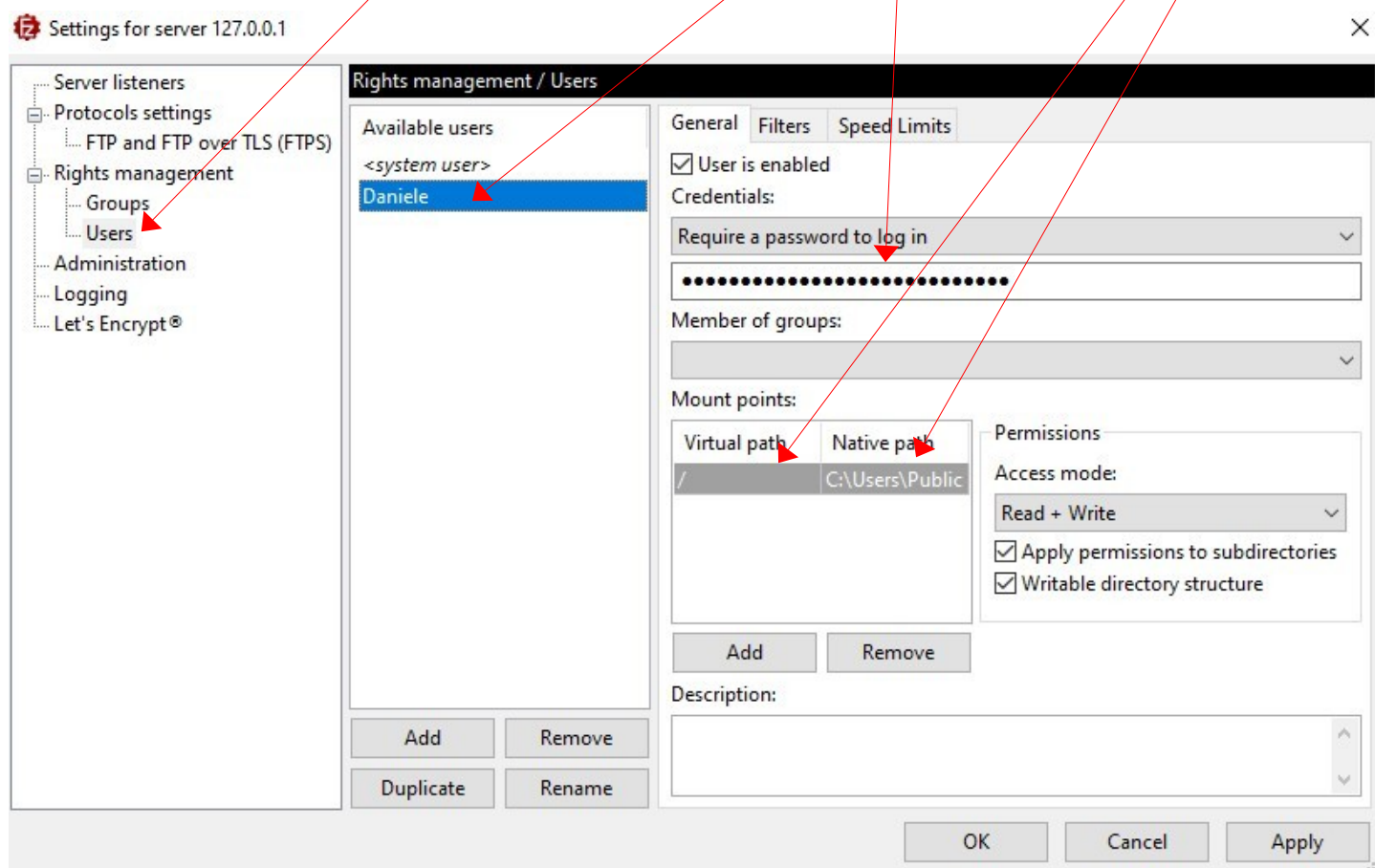
Configure the FileZilla FTP server.

Per procedere alla configurazione del server è importante conoscere il proprio indirizzo IP da inserire nella sezione address, insieme alla porta 21.



Apply per confermare.

Successivamente nella sezione Users dovremo creare un utente con password, indicando la cartella che vorremmo condividere.



Su virtual path andrà messo il simbolo di root / e su native path andrà scritto il percorso della cartella da condividere. Al termine premere Apply per confermare.

A questo punto lanciando filezilla client da un pc connesso alla stessa rete potremo accedere al server digitando l'indirizzo IP, il nome utente, la password e la porta.

ESERCITAZIONE N.2

Lavorando a coppia con due pc, si procede all'installazione del client e del server su entrambi i PC per eseguire le operazioni in maniera speculare.

Probabilmente il firewall di windows bloccherà la connessione proveniente dal client, pertanto durante l'esercitazione è opportuno disattivare il firewall sia sul server che sul client.

Cosa fare:

1. Aprire wireshark per catturare i pacchetti ftp come fatto precedentemente, e trasferire un file da client a server e viceversa.
2. Catturare i pacchetti e descrivere le differenze tra quanto visto in precedenza.

Quanto fatto finora ci ha messo in condizione di conoscere il funzionamento del protocollo FTP, che anche se datato, viene ancora oggi utilizzato in diversi casi per la sua semplicità e rapidità d'uso. Un'ultima informazione è relativa al fatto che in diversi casi è possibile accedere al server ftp utilizzando un normale browser, digitando nella barra degli indirizzi <ftp://nomeserver> il nostro browser ci consentirà di visualizzare e scaricare i file presenti nel server.

