

# Documentazione del secondo progetto

Gruppo 19 - 3 membri

## 1 Descrizione del progetto

L'idea del progetto è stata la realizzazione di un contatore intelligente, in grado di gestire i dispositivi ad esso connessi in modo da evitare di superare la soglia di potenza consentita.

Abbiamo considerato come ambiente di utilizzo un'abitazione e inserito alcuni dei vari elettrodomestici possibili con le loro rispettive potenze, prendendone i riferimenti online.

Il contatore tiene traccia dei dispositivi connessi, che inviano un messaggio di **ALIVE** ogni secondo e li considera disconnessi se non riceve nessun **ALIVE** per almeno 10 secondi.

Il server (contatore) invia periodicamente il messaggio di **HELLO** in multicast, per capire quali dispositivi vogliono connettersi alla rete; questo messaggio di discovery viene inviato ogni 2,5 secondi, un intervallo scelto perché sufficientemente lungo da non intasare la rete ma sufficiente breve da consentire tempi di risposta del sistema accettabili dall'utente finale. Non è stata presa in considerazione l'energia utilizzata per il polling né per quanto riguarda il contatore che i dispositivi, perché si suppongono collegati alla rete elettrica e non alimentati a batterie.

Abbiamo scelto la configurazione in cui è il server a effettuare l'operazione di discovery sia perché in questo modo può, con un solo messaggio, informare anche i dispositivi della potenza che può ancora erogare, sia perché altrimenti sarebbe stato necessario un messaggio di risposta al discovery del client per informarlo sulla porta del socket unicast del server, la cui ricezione, visto che ci troviamo in una situazione di protocollo inaffidabile, non è garantita.

Nel messaggio di **HELLO**, il server comunica la porta sui cui i dispositivi devono inviare il messaggio di **ALIVE** e la potenza libera rimanente. Il client, una volta ricevuto il messaggio di **HELLO** e aver controllato che la sua potenza di accensione sia minore di quella disponibile, si connette e comincia a inviare i messaggi di **ALIVE** in unicast al server.

Si è deciso di gestire la situazione in cui più client si connettono contemporaneamente superando la soglia di potenza consentita (i client non sono consci della presenza l'uno dell'altro) attraverso un protocollo ispirato al CSMA: il server comunica ai client, che rimangono sempre in ascolto per i nuovi messaggi di **HELLO**, la situazione di errore inviando una potenza negativa; a questo punto tutti i client si disconnettono e aspettano un certo tempo (deciso casualmente)

prima di provare a riconnettersi. In questo modo i client che per primi termineranno l'attesa potranno connettersi, quelli che la termineranno successivamente riceveranno messaggi di **HELLO** che segnalano una potenza non sufficiente alla connessione e quindi rimarranno in attesa di avere potenza disponibile.

Il codice è corredato di Javadoc ed è possibile modificare la porta del server e la potenza massima consentita modificando i valori nella classe *Constants* nel package *Commons*: all'utente non è stata data la possibilità di modificare la porta del server o dei client perché si suppone che ogni dispositivo abbia la propria scheda di rete e che quindi abbia tutte le porte disponibili. Visto che, inoltre, si tratta di sistemi integrati sui dispositivi, la potenza (quella disponibile del contatore o quella utilizzata dai client) è un parametro dipendente dal particolare elettrodomestico e quindi non modificabile.

## 2 Utilizzo

Come prima operazione, avviare la classe *ServerGUI*. In seguito avviare la classe *ClientGUI* per instanziare un nuovo dispositivo.

Al momento dell'apertura di *ClientGUI*, si può scegliere il tipo di dispositivo che desidera connettersi alla rete.

Una volta scelto, tramite la selezione del *jRadioButton ON*, il dispositivo proverà a connettersi rispettando le condizioni prima citate.

Sull'interfaccia del server è possibile visualizzare i dispositivi connessi e, in caso del mancato invio da parte del client dei messaggi **ALIVE** per più di 10 secondi, l'elettrodomestico verrà considerato **DISCONNECTED**. Se lo stesso client volesse riconnettersi, dovrà attendere un nuovo messaggio di **HELLO** da parte del server.

Il client potrà anche spegnersi selezionando il *jRadioButton OFF*, non inviando più i messaggi di **ALIVE** al server.

Come spiegato nel funzionamento del server, il client potrà ricollegarsi prima dei 10 secondi dall'ultimo **ALIVE** inviato, oppure dovrà aspettare un nuovo messaggio di **HELLO** da parte del server.

I client già connessi sono in grado di riconoscere lo spegnimento del server e si spegneranno a loro volta dopo 20 secondi dalla ricezione dell'ultimo messaggio di **HELLO**.

Nell'interfaccia del Client, una volta selezionato il dispositivo IOT da connettere, è possibile visualizzare le seguenti informazioni:

- indirizzo e porta del server, una volta ricevuto il messaggio di **HELLO**;
- il tipo di dispositivo IOT scelto;
- lo stato del dispositivo;
- la conferma dell'invio del messaggio **ALIVE**, una volta iniziata la comunicazione col server.

Nell'interfaccia del Server è possibile visualizzare le seguenti informazioni:

- potenza consumata e potenza totale;
- tipo di dispositivo e tempo dall'ultimo **ALIVE** ricevuto, eventualmente **DISCONNECTED**;
- il pulsante **REFRESH** per forzare l'aggiornamento della visualizzazione dello stato dei dispositivi.