

Relazione di progetto di
PROGRAMMAZIONE DI RETI
Anno 2021-2022

Cognome e Nome: Montanari Nicola

Matricola: 0000970119

Traccia: Traccia N.2

GitHub Repository: <https://github.com/NIK4658/File-transfer-UDP>

Architettura Client-Server UDP per trasferimento file

Scopo Progetto:

Lo scopo del progetto è quello di progettare ed implementare in linguaggio Python un'applicazione Client-Server per il trasferimento di file che impieghi il servizio di rete senza connessione.

Funzionalità Client:

Il Client è in grado di fornire le seguenti funzionalità con i relativi comandi:

- **List** : Richiede al Server una lista dei nomi dei file disponibili al download.
- **get <filename>** : Invia una Richiesta per il DOWNLOAD di un file specifico, che verrà eseguito immediatamente dopo aver ricevuto un messaggio di conferma da parte del Server. Oppure, ovviamente, il Client dovrà gestire l'errore in caso il server non sia in grado di inviare il file desiderato.
- **put <filename>** : Effettua l'UPLOAD sul Server di un file specificato dall'utente, presente nella cartella "Client". Riceve poi un messaggio di risposta con l'esito dell'operazione.

Funzionalità Server:

Il Server è in grado di fornire le seguenti funzionalità:

- Invio del messaggio di risposta al Client che ha precedentemente eseguito il comando "list". Il messaggio di risposta conterrà la lista dei nomi dei file presenti nella cartella "Server".
- Invio del messaggio di risposta al Client che ha precedentemente eseguito il comando "get <filename>". Il messaggio di risposta conterrà il file richiesto, se presente, od un opportuno messaggio di errore.
- Invio del messaggio di risposta al Client che ha precedentemente eseguito il comando "put <filename>". Il messaggio di risposta sarà l'esito dell'operazione di UPLOAD da parte del Client verso il Server.

Funzionamento dei due programmi:

Il Server, nel caso non ci siano messaggi di comando da parte di alcun Client, attende questi ultimi eseguendo un ciclo infinito. Alla ricezione di un messaggio di comando da parte di un Client, si attiva un Thread che avrà lo scopo di gestire quest'ultima richiesta, così da non bloccare l'intero Server nel caso ci fossero più richieste nello stesso momento. È presente un timeout al socket per assicurarsi che i file vengano trasferiti correttamente.

Schema UML del dialogo tra Client e Server

