Plataforma de Drones Aéreos Multi-tecnologia para Suporte a Comunicações Críticas
State of the Art

Project in Computer and Informatics Engineering

University of Aveiro

Beatriz Rodrigues, Bernardo Falé, Diogo Correia, Francisco Martinho, João Simões, Lara Rodrigues

Group 3

December of 2021

Chapter 1

Introduction

The goal of this *state of the art* report is to present work whose domain intersects with our project. Our work can be divided into three main topics; Communications, Data persistance, and a client oriented dashboard. First, we will start by describing the UAV's and their respective autonomous control. We will also present an overview of the concept of FANET's, aswell as all the communication protocols we will use, i.e WAVE protocol and LoraWAN. An overview of the systems that we analyzed in order to start developing this work are presented below.

Chapter 2

Unmanned Aerial Vehicles

An UAV is an aircraft that is guided autonomously, by remote control, or both; it carries several sensors and technologies that allow the drone (in this case) to communicate with other entities. These autonomous flying vehicles are equiped with software and hardware that allows the collecting of the data; These blocks of data are essential to the stability of the UAV, especially when the autonomous mode is enabled. The UAVs are able to make processes faster and more flexible, while also improving the precision.They have also become more affordable. Also, most systems in which UAV's are integrated include a groundstation control module and a communication channel between both. UAV's can intervene in numerous scenarios ranging from agriculture, tourism, surveillance and sensing missions, logistics transportation, weather monitoring, fire detection, communication relaying, and emergency search and rescue. However, such new applications have a critical requirement in common, which is the need for a communications system that allows UAVs to directly connect to each other for data exchange.

2.1 Autonomous Control

The available *unmanned aerial vehicles* are capable of executing tasks without human input, namely automated tasks. Autonomy can be considered at different levels. Remotely sending an action to be performed by the drone can be considered an autonomous task, since after receiving the command, the drone moves by itself without human interference. A higher level of autonomy would include having these commands being sent automatically, without having the user to provide each one individually. In order to achieve this, research has been done to equip the drone with an OBU (on board unit) that we will address later.

Chapter 3

Flying Ad-Hoc Networks

Flying Ad-Hoc Networks (FANET's) are a subclass of Mobile Ad-Hoc Networks (MANET's) that provides wireless communications between moving UAV's, and is the concept of having a network composed by different UAV's connected in an ad-hoc. Some of the most distinct characteristics of FANET's is that the communicating nodes are highly mobile and that most of these nodes use wireless communication protocols, such as IEEE802.11(p,a,n). However, a considerable amount of problems arise with these "characteristics", such as low band-width, (relative) high connectivity time, and low transmission rates. In a FANET, drones can communicate with each other through On-board Units (OBU's), in an environment commonly described as V2V communication, and can communicate with Roadside Units (RSU's) in a V2I paradigm.

3.1 Entities

The FANET is composed by various elements:

- On-Board Unit (OBU): It is equiped on the UAV, and is responsible for the communication between drones and road-side units. The available OBU's are the PC Engines APU 2, and they are equiped with sufficient processing power to conclude any given task. These OBU's are able to connect to other interfaces through the wireless protocols previously referenced.
- Road-Side Unit (RSU): The static node in a FANET. It is usually installed near high traffic roads and is normally connected by cable or fiber to a fixed network. The three main objectives of the RSU's are to extend the communication range of the network and to provide internet connectivity to the OBU.

3.2 Domains and Characteristics

The FANET is divided in 3 communication domains :

- In-Drone domain: Connected OBU's
- Ad-hoc domain: Where V2V and V2I communications are formed
- Infrastructure domain: Connections between the RSU's and the internet

After carefully analysing FANET's, we can say that the UAV's movement is somewhat limited, although mobile, namely because of the connection constraints and several external factors, such as the weather conditions. On the other hand, the drones are equiped sensors that make possible the prediction of movements through the retrieval of critical data i.e speed, direction, and position. If there is a high number of vehicles equipped with OBU's, the network can have a large number of connected elements, and therefore, can be considered a large scale network, which is good. After the collecting of information by the UAV's, the data is ready to be transmitted, thus the ease of developing useful applications and services.

Chapter 4

Mission

A mission consists in interacting with the drone by sending high-level commands.

4.1 Mission Planning

[1] The mission planning framework aims at improving the process of building a specific mission. Using groovy, it allows the user to write a mission script with commands close to natural language, a command may be created and integrated in groovy like a plugin. In a mission script, the user can assign any number of available drones and instruct them to perform commands according to the restrictions they see fit. Multiple drones may be controlled in the same mission, either working independently or collaborating towards a goal. During said missions a drone will also send back telemetry of itself and its sensors, a senor can also be easily created and import as a plugin in the framework

Chapter 5

Communication

5.1 Wave

IEEE 802.11p(WAVE) is an approved amendment to the IEEE 802.11 standard to address the specific problems in vehicular communications, which reduces the connection setup from values of 1-2 seconds to around 10-20 milliseconds (which is very critical in vehicular environments with very low connectivity times).
It defines enhancements to 802.11 (the basis of products marketed as Wi-Fi) required to support Intelligent Transportation Systems (ITS) applications. This includes data exchange between high-speed vehicles and between the vehicles and the roadside infrastructure, so called V2X communication (V2V and V2I communications are performed using wave). This protocol can establish rapid connections (10- 20ms) and can provide up to 1000 meters of communication range, but can not reach high bandwidth values (higher than 27 Mbps).
The WAVE communication is performed with a mini-PCIe wireless card allowing communication on the 5.9 GHz band.[2]

5.2 LoRaWAN

5.2.1 Overview

The LoRaWAN® is a Low Power Wide Area (LPWA) end-to-end system architecture designed to wirelessly connect battery operated ‘things’ to the internet in regional, national or global networks with ease.

5.2.2 Features

LoRaWAN® has a variety of features that allows it to be a great solution of a wide variety of scenarios like Internet of Things (IoT), machine-to-machine (M2M), smart city & industrial applications, in those features we can include the following:

- low-cost, mobile and secure bi-directional communication
- low power consumption
- easily scalable
- supports redundant operation
- supports geolocation

5.2.3 Network Architecture

The following image encapsulates how LoRaWAN is generally used, starting from a device connected via a gateway to a much larger network where the data is then used

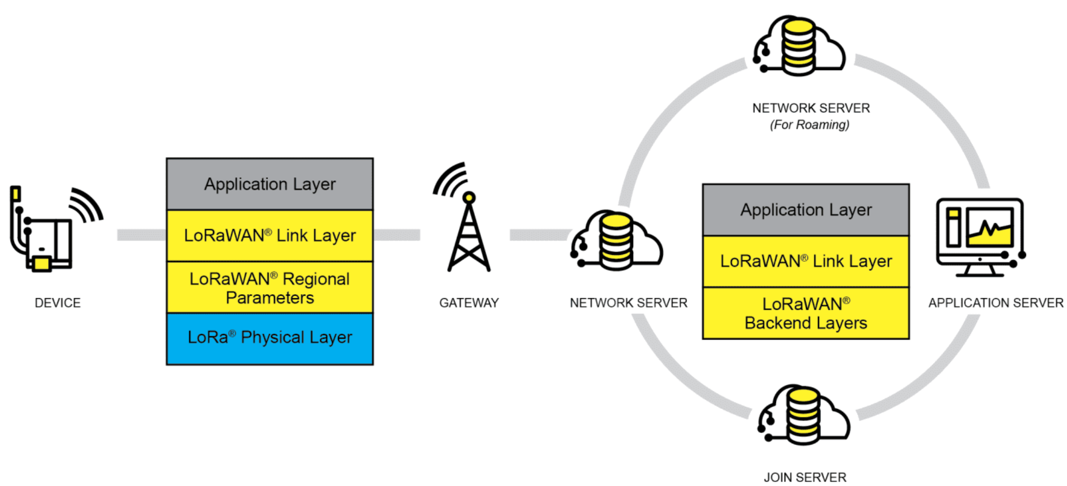

Figure 5.1: Network Architecture [3]

Chapter 6

Data Persistence

6.1 Database

The telemetry that the Drones feed the Groundstation is not kept forever. It is in fact just saved momentarily with MongoDB, and then lost whenever new information comes along. There is no data persistence.

This was good enough for the expectations set, so far, for the management of the information created on a mission. It had to be possible to watch the mission's development in real time on the Dashboard, but it didn't need to be stored for future reference, hence the "JSON-like" type of database.

Upon looking at the new objectives set for us, it is clear that the type of database has to change. The project is going towards saving the information from every mission, in order to be referred to in any time in the future.

6.1.1 Time-series Database

A relational database management system (RDBMS), like MySQL, was thought first as an alternative to MongoDB (because of its popularity, and our knowledge on databases), but then, after some research, we came to the conclusion that a time-series database (TSDB) would be the way to go.

One reason is its ability to handle time-related data, where the data is stored in "collections" that are aggregated over time. In this type of system, every point that is stored comes with a timestamp. Having in consideration our need to browse missions that occur in different points in time, a RDBMS would work, but a TSDB makes the work easier and more efficient.

TSDB are optimized for a fast ingestion rate. RDBMS on the other hand, because it has to re-index the data for it to be accessed faster, the performance tend to decrease with the size of the collection. [4]

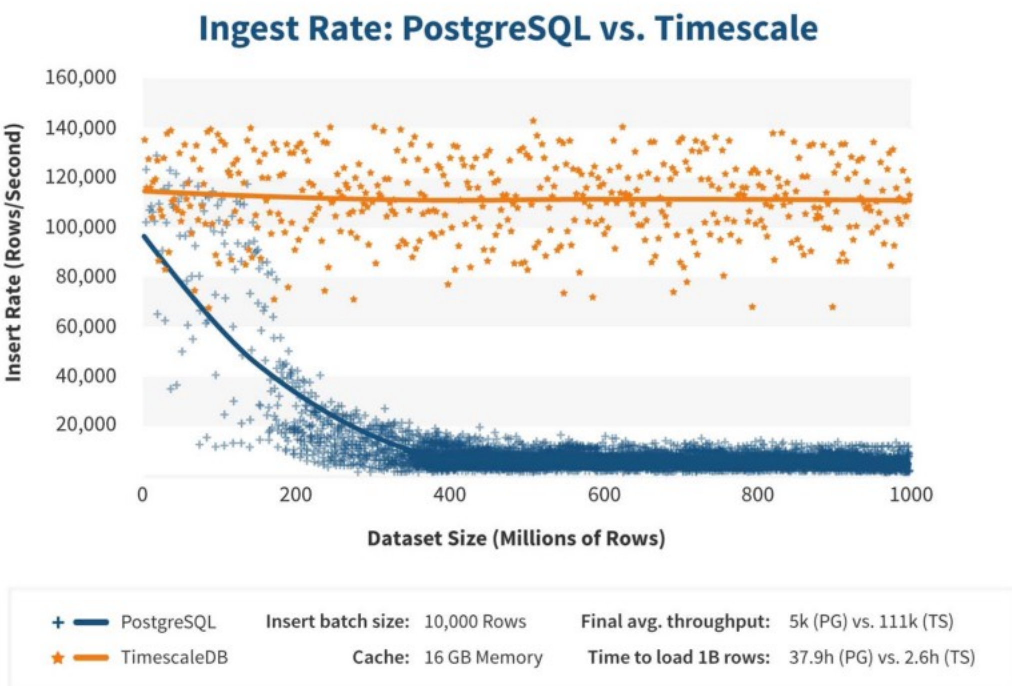

Figure 6.1: Different behaviour observed between RDBMS and TSDB [4]

6.1.2 InfluxDB

We chose InfluxDB because of its popularity among time-series databases, and also because of its easy integration with Grafana. This is important because Grafana is a tool that creates time-series data visualization to Dashboards, which is exactly what we need.

InfluxDB works with retention policy, that defines how long it keeps the data stored. This might be useful when dealing, for example, with test missions that are not meant to be archived. You could either want to check some values once and forget about it (like what is happening now with MongoDB), or save it only until next week, when you know they won't be needed anymore and would just become accumulated waste in the future, if not discarded.

6.2 API

The Application Programming Interface (API), "the connector of it all", was created with Django. It allows the Groundstation to save the telemetry from the Drones into MongoDB, and then makes it easily accessible, through routes, to the Dashboard.

We opted to use this already existing module, with the difference being the database it will be connecting to, as mentioned before.

Chapter 7

Dashboard

The dashboard is the module that has the objective to create an abstraction layer between the code behind the functionalities and the final user. It is the interface the user will use to interact with the mission. It will also give him a lot of information: values of the sensor, location, commands to control the drone, battery information, live video, and much more.

It allows the upload and tracking of missions, connect to multiple drones and ground-stations, displaying a map for easy interpretation. This dashboard also contains a section dedicated to the display and control of the drone camera quality settings, and multiple charts that illustrate the continuous status of the attached drone sensors.

The front-end is built with Nuxt.js [5], which is a JavaScript framework mainly based on Vue.js, made to create interactive SPA's (Single-Page Applications), using also HTML and CSS for the structure and visuals of the website.

The back-end uses MongoDB as a database and Django to connect to the ground-station for control and management, like described on section 6.2.

This module is very well developed, so the plan here is to continue using it and just introduce the following new features:

- Get and display as much relevant data to the user as possible from API;
- Implement the possibility to further modify the missions already running;
- Allow creating missions without needing to have programming skills;
- Others that may be considered useful throughout the project's development.

References

[1] M. Silva, *A mission planning framework for fleets of connected drones*, (accessed: 01.12.2021).

[2] A. Figueiredo, *Mobility sensing and V2X communication for Emergency Services*, (accessed: 01.12.2021).

[3] L. Alliance. "What is lorawan® specification". (), [Online]. Available: `https://lora-alliance.org/about-lorawan/`. (accessed: 01.12.2021).

[4] R. Kiefer. "Timescaledb vs. postgresql for time-series". (), [Online]. Available: `https://blog.timescale.com/blog/timescaledb-vs-6a696248104e/`. (accessed: 02.12.2021).

[5] N. Team. "Nuxt - the intuitive vue framework". (), [Online]. Available: `https://nuxtjs.org/`. (accessed: 02.12.2021).

[6] D. Amaral, G. Pereira, J. L. Costa, and J. T. Rainho, *Drone Relay - Technical Report*. 2020/2021, (accessed: 03.12.2021).