

Documentazione Progetto

I Device sono connessi tra loro e con il Server con connessioni di tipo TCP. Per la fase di **signup** il device invia MSG_SIGNUP contenente username e password (Anche il comando signup prende come argomento *server_port*) su socket tcp e il server salva i dati sul file SignUp.txt e risponde con MSG_OK se non ci sono errori. Per la gestione del comando **in** il device invia MSG_IN con le credenziali e la porta su cui è in ascolto, il server inserisce l'username, la porta e il timestamp_login (timestamp_logout = \0 perché è online) nella lista dei device connessi salvata in memoria e risponde con MSG_OK se non ci sono errori. Il device rimane connesso al server con un socket tcp, questo permette al server di individuare subito le disconnessioni andando a inserire il timestamp_logout del device.

Per l'invio dei messaggi di testo (*Figura 1*) il *dev_1* invia il primo messaggio (di tipo MSG_TEXT) al server, il formato comprende oltre al messaggio di testo, l'username e la porta, mittente e

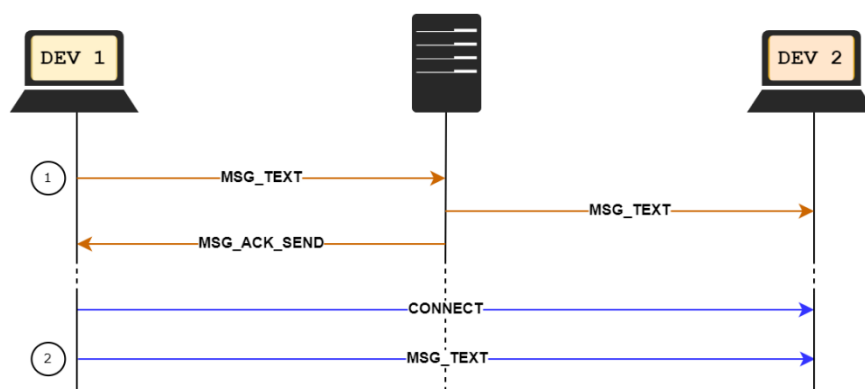


Figura 1

che possiede ora la porta di *dev_2*, tenta di connettersi con un nuovo socket tcp per inviare direttamente i messaggi successivi al primo.

I Device salvano le informazioni degli altri device con cui interagiscono in una lista salvata in memoria (con formato user, port, fd, e campo *chatAttiva*). All'esecuzione di **chat username** (o **la username**) *chatAttiva* di username viene messo a true, i messaggi vengono inviati a tutti i device che hanno *chatAttiva* a true (se il campo port = 0 è il primo messaggio, quindi il device invia al server). Il comando **la** disattiva tutti i campi *chatAttiva* della lista.

Quando un utente richiede la lista dei device online (comando **lu**) invia un messaggio di tipo MSG_ONLINELIST (*Figura 2*) e il server risponde con un pacchetto di tipo MSG_ONLINELIST_N con il numero (*k*) di device online e poi fa *k* invii di messaggi MSG_ONLINELIST con l'username online.

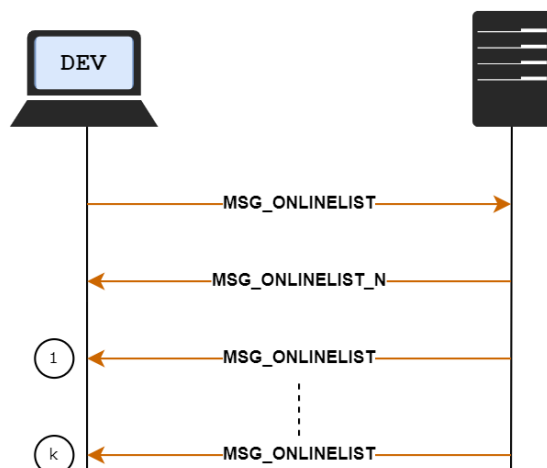


Figura 2

Per ricevere la lista degli utenti che hanno inviato dei messaggi mentre il device era offline (comando **hanging**), viene inviato al server un messaggio di tipo MSG_HANGING (*Figura 3*), il server conta per ogni utente quanti messaggi ha inviato e risponde con una serie di pacchetti di tipo MSG_HANGING_RESPONSE e formato tipo, username mittente, numero di messaggi

inviati e ultimo timestamp, l'ultimo di questa serie di messaggi avrà tipo MSG_END per notificare al device che la lista è terminata.

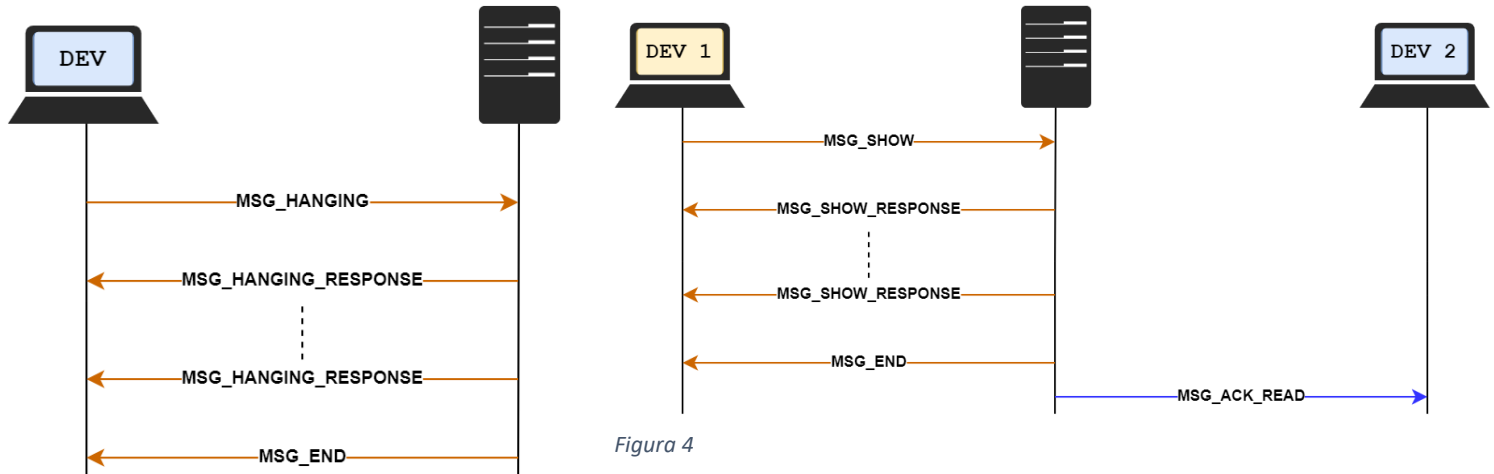


Figura 4

Figura 3

Quando viene eseguito il comando **show username** (Figura 4) il device invia un messaggio MSG_SHOW contenente username e il proprio username, il server risponde con un pacchetto per ogni messaggio (di tipo MSG_SHOW_RESPONSE) contenente il testo del messaggio, l'ultimo di questi pacchetti avrà tipo MSG_END per informare il device che non ci sono altri messaggi da ricevere. Alla fine del comando show il server invia l'**Ack** all'username mittente (dev_2) dei messaggi, prepara un pacchetto con tipo MSG_ACK_READ e username destinatario (dev_1), se il mittente (dev_2) è online invia direttamente il pacchetto altrimenti lo salva su file (dev_2/ackRead.txt) e lo invierà quando l'username farà un nuovo accesso (comando in).

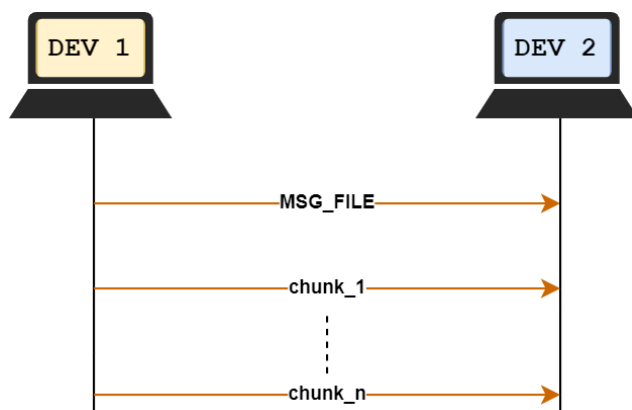


Figura 5

Con il comando **share file** viene inviato il file a tutti i device che hanno *chatAttiva* a ture, se c'è già un socket aperto con il device viene inviato direttamente, altrimenti viene richiesta la porta del device al server (MSG_PORT_REQ) e poi crea una connessione con un nuovo socket. Il device invia come primo messaggio un pacchetto con tipo MSG_FILE (Figura 5) contenente nome e porta mittente, nome file e *nChunk* (numero di pacchetti che verranno inviati, ogni chunk è di 256 byte). Successivamente viene inviato il file suddiviso in *nChunk* pacchetti.

I socket tcp permettono ai device di individuare la disconnessione del server e degli altri device con cui sono connessi, se il server va offline gli utenti possono continuare a comunicare tra loro (con messaggi di testo e file), aprire o aggiungere un utente a una nuova chat purché il device possieda già il numero di porta del device destinatario (ricevuta dal server in precedenza).

Ogni device crea una cartella contenente il file *Rubrica.txt* e un file *username.txt* dove viene salvata la cronologia dei messaggi con username (con formato inviato, recapitato e testo). All'interno della cartella del server è presente il file *SingUp.txt* (con formato username e password) e una cartella per ogni utente contenente *ackRead.txt* (se ci sono degli ack bufferizzati da inviare) e il file *hanging.txt* dove sono salvati i messaggi bufferizzati destinati all'utente con formato userSource, timestamp di invio e testo del messaggio.