Dronazon

Gregorio Ghidoli 983827

Order management

Master service

Order Queue

Send Orders Service

(3) chooseAndBlockDrone {lockChooseDrone}

(4) null ? *addOrderIfNotQuitting else (5) send*

Order Receiver Service (Mqtt sub)

Order xxxxx

(1) takeOrderAndNotifyIfEmpty

(2) Order xxxxx

Pending Send Order (gRPC) Clients
(synchronized list)

Send Order (gRPC) Client

v manageOrderResponse
- **updateDroneStatistics**
- releaseAndUpdateDrone {lockChooseDrone}

x - addOrderIfNotQuitting
- removeDrone

Choosen drone

Send Global Statistic Service

Order management - Quit phase

!Chosen Drone

Borderline case #1

A new order arrives but is not being handled yet and the quit phase begins.

Borderline case solution #1

Wait to complete order with a timeout (gRPC_DEFAULT_TIMEOUT).
The gRPC call fails and the master puts the order back into the queue.

!Chosen Drone

Borderline case #2

The drone is **exiting** and an order arrives between the end of the **previous** order management and the **shutdown** of the gRPC server.

Borderline case solution #2

The management of the new order stops. The gRPC call fails and the master puts the order back into the queue.

MASTER DRONE QUIT PHASE

- wait to complete order
- disconnect **Order Receiver Service (MQTT sub)**
- wait if order queue is not empty (***last chance to send orders***)
    - take order and choose drone
        - if there are no available drones the order is not put back in the queue (the order is lost)
        - else **Send Order Client**
            - if the gRPC call is not successful then do not put the order back in the queue (the order is lost)
- wait to receive all orders responses
- quit drone service (gRPC server)
- send last global statistic
- exit from administrator server

! Master Drone

FUTURE WORK: not discard orders and put them into a global queue (or wait until all orders are correctly handled)

Election - some borderline cases (1)

Drone quit while election

Normal flow of execution: A drone that needs to send a message tries to send it until it finds a (next) drone available. So sending a message always ends. But ...

Bordeline case #1

The drone designated as the master sends the Elected message and exits the network. The Elected message will not be stopped by anyone and will continue to run on the network.

Bordeline case solution #1

A drone cannot leave the network if there is an election in progress.

… Another solution

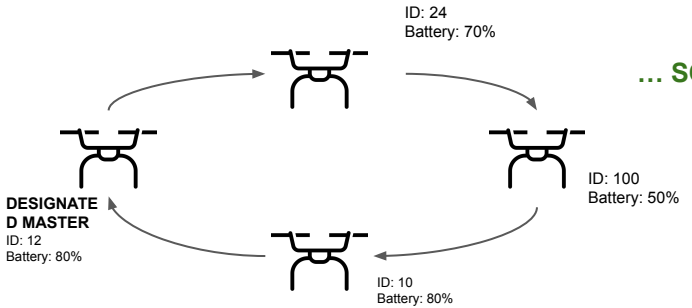
A drone stores Elected messages (with unique ID) and does not forward Elected messages already received.

Bordeline case #2

The drone that will be the new master exits the network when it receives the last Election message but before sending the Elected message. There will be no more messages in the network, no drone will note the master drone and all drones will still be marked as participating in the election discarding all the messages of the drones with smaller Id (of the election) and the network will remain blocked until a node with larger Id will not join the network.

Bordeline case solution #2

A drone cannot leave the network if there is an election in progress.

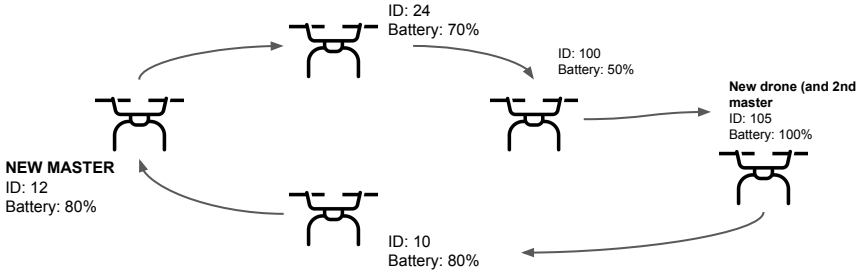


… SO A drone cannot leave the network if there is an election in progress.

Election - some borderline cases (2)

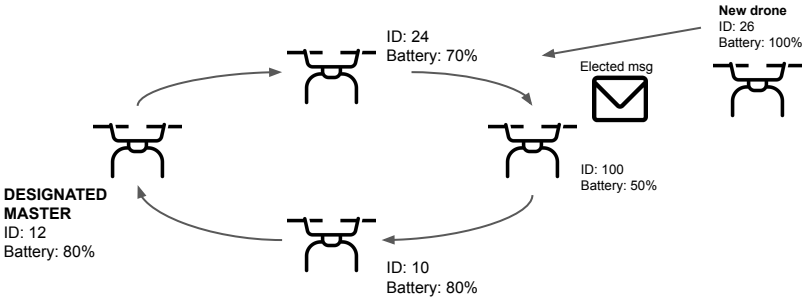
New drone while election

Normal flow of execution: a new drone enters the network, introduces itself to all drones and does not find the master drone. This means that there is an election in progress and therefore it does not start a new election. The new drone will participate to the election by receiving the Election and Elected messages.



If a new election starts, there may be two Master drones at the same time, because the first one does not end its execution anyway!!

Borderline case: a new drone enters the network while there is an election. The new drone introduces itself to the previous drone in the ring which, however, has already forwarded the Elected message. The new drone will have no master because it has not received any Elected messages, but has presented itself to everyone.



Who is the master!?

Solution: The drone will know the master when the master drone gives it an order (or in a future election).

Sensor implementation