

# Progetto Dronazon

Sistemi Distribuiti e Pervasivi – Appello Settembre 2021

Marco Cutedchia

Matricola: 983828

## Info varie

- Ogni 2 secondi, dopo essere entrati nel sistema, tutti i droni provano a mandare un messaggio di heartbeat al master per essere sicuri che sia ancora vivo
- Tutte le comunicazioni gRpc hanno una deadline di 2 secondi, unica eccezione è la richiesta del lock per l'area di ricarica che non ha una deadline
- Gli endpoint RESTful del server admin sono:

```
GET /drones
POST /drones/{droneId}
DELETE /drones/{droneId}
GET /stats/last/{howMany}
GET /stats/after/{ts1}/before/{ts2}
GET /stats/average-deliveries/after/{ts1}/before/{ts2}
GET /stats/average-kms-travelled/after/{ts1}/before/{ts2}
POST /stats/
```

# Fase di avvio

Server Admin



Drone in entrata



Droni nel sistema



POST /drones/4  
Ciao, sono <ID=4, ip=localhost, port=9004>



# Fase di avvio

Server Admin



Drone in entrata



Droni nel sistema



POST /drones/4  
Ciao, sono <ID=4, ip=localhost, port=9004>

Benvenuto. Ti trovi a (x=2, y=7)  
Droni nella rete: [  
  <ID=4, ip=localhost, port=9004>,  
  <ID=1, ip=localhost, port=9001>,  
  <ID=2, ip=localhost, port=9002>,  
  <ID=3, ip=localhost, port=9003>  
]

# Fase di avvio

Server Admin



Benvenuto. Ti trovi a (x=2, y=7). Droni nella rete: [...]

Drone in entrata



Droni nel sistema



Sono <ID=4, ip=localhost, port=9004>

Sono <ID=4, ip=localhost, port=9004>

Sono <ID=4, ip=localhost, port=9004>

# Fase di avvio – Master esistente

Server Admin



Benvenuto. Ti trovi a (x=2, y=7). Droni nella rete: [...]



Drone in entrata



Droni nel sistema



- Tutti i droni che hanno risposto vengono salvati.
- Se un drone ha risposto dicendo di essere il master viene segnato

Sono <ID=1, ip=..., port=9001, NON MASTER>

Sono <ID=2, ip=..., port=9002, NON MASTER>

Sono <ID=3, ip=..., port=9004, MASTER>

# Fase di avvio – Master mancante

Server Admin



Benvenuto. Ti trovi a (x=2, y=7). Droni nella rete: [...]



Drone in entrata



Droni nel sistema



- Tutti i droni che hanno risposto vengono salvati
- Il drone va avanti senza conoscere un master
- Inizierà un elezione solo quando avrà bisogno di comunicare con il master

Sono <ID=1, ip=..., port=9001, NON MASTER>

Sono <ID=2, ip=..., port=9002, NON MASTER>

Sono <ID=3, ip=..., port=9004, NON MASTER>

## Elezione – Chang and Roberts

- Ogni drone mantiene una lista contenente tutti gli altri droni nella rete
- Questa lista è utilizzata per capire chi è il prossimo drone nell'anello
- Ogni volta che una comunicazione con il drone master fallisce (o non è possibile) una elezione viene avviata
- Una volta che un nuovo drone viene eletto master questo richiede i dati a tutti i droni nella rete

Due tipi di messaggi vengono scambiati durante un'elezione: ELECTION ed ELECTED



## Elezione – Casi limite durante l'uscita di un drone

- *Durante l'invio dei messaggi ELECTION o ELECTED il drone futuro leader abbandona l'anello, portando gli altri droni a scambiarsi all'infinito il messaggio senza mai finire*
- *Il drone futuro master riceve il suo messaggio ELECTION ma abbandona il sistema prima di mandare il messaggio ELECTED, lasciando in stallo tutti gli altri droni*

SOLUZIONE: Se un drone ha ricevuto un messaggio ELECTION allora non può completare lo spegnimento fino alla fine dell'elezione.

Fine dell'elezione vuol dire:

- Per i droni master quando hanno ricevuto indietro tutti i messaggi ELECTED che hanno inviato
- Per i droni follower quando hanno ricevuto ed inoltrato un messaggio ELECTED

## Elezione – Un caso limite durante l'entrata di un drone

*Un drone entra durante una elezione, il futuro master risponde di non essere lui il master perché non ha ancora ricevuto indietro il suo messaggio ELECTION. Il drone entrante perde il messaggio ELECTED perché il drone precedente nell'anello ha processato prima il messaggio ELECTED del suo messaggio.*

### SOLUZIONE:

Se un drone già master riceve un messaggio ELECTION allora ferma il messaggio ed invia un messaggio ELECTED con se stesso per sicurezza.

I droni che ricevono un messaggio ELECTED conoscendo già il master inoltrano senza far nulla.

# Assegnamento degli ordini

## Synchronized

Coda ordini pendenti

1	2	3	4
---	---	---	---

1. Assegno ordini pendenti con i droni disponibili
2. Mando in parallelo queste richieste

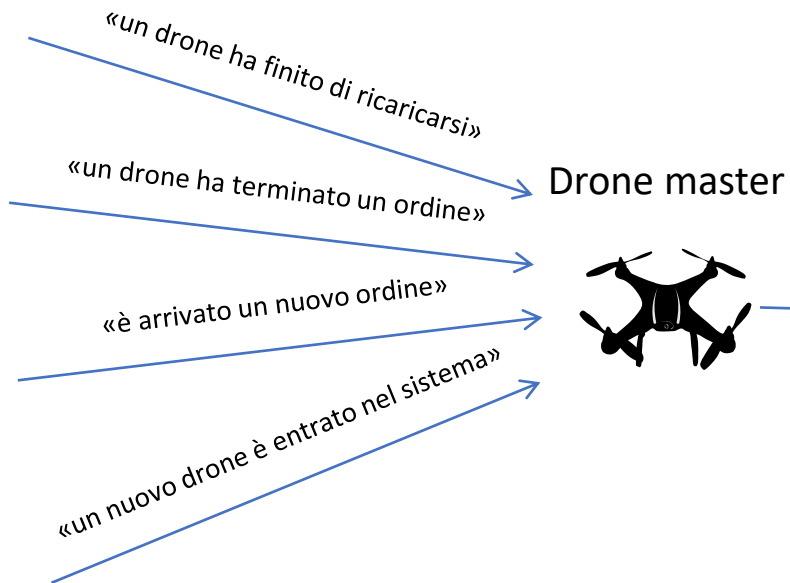
Non è detto che assegno tutti gli ordini!

Rimettere un ordine nella coda riattiva questo processo

**Se il drone accetta la consegna:**  
Mi segno che il drone è impegnato e non proverò più ad assegnargli ordini fino alla conferma di consegna

**Se il drone rifiuta la consegna:**  
Rimetto l'ordine nella coda. Mi segno che il drone non è disponibile per le consegne fino a suo aggiornamento.

**Se il drone non è raggiungibile:**  
Rimetto l'ordine nella coda. Cancello il drone dalla lista dei droni nel sistema



# Mutua esclusione per la ricarica

- Quando si richiede il lock vengono aperte più chiamate sincrone (bloccanti) verso tutti gli altri droni su diversi thread e viene fatta una join su tutti i thread
- I droni che ricevono la richiesta del lock rispondono immediatamente se vogliono concedere il lock, oppure chiamano *wait()* se hanno la precedenza ed useranno *notifyAll()* quando non sono più interessati al lock
- Unica chiamata gRpc senza alcuna deadline: se il drone è irraggiungibile è come se avesse concesso il lock

# Mutua esclusione per la ricarica

- Un drone a cui viene concesso il lock inizierà a rifiutare tutti i futuri ordini
- Non è detto che inizi a ricaricarsi subito! Potrebbe dover completare un ordine assegnato mentre aspettava il lock
- Inizierà ricaricarsi subito dopo aver segnalato al master che ha terminato la consegna
  - Questo potrebbe implicare una o più elezioni nel frattempo
- Una volta completata la ricarica segnala al master di aver finito di ricaricarsi e ricomincia ad accettare gli ordini

# Fase di shutdown del master

Esegui quando non sto consegnando un ordine

Ferma l'arrivo di ordini da MQTT

Quando tutti gli ordini pendenti sono stati assegnati ed accettati, e quando questo drone non sta consegnando nulla

Aspettiamo che non stia avvenendo un'elezione, prendiamo il lock e stacciamo il server gRpc

Mandiamo le statistiche rimaste, chiediamo al server admin di uscire e chiudiamo il processo

```
drone.doWhenThereIsNoOrderToDeliver(
    () -> {
        synchronized (drone) {
            if (drone.isDeliveringOrder()) {
                // Retry next time the drone is free
                return false;
            }
        }
        orderSource.stop();
    }
    orderAssigner.doAsSoonAsThereAreNoPendingOrders(
        () -> {
            drone.doWhenThereIsNoOrderToDeliver(
                () -> {
                    synchronized (drone) {
                        synchronized (electionManager) {
                            electionManager.waitForNoElectionIsHappening();
                            communicationClient.shutdown();
                        }
                    }
                    statsTracker.sendStatsAndShutdown();
                    adminServerClient.requestDroneExit(drone);
                    System.exit(0);
                }
            }
            return true;
        });
    });
    return true;
});
```

Se siamo sfortunati il drone riceve un ordine appena entrato in questo callback. Nel caso, riproviamo la prossima volta che il drone è libero

# Fase di shutdown del master – un caso limite

*Ci sono ordini pendenti ma nessun drone disponibile per consegnarli*

In questo caso, se il master ha abbastanza batteria, proverà lui stesso a consegnarli. Altrimenti resterà in attesa che altri droni entrino nel sistema.

*Un altro caso limite perché se no questa slide è triste*