

Relazione Progetto

Traccia 1 - Progetto Droni

A.A 2021/2022

Gabos Norbert

0000970451

tiberiunorbert.gabos@studio.unibo.it

Introduzione	3
Istruzioni esecuzione	4
Istruzioni per l'uso	4
Descrizione	4
drone.py	4
gateway.py	4
client.py	5
client_gui.pyw	5
udp_server.py	6
udp_server_multi_client.py	6
Dettagli implementativi	6
Dimensioni buffer	6
Thread	6
Requisiti	6
Librerie utilizzate	6

1. Introduzione

Il progetto consiste nella realizzazione di un sistema che gestisce la comunicazione tra un **client** e 3 **droni** per le consegne. Tale interazione deve avvenire tramite un intermediario, chiamato **gateway**. Il client sa quali droni sono liberi e può scegliere di assegnargli una consegna. I droni una volta ricevuto l'ordine devono "consegnare" il pacco e presentarsi al gateway come disponibili.

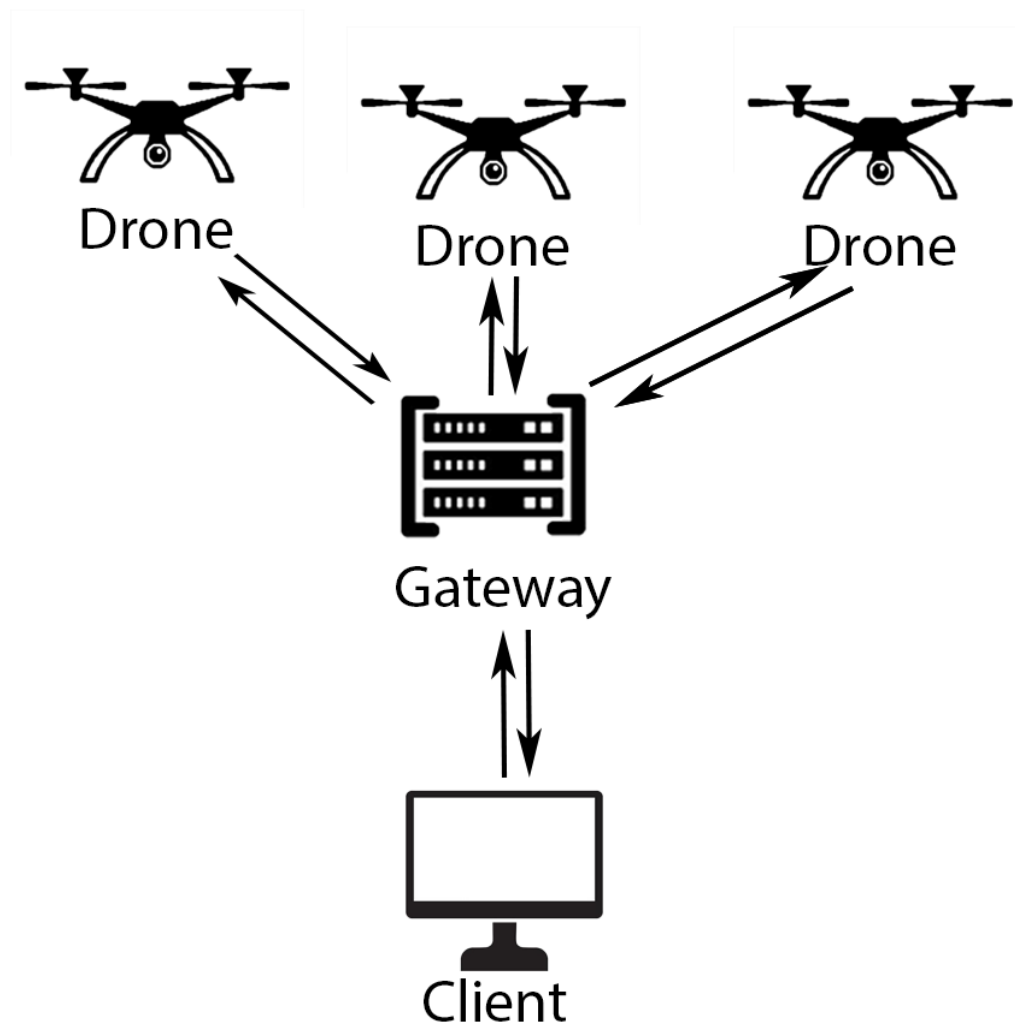


fig. 1.1 Illustrazione dell'interazione tra la varie componenti

2. Istruzioni esecuzione

Per eseguire il codice è necessario rispettare l'ordine di esecuzione dei vari script. Per prima cosa si avvia il gateway, poi si possono avviare i vari droni e il client (questi ultimi si possono avviare in qualsiasi ordine).

Per avviare le varie componenti basta scrivere:

```
python3 <nome_script>
```

oppure, per quanto riguarda il **client_gui**, basta fare doppio click.

3. Istruzioni per l'uso

Per prima cosa si avvia il gateway, sul quale sono visualizzate tutte le comunicazioni del client e dei vari droni.

I droni possono essere aggiunti in qualsiasi momento anche dopo che il client ha iniziato a mandare le istruzioni agli altri droni.

Per quanto riguarda il client si può scegliere di utilizzare la versione CLI (text-based)

client.py, oppure la versione GUI (con interfaccia grafica) **client_gui.pyw**.

Nella versione CLI, per aggiornare la lista dei droni disponibili e' necessario digitare "update".

All'interno del progetto c'è un file setup.json, che permette di setuppare alcune caratteristiche del gateway, come l'ip e la porta, inoltre si può anche aumentare il numero massimo di droni che il gateway può ospitare.

4. Descrizione

Il codice è diviso in vari programmi:

4.1. drone.py

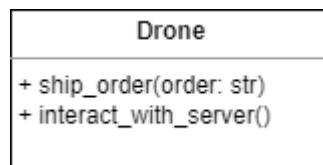


fig. 4.1 Rappresenta la struttura di drone

Il drone si occupa di collegarsi al gateway, dopo di che aspetta di ricevere ordini. Questa componente memorizza la destinazione data dal gateway e non fa altro che aspettare un tempo tra i 5 e i 10 secondi, poi si ripresenta dal gateway come disponibile aspettando un nuovo ordine.

4.2. gateway.py

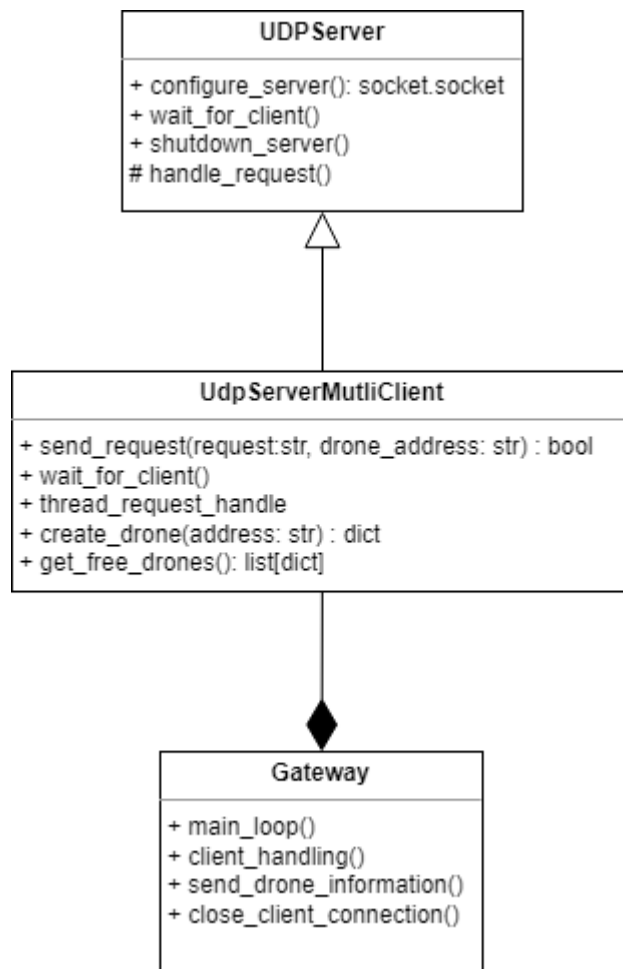


fig. 4.2 Rappresenta la struttura del gateway

Questa componente è il core dell'applicazione. Si occupa di creare il collegamento tra i vari droni (mediante protocollo UDP) e il client (mediante il protocollo TCP).

Il suo compito principale è quello di ricevere ordini dal client e di trasmetterli al corrispettivo drone. Inoltre controlla che i pacchetti inviati dai vari componenti siano correttamente formattati.

4.3. client.py

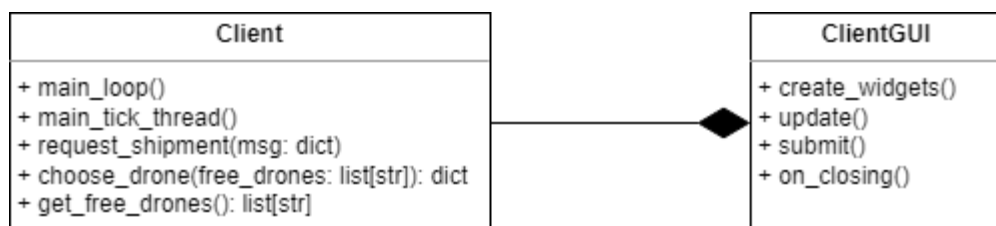


fig. 4.3 Rappresenta la struttura del client

Il client si occupa di inviare al gateway gli ordini che saranno poi assegnati ai vari droni. Inoltre è in grado di ricevere messaggi, come la lista dei droni attualmente disponibili.

4.4. client_gui.pyw

Il client_gui non è altro che un'estensione del client. Questa componente è una

versione più user friendly della precedente. Come feature aggiuntiva, è in grado di aggiornare la lista dei droni senza l'interazione con l'utente.

4.5. udp_server.py

Questo script non è altro che una versione abstract di un server udp.

4.6. udp_server_multi_client.py

All'interno di questo script vengono gestite tutte le interazioni e la creazione dei vari droni.

5. Dettagli implementativi

5.1. Dimensioni buffer

Per la dimensione dei buffer è stata utilizzata 1024 sia per la comunicazione con il client, che con i droni, in quando i dati trasmessi sono molto simili e non superano i 50 caratteri, che codificati in UTF-8, non si superano per l'appunto i 50 byte e rimane abbastanza spazio sulla banda per implementazioni future (le dimensioni dei buffer possono essere modificate dai rispettivi file).

5.2. Thread

- gateway
 - Viene creato un thread per la ricezione e l'invio dei dati verso i droni
 - Viene creato un thread per la creazione dei vari thread dei droni
 - Viene creato un thread per l'ascolto e la ricezioni di pacchetti da parte del client
 - Viene creato un thread per la gestione di tutti i droni
- client
 - Viene creato un thread per la ricezione dei dati solo per la versione con la GUI per rendere la user-experience più gradevole

6. Requisiti

Il progetto per poter essere eseguito necessita di una versione di python ≥ 3.9 . Per quanto riguarda le librerie esterne, sono state utilizzate solamente librerie già preesistenti dentro a python, quindi non è presente alcun file requirements.txt.

7. Librerie utilizzate

- tkinter
- json
- socket
- threading
- time
- random
- os
- sys
- abc