Usaremos Python Pareja 01 del grupo 2462

BO

Dr

-> send\_msg/file

-> fly/land (a través de las ET)

-> send\_file

-> get\_status (tmbn va a los drones)

-> shutdown las ET apagan sus drones

-> send\_msg/file

-> link/unlink

-> fly/land

-> disconnect

-> link/unlink

-> telemetry

-> connect

-> disconnect

ET

DR

BO

Requisitos:

-> Todas las comunicaciones deberán ser seguras (secretas y auténticas)

-> Cada actor cuenta con id alfanumérico único que puede ser establecido por el usuario o ser autmático

-> Las claves de conexión entre cada par de actores deben ser diferentes

-> Claves publico/privada que se intercambian en el registro de dos pares de actores

->Ls conexiones se inician con una generación de clave de sesión, cifrada con las claves asimétricas, solo una de las partes genera la clave.

Requisitos funcionales:

-> Las ET y BO deben ser capaces de enviar y recibir mensajes y ficheros

-> Tanto desde las ET como las BO se debe poder enviar instruccińes a los drones

-> Las ET se deben poder comunicar con cualquier número de drones y con la BO

-> Los drones pueden estar registrados a varios ET pero solo conectados a una.

-> La BO se conecta solo con las ET y están hacen de intermediario

-> Las telemetrías se envían cada dos segundos con el estado de vuelo y de la batería al menos

-> La batería de los drones deberá tener una carga de un minuto.

-> En cuanto la batería llegue al 0%, el dron deberá realizar un land

*Implementación:*

*Modelo cliente servidor, se inicia el programa de cada elemento pasando por argumento su identificador y puerto, se crea el elemento adoptando los valores introducidos y escribiendolos en el fichero de ids e intenta bindear el puerto seleccionado, tras esto, se lanza un hilo que actua como cliente donde se recoge el input del usuario y se realizan las acciones, mientras que por otro lado mediante un pool de hilos, vamos escuchando y ejecutando las respuestas a realizar como servidor.*

*Guía de uso:*

*Dron:*

*Ejecución:*

*python3 src/elementos/drone.py DRID PORT*

*Comandos:*

*--link idET Permite linkear el drone con una estación terrestre, mandando su id, puerto y clave público a la estación y como respuesta espera recibir el puerto y clave pública de la ET a la que se manda el link. para así poder realizar comunicaciones posteriormente.*

*--unlink idET Permite desvincular el drone de una ET, para ello primeramente comprueba que la ET estuviera previamente, en caso afirmativo, envía el mensaje de desvinculación a la ET y borra esta del diccionario de ETs vinculados quitando la ET conectada en caso de querer desvincularse de la misma.*

*--connect idET Permite iniciar la conexión con una ET, para ello primaremente comprueba su vinculación con el dron, en caso afirmativo, se envía la solicitud de conexión y si la respuesta es positiva, se abre un socket de conexión para la comunicación con la ET, se guarda la información de la ET conectada para finalmente lanzar un hilo encargado de mandar periodicamente la telemetría del dron.*

*--disconnect idET Permite desconectarse de la ET previamente vinculada y conectada, para ello comprueba si la idET es la de la ET conectada, en caso afirmativo, manda el mensaje de desconexión cierra el socket de abierto en la conexión y pone la Etconectada con los valores que indican la desconexión.*

*--close Cierra el programa con los hilos lanzados, para ello pone a true la variable encargada de detener los hilos y lanza una señal SIGINT que es recogida por un manejador encargado de desconectar el dron cerrando los sockets*

*Estación Terrestre:*

*Ejecución:*

*python3 src/elementos/estacionTerrestre.py ETID PORT*

*Comandos:*

*--link idBO Permite linkear la estación terrestre con una base de operaciones, mandando su id, puerto y clave público a la estación y como respuesta espera recibir el puerto y clave pública de la BO a la que se manda el link. Iniciando también el socket de conexiones con la BO y guardando la valores de la base en la variable de base linked*

*--unlink idBO Permite desvincular la ET de la BO, para ello, primeramente que el idBO corresponda la base a la que está linkeado, posteriormente manda el mensaje de desvinculación, cierra el socket de comunicación y restaura los valores que indican que no está linkeado con ninguna BO en la variable de baseLinked.*

*--fly idDron Permite iniciar el vuelo de algún dron conectado con la ET, para ello inicia comprobando que el dron se encuentra en el diccionario de drones conectados para posteriormente mandar la instrucción a partir del socket de conexión generado previamente en la conexión entre dron y estacion terrestre.*

*--land idDron Permite finalizar el vuelo de algún dron conectado con la ET, para ello inicia comprobando que el dron se encuentra en el diccionario de drones conectados para posteriormente mandar la instrucción a partir del socket de conexión generado previamente en la conexión entre dron y estacion terrestre.*

*--disconnect idDron Permite desconectarse de un dron previamente vinculada y conectada, para ello comprueba si el idDron está en diccionario de drones conectados a la estación terrestre, en caso afirmativo, manda el mensaje de desconexión cierra el socket de abierto en la conexión y borra la entrada correspondiente a ese dron dentro del diccionario de drones conextados.*

*--send\_msg dest destID msg Permite mandar el mensaje pasado por argumento a otra ET (colocando el dest 1 y en destID el id de la estación) o a la base linkeado (el dest deberá valer 0 y destID el id de la base linkeada)*

*--send\_file dest destID fileName Permite mandar un fichero cuyo nombre es a pasado por argumento a otra ET (colocando el dest 1 y en destID el id de la estación) o a la base linkeado (el dest deberá valer 0 y destID el id de la base linkeada), para ello se lee el fichero en modo binario y los bytes leidos son enviados a su destinatario quien los volcará creando un fichero con el nombre pasado por argumento.*

*--close Cierra el programa con los hilos lanzados, para ello pone a true la variable encargada de detener los hilos y lanza una señal SIGINT que es recogida por un manejador encargado de desconectar conexiones abiertas*

*Base de operaciones:*

*Ejecución:*

*python3 src/elementos/estacionTerrestre.py ETID PORT*

*Comandos:*

*--fly idDron Permite iniciar el vuelo de algún dron conectado a cualquiera de las ET conectadas a la base, para ello manda el mensaje de fly a cada una de sus ET vinculadas hasta que una de ellos nos mande una respuesta positiva o hasta terminar la lista de ET vinculadas*

*--land idDron Permite terminar el vuelo de algún dron conectado a cualquiera de las ET conectadas a la base, para ello manda el mensaje de fly a cada una de sus ET vinculadas hasta que una de ellos nos mande una respuesta positiva o hasta terminar la lista de ET vinculadas*

*--get\_status Manda a cada una de sus ET un mensaje get\_status quienes retornaran su información así como la de drones vinculados y conectados con su estado y batería.*

*--shutdown Manda a cada una de sus ET vinculadas un mensaje de shutdown las cuales a su vez mandarán el shutdown a los drones desconectandolos y haciendolos aterrizar*

*--send\_msg etID msg Permite mandar el mensaje pasado por argumento a una ET vinculada previamente cuto id sea etID con la base de operaciones.*

*--send\_file etID fileName Permite mandar un fichero cuyo nombre es a pasado por argumento a una ET vinculada previamente con id etID, para ello se lee el fichero en modo binario y los bytes leidos son enviados a su destinatario quien los volcará creando un fichero con el nombre pasado por argumento.*

*--close Cierra el programa con los hilos lanzados, para ello pone a true la variable encargada de detener los hilos y lanza una señal SIGINT que es recogida por un manejador encargado de cerrar las conexiones abiertas*