Documentazione del Progetto di Programmazione di Reti

1 Introduzione

Questa è la documentazione del progetto di Programmazione di Reti relativo alla Traccia 1: Sistema di Chat Client-Server in Python. Il progetto mira a implementare una chat che permetta a più utenti di connettersi a un server centrale e di comunicare tra loro in una chatroom condivisa.

Il progetto è suddiviso in due componenti principali: il client e il server. Il server gestisce le connessioni dei client e instrada i messaggi in modalità broadcast all'interno della chat. Il client permette agli utenti di connettersi al server tramite il protocollo TCP, inviare messaggi e visualizzare i messaggi ricevuti dagli altri utenti nella chatroom. I messaggi precedenti alla connessione non vengono "salvati" nel server.

Sono state adottate scelte architetturali per garantire affidabilità e concorrenza, come l'utilizzo del protocollo TCP e dei thread per gestire l'invio e la ricezione dei messaggi su entrambi i lati della connessione.

2 Client

Il client utilizza il modulo socket di Python per stabilire una connessione con il server della chatroom. L'utente deve specificare l'indirizzo IP del server e un nome utente univoco.

Una volta avviato, il client crea un socket e tenta di connettersi al server. Se la connessione è stabilita, il client invia il nome utente al server per identificarsi. Il client avvia due thread separati per gestire la trasmissione e la ricezione dei messaggi, permettendo all'utente di scrivere e leggere contemporaneamente senza bloccare il programma.

L'interfaccia grafica (GUI) del client è stata sviluppata utilizzando Tkinter e include pulsanti per visualizzare la lista degli utenti online e per inviare messaggi privati.

2.1 Comandi supportati dal client

- `/list`: Visualizza la lista degli utenti online nella chatroom.
- `/pvt`: Invia un messaggio privato a un altro utente.

3 Server

Il server della chatroom consente agli utenti di connettersi e comunicare utilizzando il protocollo TCP/IP. Il server mantiene una lista dei client connessi e gestisce la trasmissione dei messaggi tra di essi. Utilizza un approccio basato su thread per gestire simultaneamente le connessioni multiple.

Quando arriva un nuovo messaggio, il server verifica la presenza del destinatario e inoltra il messaggio se l'utente è online. Se il destinatario non è presente, viene inviata una risposta al mittente.

3.1 Funzionalità principali del server

- Gestione delle connessioni dei client.
- Trasmissione di messaggi broadcast e privati tra client specifici.
- Fornitura della lista degli utenti connessi.

3.2 Comandi supportati dal server

- `/list`: Restituisce la lista degli utenti connessi.
- `/pvt`: Consente di inviare un messaggio privato a un altro utente.