

Rete per consegne di pacchi con i droni

Elaborato d'esame per il corso di Programmazione di Reti

A.A. 2022-23

Alessandro Antonini

alessandro.antonini6@studio.unibo.it

Matricola: 0000988825

Analisi

Si dovrà gestire una rete per le consegne a domicilio di pacchi di piccole dimensioni tramite l'utilizzo di droni.

La rete sarà composta da un client, tre droni ed un gateway.

Un operatore potrà assegnare ai droni le consegne tramite l'applicazione client.

Il gateway provvederà ad effettuare il relay dei messaggi verso i droni e ad agire da concentratore per raccogliere i messaggi provenienti dai droni ed inviarli al client.

Requisiti

Il client dovrà poter inserire da console l'indirizzo di consegna del pacco e l'identificativo (o IP address) del drone che dovrà effettuare la consegna.

Il client potrà richiedere al drone la consegna di un pacco solo dopo che lo stesso drone si sarà presentato al gateway come "disponibile".

Quando il gateway riceverà la richiesta di consegna dal client, provvederà ad inviare le informazioni sulla destinazione del pacco al drone incaricato. Tali informazioni saranno visibili sulla console del gateway.

Il drone quindi, una volta ricevuto il messaggio di richiesta di consegna, risulterà occupato per un tempo randomico in modo da simulare l'attesa della consegna. Una volta effettuata la consegna il drone si renderà nuovamente disponibile al gateway inviando un messaggio di avvenuta consegna.

Sulla console del drone dovrà essere visibile l'indirizzo di consegna del pacco per ogni spedizione a lui richiesta.

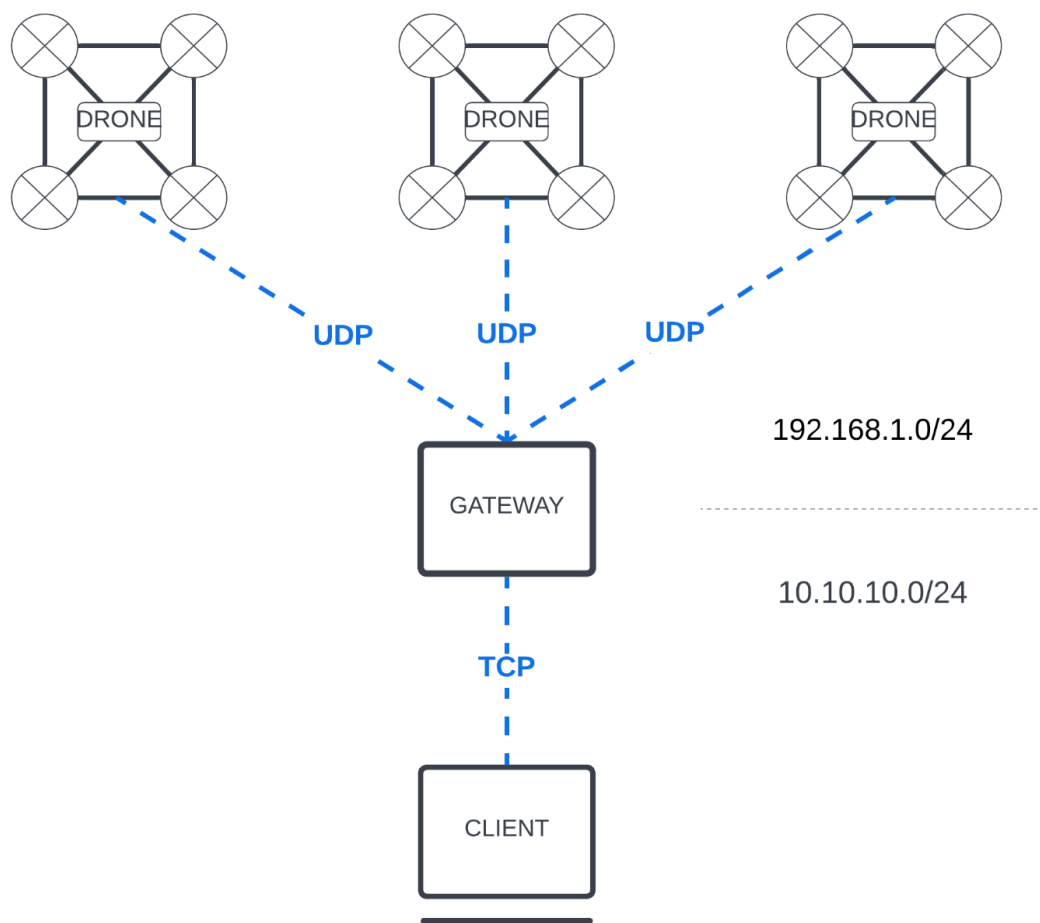
Sulla console del gateway dovranno inoltre comparire tutti i messaggi in transito con sorgente e destinatario.

La connessione tra droni e gateway è di tipo UDP, mentre la connessione tra client e gateway è di tipo TCP.

Ogni drone ha un suo indirizzo IPv4, e così le due interfacce del gateway e il client.

I 3 droni hanno un indirizzamento appartenente ad una rete di Classe C del tipo 192.168.1.0/24

Il Gateway ha due interfacce di rete: quella verso i droni il cui IP Address appartiene allo stesso network dei dispositivi e quella verso il client che ha indirizzo IP appartenente alla classe 10.10.10.0/24, classe a cui appartiene anche l'IP address del gateway.



System model: Client, Gateway and Drones

Design

Sono state realizzate tre applicazioni: una per il client, una per il gateway ed una per i droni. Le varie parti comunicano tra loro scambiandosi messaggi attraverso la rete.

L'applicazione più complessa è quella del gateway. Questa infatti deve ricevere gli ordini dal client e, nel frattempo, continuare a scambiare messaggi con i droni al fine di gestire le consegne. L'applicazione gateway deve dunque gestire almeno due socket diversi (uno per comunicare con i droni e uno per comunicare con il client) ed è stata pertanto divisa in due macro componenti: `clientSideGateway`, che gestisce la comunicazione con il client, e `droneSideGateway`, che invece si occupa di gestire i droni.

Architettura

Il sistema è basato su un'architettura client-server. Un operatore utilizza l'applicazione client per assegnare le consegne ai droni. Il gateway funge da server, gestendo le comunicazioni tra il client e i droni. Il gateway raccoglie inoltre i messaggi provenienti dai droni per inviarli al client.

La connessione tra droni e gateway è di tipo UDP, mentre la connessione tra client e gateway è di tipo TCP. Ogni dispositivo nella rete ha un indirizzo IPv4 univoco. I droni e il gateway appartengono inoltre a due sottoreti diverse, ovvero la rete di classe C 192.168.1.0/24 e la rete di classe A 10.10.10.0/24. Ciò permette di gestire la rete di consegne in modo efficiente e affidabile.

Protocollo Applicativo

I messaggi scambiati tra le applicazioni del sistema sono conformi ad un protocollo applicativo di proprietà, progettato sulla base delle esigenze e dei requisiti del sistema.

Il protocollo applicativo viene descritto di seguito analizzando come le varie componenti del sistema interagiscono tra loro.

Messaggi

I messaggi scambiati tra le applicazioni sono rappresentati nel software dagli oggetti della classe `Message`.

I messaggi sono formati da due campi: comando (`cmd`) e dati (`data`). Il primo campo stabilisce lo scopo del messaggio, mentre il secondo è utilizzato per includere alcuni argomenti relativi al comando scelto.

Ogni oggetto messaggio può essere tradotto in bytes per poi essere incluso nel pacchetto di livello applicativo ed essere spedito attraverso il socket appropriato.

Stato dei droni

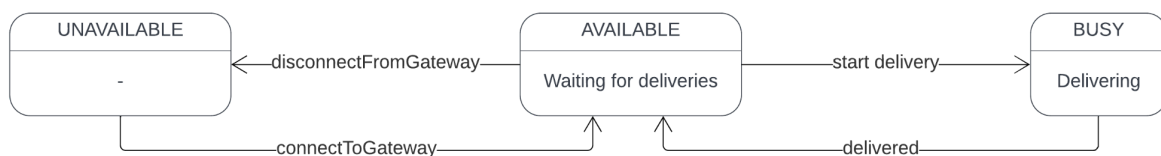
Ogni drone può essere in uno dei seguenti tre stati: `available`, `busy`, `unavailable`.

Se un drone è nello stato `available` significa che si è presentato come disponibile al gateway e sta aspettando di ricevere da quest'ultimo consegne da effettuare.

Quando un drone nello stato available riceve una richiesta di consegna ed inizia la spedizione passa nello stato busy. Durante lo stato busy il drone è appunto occupato ad effettuare la consegna. Il dispositivo non è più disponibile per compiere nuove consegne, ma è ancora connesso al gateway.

Quando la consegna è stata effettuata il drone lo comunica al gateway e ritorna nello stato available.

Quando un drone available vuole escludersi dal sistema e rendersi fuori servizio, allora passa nello stato unavailable. Non è possibile per un drone passare nello stato unavailable se è in consegna (e quindi se il suo stato è busy). È invece necessario che il dispositivo sia nello stato available prima di potersi disconnettere.



UML state diagram: Drone states

Comunicazione Droni-Gateway

Tutte le comunicazioni tra droni e gateway sono di tipo UDP. Il protocollo di trasporto UDP non implementa alcun meccanismo di ritrasmissione o controllo di errori, quindi non è affidabile.

Si è scelto quindi di implementare a livello applicativo un servizio di ritrasmissione dei messaggi persi. L'applicazione ritrasmette tutti i messaggi per cui dopo un certo intervallo di tempo non si è ricevuto un messaggio di conferma di ricezione da parte del destinatario.

I droni, quando si rendono disponibili, vengono memorizzati dal gateway. Ogni drone è identificato dal gateway grazie al proprio indirizzo IP. Il gateway per ogni drone connesso conosce qual è il suo stato, oltre che altre informazioni come il nome del drone, la porta su cui comunicare e da quanto tempo il drone è disponibile.

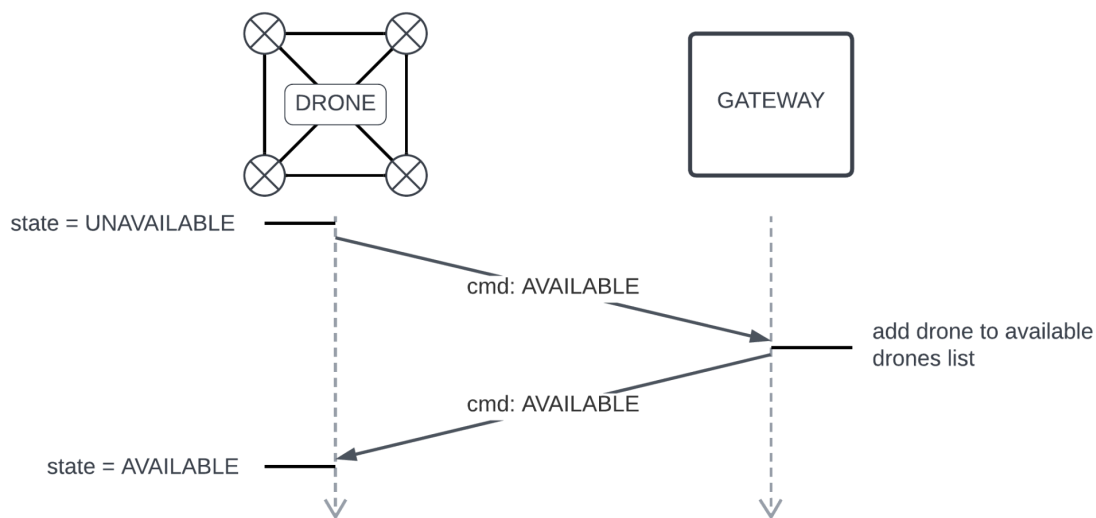
Connessione di un drone al gateway

Quando un nuovo drone deve connettersi al gateway invia a quest'ultimo un messaggio con comando AVAILABLE. Con questo messaggio il drone sta comunicando al gateway che d'ora in poi è disponibile per eventuali nuove consegne.

Nel campo data del messaggio il drone potrà indicare alcune informazioni come per esempio un nome da utilizzare nel sistema (che potrebbe essere utile al client per identificare i droni in modo alternativo all'indirizzo IP).

Il gateway memorizzerà quindi il drone nell'elenco di quelli disponibili e invierà un messaggio di conferma al drone. Il messaggio di conferma è identico al messaggio ricevuto dal drone. Il drone una volta ricevuto il messaggio di conferma passa nello stato AVAILABLE e rimane in attesa di consegne da effettuare.

Se il messaggio di conferma non è ricevuto dal drone entro un certo intervallo di tempo, allora il drone reinvia il primo messaggio al gateway e attende nuovamente la conferma.



Sequence diagram: New drone connection

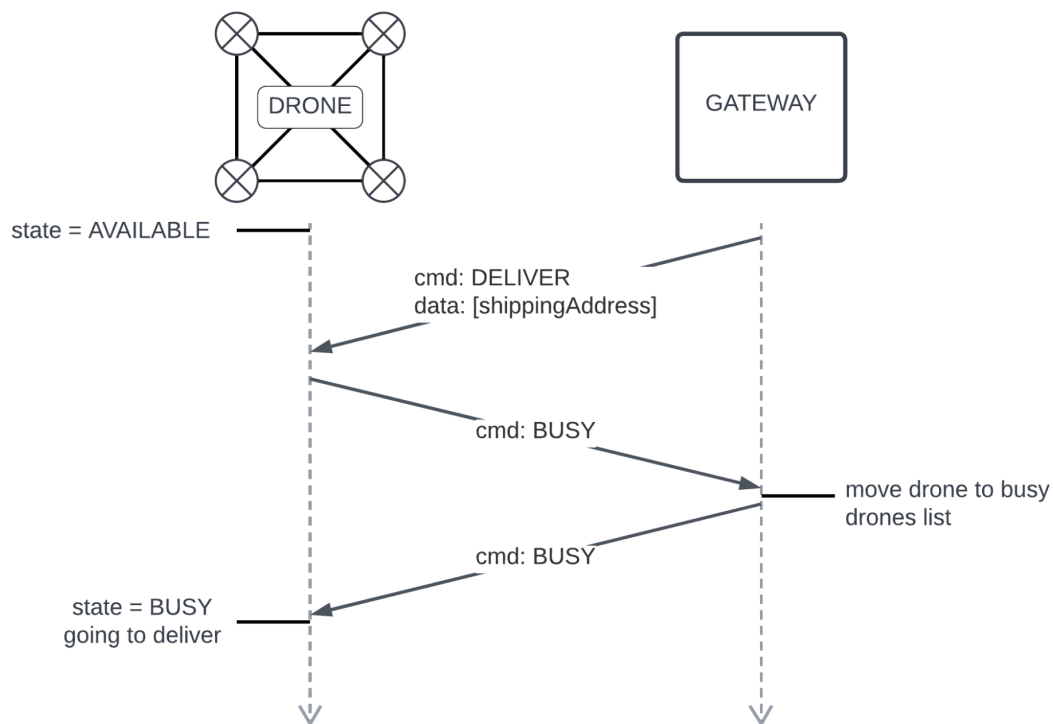
Richiesta di consegna ad un drone

Quando il gateway riceve da parte del client una richiesta di consegna per uno dei droni disponibili, allora deve provvedere ad inoltrare la richiesta al drone e gestire la consegna. Per fare ciò, il gateway invia al drone un messaggio di tipo DELIVER. Il messaggio contiene nel campo dati l'indirizzo di consegna del pacco (shipping address).

Il drone a questo punto per confermare la consegna invia al gateway un messaggio di tipo BUSY, questa volta con il campo dati vuoto. Prima di partire però il drone aspetta di sapere se il gateway ha ricevuto il messaggio attendendo a sua volta da quest'ultimo un messaggio di conferma con il comando BUSY.

Quest'ultimo messaggio risulta essere necessario, infatti qualora il gateway non ricevesse la conferma al comando DELIVER, allora ritenterebbe di inviare il comando DELIVER. Se il drone non aspettasse l'ultima conferma del gateway, ma partisse per la consegna subito dopo aver inviato il messaggio BUSY, nel caso il gateway rinvi il comando DELIVER, il drone non potrebbe riceverlo e non potrebbe riconfermare al gateway la consegna. Il gateway allora penserebbe di aver perso la connessione con il drone ed annullerebbe la consegna, che invece il drone sta svolgendo regolarmente.

È importante notare che quando il gateway riceve il comando BUSY di conferma dal drone, provvede a spostare il drone nella lista dei droni occupati. In questo modo sa che, anche se il drone è in servizio, non può inoltrare nuove richieste di consegna finché il drone non torna nello stato AVAILABLE.

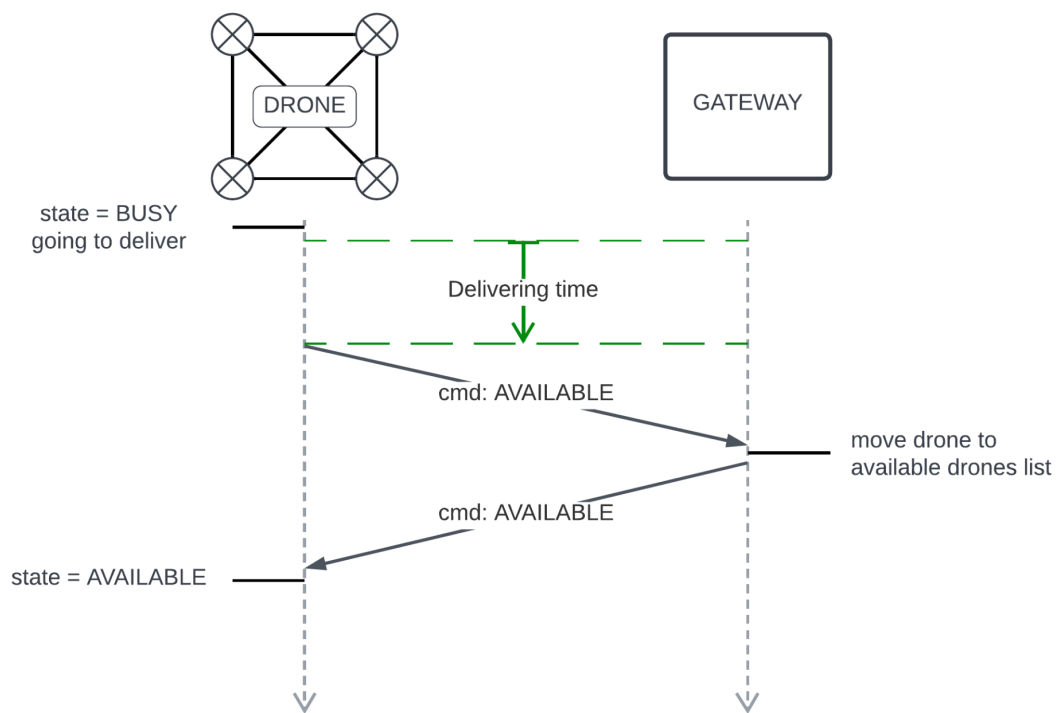


Sequence diagram: New delivery request

Conferma di consegna effettuata

Nel momento in cui il drone ha ricevuto il messaggio BUSY di conferma, parte per la consegna. Per un determinato intervallo di tempo il drone sarà quindi in viaggio e non sarà disponibile a ricevere nuove istruzioni da parte del gateway. Una volta consegnato il pacco, il drone dovrà comunicare al gateway che è pronto per svolgere nuove attività. Il drone invierà allora al gateway un messaggio con il comando AVAILABLE. Si noti che il messaggio ricevuto dal gateway per la notifica della consegna effettuata è uguale al messaggio che il gateway riceve da un nuovo drone. Tuttavia il gateway sa se un drone è occupato o meno in una consegna in quanto, durante il processo di richiesta di consegna, ha spostato il drone nell'elenco dei droni occupati.

Il gateway quindi, quando riceverà un messaggio di tipo AVAILABLE, verificherà se il drone fosse occupato in qualche consegna e, se lo era, lo trasferisce nuovamente nella lista dei droni disponibili. Successivamente invierà una conferma al drone con un nuovo comando AVAILABLE. Il drone di conseguenza si mette nuovamente in attesa di messaggi da parte del gateway.



Sequence diagram: Delivery confirmation

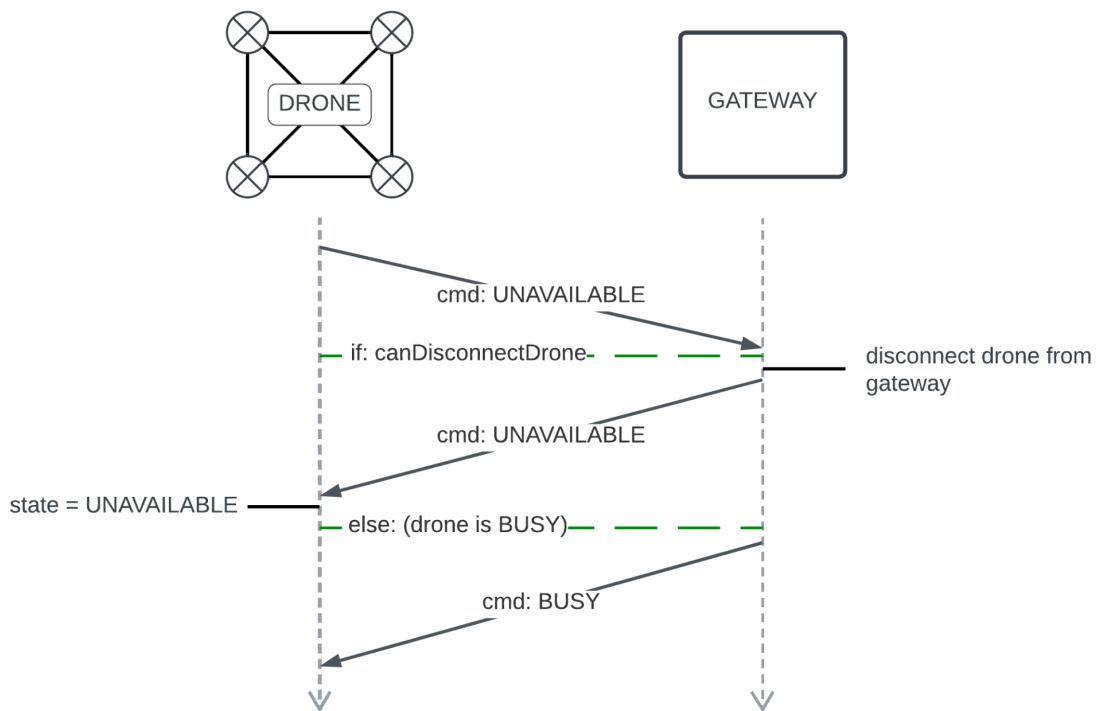
Disconnessione di un drone dal gateway

Quando un drone vuole disconnettersi dal gateway e non contribuire più alle consegne deve inviare a quest'ultimo un messaggio con il comando UNAVAILABLE.

Il gateway, se riceve questo messaggio da un drone, elimina l'identificativo del drone dalla sua memoria e non potrà più comunicare con questo.

Se il drone vorrà rendersi nuovamente disponibile dovrà reinviare un nuovo messaggio di connessione al gateway.

Anche in questo caso il drone attenderà una conferma di ricezione del messaggio UNAVAILABLE dal gateway. Il gateway potrà anche non permettere al drone di disconnettersi, per esempio perché è impegnato in una consegna. In questo caso il gateway invia un comando BUSY al drone.



Sequence diagram: Disconnect drone

Perdita di connessione o aggiornamento dei dati di un drone

Un drone, per disconnettersi correttamente, dovrà notificare al gateway la sua disconnessione secondo il protocollo.

Un drone però potrebbe disconnettersi in modo improvviso (per esempio: si scarica la batteria, si riavvia per qualche errore o non ha più una connessione disponibile).

In questo caso il gateway che non ha rimosso il drone dalla propria lista continuerà a provare a contattarlo. Nel frattempo il drone potrebbe cercare di riconnettersi, ma verrebbe rifiutato dal gateway in quanto l'indirizzo IP sarebbe già presente in memoria.

Inoltre, anche se il drone si potesse ricollegare in qualche modo, a seguito del riavvio, potrebbe aver cambiato la propria porta del socket.

Per questo motivo se un drone si disconnette senza seguire il protocollo e prova a riconnettersi inviando un messaggio AVAILABLE, il gateway osserverà che l'indirizzo IP del drone è già nella lista dei droni disponibili e lo riconnetterà. Inoltre se alcune informazioni sono cambiate, per esempio la porta, verranno aggiornate.

Il meccanismo di connessione può quindi anche essere utilizzato semplicemente per aggiornare alcune informazioni come la porta o il nome del drone. È sufficiente inviare, anche se il drone è già connesso, un messaggio AVAILABLE contenente nel campo data le informazioni aggiornate.

Comunicazione Client-Gateway

Il sistema gestisce un solo client per volta. Le comunicazioni tra client e gateway avvengono su una connessione di tipo TCP. Il protocollo TCP fornisce un servizio di trasferimento affidabile che comprende la ritrasmissione dei messaggi persi o con errori di bit. Pertanto il protocollo applicativo per la comunicazione con il client è più semplice in quanto non comprende alcun messaggio di conferma di ricezione.

Connessione di un client

Per instaurare una connessione tra client e gateway, il client invia una richiesta di handshake all'apposito socket del gateway.

Il gateway riceve la richiesta e connette il client, ma non inizia subito ad accettare comandi dal client. Occorre infatti accettare la richiesta di connessione del client dalla console del gateway. Se la richiesta viene rifiutata allora il gateway disconnette subito il client senza aver preso in considerazione alcun comando impartito dal client.

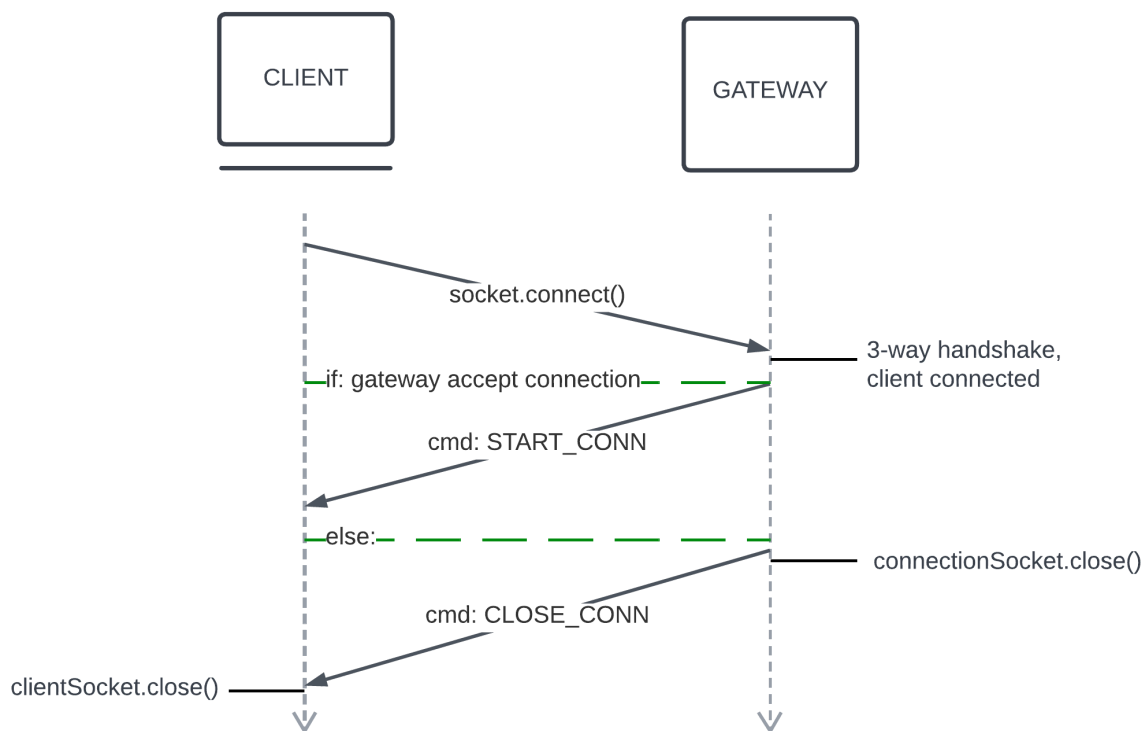
Questa scelta è stata fatta in quanto una volta connessi, solo il client può interrompere la connessione con il gateway. Inoltre, finché un client è connesso, il gateway non gestisce ulteriori connessioni con altri client.

Per ragioni di sicurezza si è ritenuto opportuno implementare un sistema di accettazione del client sul gateway.

Per notificare al client che la richiesta è stata accettata il gateway invia al client un comando `START_CONN`.

Una volta accettato un client quest'ultimo avrà il controllo del sistema finché non sceglie di disconnettersi.

Se invece il client viene rifiutato, il gateway disconnette il client e lo informa con un messaggio `CLOSE_CONN`. Il client si disconnette a sua volta chiudendo il socket.



Sequence diagram: Client connection

Ottenere la lista dei droni disponibili

Per ottenere la lista dei droni disponibili il client invia al gateway il comando LIST_DRONES. Il gateway provvede quindi a rispondere con un messaggio LIST_DRONES, includendo nel campo data la lista dei droni disponibili.

Se non vi sono droni disponibili il gateway invia un messaggio di tipo EXCEPTION, specificando nei data il motivo dell'eccezione, ovvero: non c'è alcun drone disponibile.

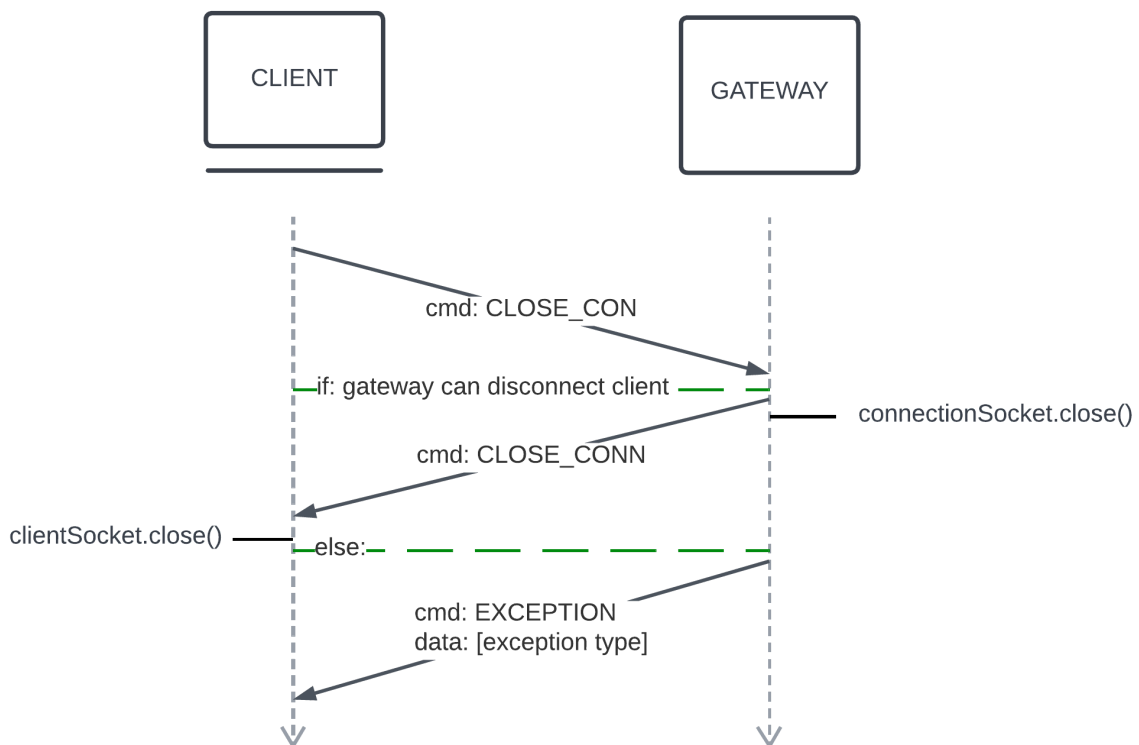
Richiesta di consegna

Inviando un comando DELIVER ed includendo nei dati del messaggio l'indirizzo di consegna e l'identificativo del drone che deve svolgerla, il client invia una richiesta di consegna al gateway. Il gateway si occuperà di verificare che il drone sia effettivamente disponibile ed inoltrerà la richiesta al drone secondo il protocollo.

Se il drone specificato non è disponibile o avviene qualche altro tipo di problema, viene inviato al client un messaggio di tipo EXCEPTION, specificando il problema nel campo data.

Disconnessione del client

Per disconnettersi, un client invia un messaggio CLOSE_CONN al gateway. Quest'ultimo risponde con un CLOSE_CONN in caso la connessione possa essere chiusa. Altrimenti invia un messaggio di tipo EXCEPTION specificando perchè non può disconnettere il client (per esempio potrebbe essere in corso una consegna richiesta dal client).



Strutture dati

Le applicazioni utilizzano alcune strutture dati per mantenere in memoria le informazioni riguardanti i droni e le consegne.

Inoltre vengono utilizzate dal gateway delle code FIFO per raccogliere i messaggi in ingresso e in uscita sulla connessione UDP con i droni.

Drone Dictionary

La classe DroneDictionary ha il compito di organizzare le informazioni dei droni nel gateway. Vengono utilizzati due dizionari, uno per i droni disponibili (AVAILABLE) ed uno per i droni occupati (BUSY). La classe fornisce i metodi per inserire e spostare i droni da un dizionario e l'altro.

Il dizionario availableDrones associa all'indirizzo IP dei droni disponibili un dizionario contenente le informazioni di quel drone (nome, porta, timestamp di quando si è reso disponibile).

Il dizionario busyDrones similmente, associa l'indirizzo IP dei droni occupati con il dizionario delle informazioni di quel drone (nome, porta, timestamp di quando si è reso occupato).

Un drone non disponibile (UNAVAILABLE) viene invece rimosso completamente dalla memoria del gateway.

Deliveries Registry

La classe DeliveriesRegistry mantiene nella memoria del gateway le informazioni riguardanti le consegne in attesa di approvazione, in corso e concluse.

Internamente DeliveriesRegistry utilizza due dizionari e una coda.

Il primo dizionario, pendingDeliveries, mantiene le informazioni delle consegne non ancora concluse.

Per ogni consegna non conclusa, identificata dall'IP del drone incaricato, viene associata ad un dizionario che mantiene le informazioni riguardanti quella consegna (stato, indirizzo di spedizione e timestamp dei passaggi da uno stato all'altro).

Gli stati per una consegna sono autoesplicativi e possono essere i seguenti:

- REQUESTED
- DELIVERING
- DELIVERED
- CANCELLED

Una consegna in corso può essere identificata solamente dall'indirizzo IP del drone in quanto se il drone è occupato non sta effettuando altre consegne.

Una consegna rimane nel dizionario pendingDeliveries finché il drone non ha consegnato il pacco. Quando la consegna viene effettuata, il gateway la sposta nel dizionario deliveryHistory.

Questo dizionario è organizzato similmente a pendingDeliveries, tuttavia una consegna conclusa è identificata dall'indirizzo IP del drone che l'ha svolta accoppiato al timestamp del momento in cui è stata consegnata.

Infatti un solo drone può aver svolto più di una sola consegna, pertanto l'identificativo per le consegne composto da solo l'IP del drone non era sufficiente.

Se invece una consegna viene annullata per qualche motivo viene trasferita nello storico delle consegne specificando come stato CANCELLED.

Si rende necessario però aggiungere una coda che mantiene le nuove richieste di consegna. Il gateway infatti ha necessità di sapere per quali consegne ancora non ha inoltrato una richiesta al rispettivo drone.

Quando una nuova spedizione viene richiesta, viene inserita sia nel dizionario delle consegne in corso (con status = REQUESTED) che nella coda delle consegne da richiedere ai droni.

Il gateway estrae dalla coda le richieste e le invia ai droni. Solamente quando il drone ha accettato la richiesta allora viene aggiornato lo stato della consegna in sospeso da REQUESTED a DELIVERING.

Questo meccanismo permette al gateway di sapere quali messaggi di conferma sta attendendo dai droni, oltre che quali richieste di consegna inoltrare.

ToSendQueue/receivedMsgsQueue

La coda receivedMsgQueue viene utilizzata dalla componente droneSideGateway per salvare i messaggi ricevuti in attesa di essere elaborati dall'applicazione. La coda toSendQueue viene invece utilizzata, sempre dalla stessa componente del gateway, per salvare i messaggi in attesa di essere inviati tramite il socket UDP.

Threads

Il software del gateway deve gestire due socket diversi: quello UDP per la connessione con i droni e quello TCP per la comunicazione con il client.

Per questo motivo sono necessari più thread per il corretto funzionamento del gateway.

Quando avviamo il gateway il main thread inizializza le due parti del gateway:

clientSideGateway e droneSideGateway.

Il clientSideGateway durante l'inizializzazione crea il socket di handshake per la connessione TCP con il client. In seguito si mette in ascolto aspettando richieste di connessione.

Quando un client si connette viene avviato un nuovo thread modellato nella classe ClientHandler. Questo thread riceve il socket della connessione instaurata con il client dopo l'handshake e gestisce lo scambio dei messaggi con il client.

La componente droneSideGateway, durante l'inizializzazione crea altri due thread. Il thread SocketOperator si occupa di gestire il socket UDP per l'invio e ricezione di messaggi dai droni. Quando riceve i messaggi li inserisce nella coda receivedMsgsQueue, mentre se ci sono messaggi nella coda toSendQueue, questi vengono inviati. Il thread SocketOperator si occupa anche di creare dei thread RetransmitTimer che si occupano di reinserire nella coda di invio i messaggi per cui non si ha ricevuto risposta entro lo scadere del timer.

Il thread della classe DroneMsgsHandler invece si occupa di elaborare i messaggi ricevuti dai droni e che sono stati inseriti nella coda receivedMsgsQueue dal thread SocketOperator.

Buffers

La dimensione del buffer è stata impostata per tutti i socket al valore 1024. Questo valore è risultato congruo e sicuro in quanto durante il testing non ha causato alcuna perdita di messaggi e al tempo stesso la trasmissione dei dati è risultata fluida all'utente.

La possibilità di utilizzare buffer bassi è resa possibile anche grazie al thread Socket Operator che periodicamente svuota i dati della connessione UDP con i droni evitando la perdita dei pacchetti e non causa quindi il riempimento del buffer stesso.

Istruzioni per l'esecuzione

Le applicazioni sono già configurate per utilizzare l'interfaccia di loopback.

Per avviare il sistema seguire in ordine i seguenti step:

- 1) Eseguire il file gateway.py
 - a) Il gateway inizia a ricevere messaggi sul socket UDP, quindi è già possibile collegare dei droni
 - b) Il gateway crea il socket di handshake, quindi è possibile inviare una richiesta di connessione da parte di un client
- 2) Eseguire il file client.py
 - a) Viene inviata una richiesta di connessione al gateway
 - b) Accettare la richiesta sulla console del gateway inserendo da tastiera il carattere Y.
- 3) Eseguire il file drone.py
 - a) Un drone viene avviato e si collega al gateway.
 - b) Al momento utilizzando l'interfaccia di loopback è possibile collegare un solo drone in quanto il gateway lo identifica tramite il suo indirizzo IP. Tuttavia se si esegue il file da altri dispositivi con indirizzi IP diversi è possibile connettere più droni al gateway. Sarà necessario modificare la variabile gatewayIP nel file drone.py in modo da connettersi all'host su cui esegue il gateway.
- 4) Sulla console del client sono visualizzati tutti i comandi e le istruzioni per utilizzare l'applicazione. In caso di necessità digitando help come comando si ottiene la lista dei comandi disponibili.