UAV as a Service: A Network Simulation Environment to Identify Performance and Security Issues for Commercial UAVs in a Coordinated, Cooperative Environment

Justin Yapp, Remzi Seker(✉), and Radu Babiceanu

Electrical, Computer, Software, and Systems Engineering,
Embry-Riddle Aeronautical University, Daytona Beach 32114, USA
{yappj,sekerr,babicear}@erau.edu
http://daytonabeach.erau.edu/about/labs/cybase/

Abstract. UAV as a Service (UAVaaS) is a proposed cloud orchestration framework aiming to provide efficient coordination and cooperation of commercial Unmanned Aerial Vehicles (UAVs). This work proposes a simulated environment to perform analysis and testing for UAVaaS integration. The environment is realized using off-the-shelf frameworks such as Flight Gear and Ardupilot's Software In The Loop to simulate real world UAV hardware, as well as web service and messaging API's such as RabbitMQ and Java Spring Framework to simulate UAVaaS cloud coordinator and client functionality. This simulation environment is devised to conduct further research into the network performance and security issues associated with UAVaaS configurations.

Keywords: UAV as a service · Unmanned Aerial Vehicles · Software Defined Networks · Network simulation

1 Introduction

Unmanned Aerial Vehicles (UAVs) have been impacting on multiple domains, including defense, business, societal, and legal domains. Although UAV technology was traditionally used in military applications, it has quickly become a practical solution for a wide range of commercial and industrial problems. Today, UAVs can be registered and certified to fly commercial operations involving structural inspections, wildlife protection, emergency management, land surveying, real estate, film production, border patrol, and agriculture, while simultaneously allowing users to upload and share visual data into the cloud [1]. As a consequence of the wide-range of solutions suggested by UAV applications, many organizations and research groups have been working on cloud-based solutions for UAV operations [2].

Increased application areas and proliferation of low-cost UAVs started posing problems for public safety and privacy. In order to address these challenges, the

J. Hodicky (Ed.): MESAS 2016, LNCS 9991, pp. 347–355, 2016.
DOI: 10.1007/978-3-319-47605-6_28

US Federal Aviation Administration (FAA) established rules and guidelines for registering and operating UAVs [3,4].

UAV as a Service (UAVaaS) is a cloud-provided, pay-as-you-go utility for commercial Unmanned Aerial Vehicles (UAVs) that eliminates the need for owning and bearing operating risks of UAVs associated with business solution integration. UAVaaS operations take place mostly in heavily populated areas with communication channels that span over multiple, wide-area networks using heterogeneous "on-the-wire" interfaces. Because of the risk of causing harm to individuals and property, the system must inherently be treated as real-time and safety critical where network reliability, security, and resiliency pose as major factors that need further investigation. Since it is not feasible to perform thorough operational testing on live systems, a simulation approach that models UAVaaS topology and behavior is taken.

2 UAV as a Service Overview

The cloud computing concept of UAV as a Service is not new, but rather analogous to the more widely known Infrastructure as a Service (IaaS) model [5]. In IaaS, third party providers host hardware, software, servers, storage, and other components on behalf of their customers. Infrastructure maintenance, backups and updates are performed behind-the-scenes, providing a hassle free, user experience. Other characteristics of IaaS include scalable resources that adjust on-demand and allow billing on a pay-per-use basis.

The IaaS concepts mentioned in previous paragraph can be mapped to UAV as a Service model where server and storage infrastructure is replaced by UAV clusters. With regards to maintenance, UAVs are repaired and serviced by trained employees of the UAVaaS provider and is performed behind the scenes. Customers who use the service will only be charged for flight time used per each UAV as well as the amount and quality of data recorded.

UAVs, while heterogeneous in design and hardware, share common network interfaces and tightly coupled software components that enable standardized communication to UAVaaS providers. This feature allows multiple vendors and manufacturers to connect customized, mission specific UAVs that can be automatically tasked and deployed into the field.

Typically, UAVs will be assigned to particular service area hubs, based on demand. Figure 1 depicts how these hubs may be located for their respective service areas. As illustrated, a particular geographical area may contain different UAV requirements across multiple domains. Three UAV hubs are implemented in order to provide sufficient coverage area for agriculture, construction, emergency management, and mining while simultaneously allowing management and news reporting entities to spectate operations from anywhere outside service area coverage.

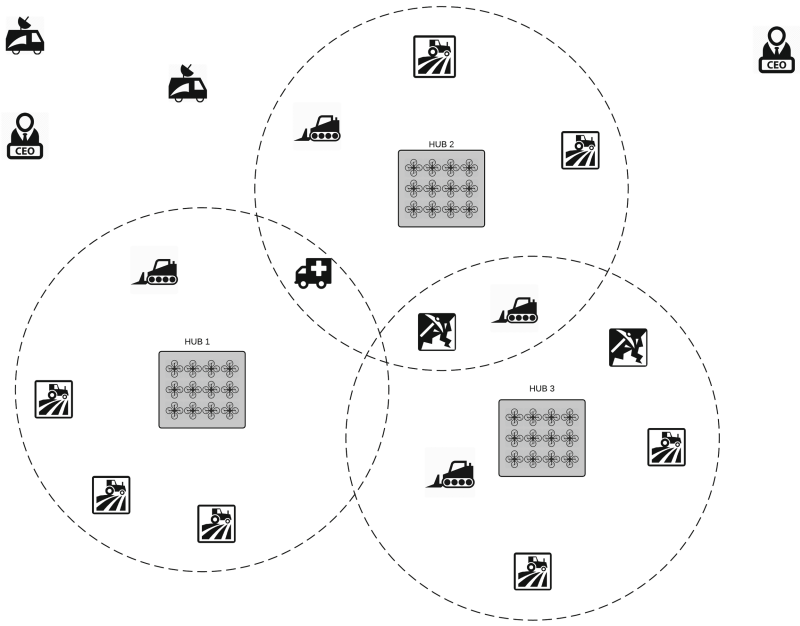

Fig. 1. A presentation of hubs for UAVaaS

3 Components of UAVaaS System and Communication Channels

We identify five primary actors who would be interacting with the UAVaaS system. These actors are operators, spectators, emergency services personnel, UAV ground crew personnel, and third party UAV vendors. Figure 2 depicts the interaction of these actors with a UAVaaS Coordinator, the primary cloud service that acts as a liaison for all operations.

Operators are the primary users who require access to UAVs to satisfy business requirements. Operators plan and control flight operations, while spectators only require access to collected data (e.g. imagery). Emergency Service Personnel represent privileged higher priority organizations such as fire departments, police departments, and other emergency first responders that re-task UAVs mid-flight or on ground for the purpose of using them in high-risk situations. For example, fire fighters trying to extinguish a major forest fire may need aerial assistance of nearby UAVs for reconnaissance, planning and detection of human life within the area. Fire fighters may need current UAVs nearby re-tasked (with current customer's permission). The ground crew maintains and operates strategically located UAV hubs. Their role is to monitor the safety of all UAVs during take-off, while en-route, and landing as well as to respond to any accidents within specified service areas. UAV Vendors represent manufacturers who supply UAVs to the UAVaaS system.

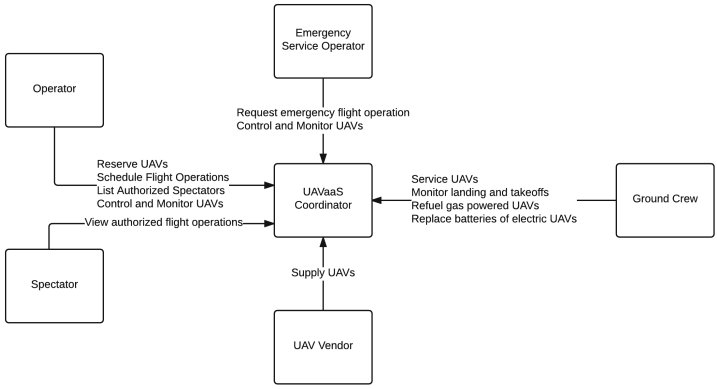

Fig. 2. High level system context diagram of primary actors

Three types of communication channels are identified in the proposed design: management, control, and data channels. All three channels pass through a centralized cloud coordinator that multicast data to authorized actors using load balancers and other message distribution techniques such as Advanced Message Queuing Protocol (AMQP). Delegating the replication and distribution of messages to a cloud coordinator, overhead and bandwidth requirements necessary for managing large amounts of spectators, controllers, and ground crew personnel for a given mission are reduced. This delegation can enable using simpler UAVs and have them use their resources on the mission.

Data messages contain telemetry, logs, and sensory information sent back to the UAVaaS Coordinator (as depicted in Fig. 3). Data messages are distributed to appropriate controllers and spectators as needed. These messages may include high resolution imagery or some environmental data, real-time altitude, speed, and location information needed by the controller to make decisions about a current mission.

Control messages are relatively light-weight and provide controllers an interface to send control signals to UAVs. Since the UAVs will be flying autonomously for the entire duration of the operation, these messages will be simple control commands for altitude and speed changes, uploading new waypoints or points of interest, or control camera and other payload features.

Management messages provide the coordination and administration properties of UAVaaS. Before any flight operation and any communication between customers and UAVs can begin, authorization, access control, mission tasking and session negotiation need to be handled. Similarly, when flight operations are finished, termination of sessions, and disassociation of UAVs, controllers and spectators must be performed.

Controllers communicate via management messages to authenticate to the UAVaaS cloud provider, create mission plans, reserve and schedule UAVs, and request specific session keys needed to communicate with UAVs. Controllers also retrieve data when a mission starts. Spectators communicate via management

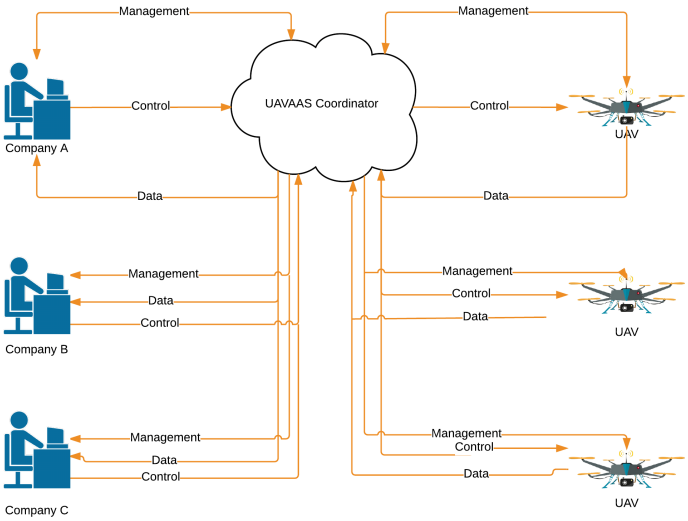

Fig. 3. Communication topology between for UAVaaS

messages to authenticate to the UAVaaS cloud provider and to retrieve session keys needed to collect UAV data by controllers. UAVs communicate via management messages to send status and availability updates to the UAVaaS cloud provider for tasking/re-tasking purposes and to retrieve session keys needed to communicate with controllers and to send collected data back.

The UAVaaS Coordinator is designed with two key services that work together to satisfy all communication requirements. Two services are needed because of differing natures of management, control, and data messages. Management messages are sent and received as a request-response pair, meaning that when a request is received by the UAVaaS Coordinator, a response is sent directly back to the sender. Such messages may include logging into the system, requesting a list of currently available mission plans, paying bills, and starting missions. Control and data messages work differently. Instead of the original sender receiving a response, the message is forwarded to its appropriate receiver(s). Such messages include control commands that are forwarded to one or more UAVs and video feeds that are forwarded to one or more spectators. Management messages can therefore be processed using a simple REST service [6], and control and data messages can be processed using a forwarding/messaging service. A third type of communication occurs in which the messaging service is dynamically configured by the management service based on requests sent to it on behalf of UAVs or operators and spectators.

4 Simulation Scope and Test Environment

The scope of the simulation comprises data producing and consuming agent clusters that represent four identified actors (UAV vendors are omitted due to

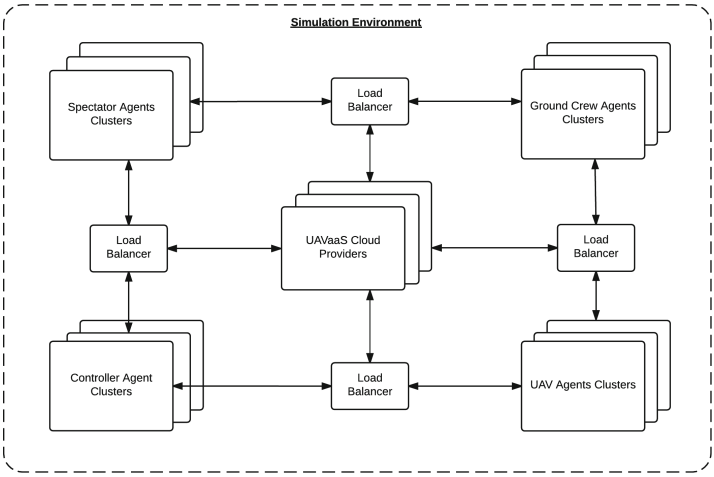

Fig. 4. Simulated network environment for UAVaaS

lack of direct interaction). By prioritizing communication channels, adjusting factors such as cluster sizes and data production rates, it is possible to study the effects on bandwidth contention and overall availability of the system. The simulated network environment is presented in Fig. 4.

The testing environment consists of 24 Dell OptiPlex desktop computers (8 GB RAM) connected to an HP ProLiant Server via a Cisco Catalyst switch that is networked in an isolated environment (*i.e.*, without connectivity to the outside world). Simulated components such as the UAVaaS cloud coordinator's messaging and management services, UAVs and controllers are developed on virtualized guest operating systems using Virtual Box where they are cloned and deployed across the network to appropriate host machines. This setup provides a consistent and controlled environment for future testing as external noise to the network is removed, each simulated device is given its own physical network card, and allowing for changes to components on the fly through cloning. The test environment is presented in Fig. 5.

5 Simulation

Three components are simulated for this testing environment: (1) UAVs, (2) UAVaaS Coordinator, and (3) Clients. Note that clients collectively include controllers, spectators and ground crew personnel, as they share common implementation to generate and retrieve data, to and from the UAVaaS Coordinator.

5.1 UAV Simulation

This component simulates real world UAVs that operate within a UAVaaS environment. Its scope is limited to a simulation of basic hardware components such

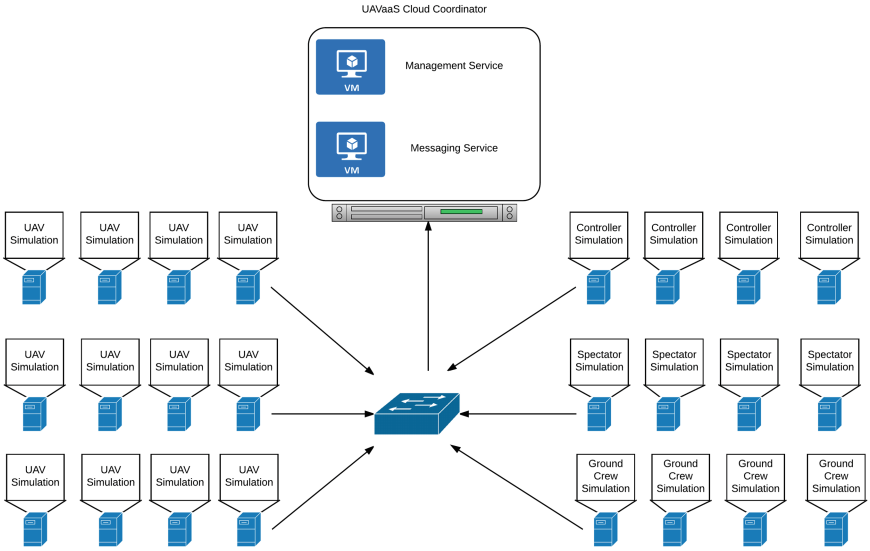

Fig. 5. Test environment for UAVaaS

as motors, GPS, altimeter, batteries, cameras, companion board software, and 4G LTE uplink and downlink. It also emulates the on-board flight controller to send and receive Micro Air Vehicle Communication Protocol (MAVLink) messages.

The UAV simulation is developed using Ardupilot's Software In The Loop Simulator (SITL). This is a complete, out-of-the-box utility implemented in C++ that emulates a quad copter's flight controller without the need for any hardware. It natively supports communication of MAVLink messages and properly simulates hardware components such as GPS and batteries. To communicate with SITL, Ardupilot's DroneKit API is used to provide a programmatic interface to control simulated UAVs. The DroneKit API is written in Python and can easily issue the simulated quad copter navigation and flight commands. To better visualize the UAV simulation, Flight Gear, an open sourced flight simulator is connected to SITL, captures flight information, and accurately displays the UAV model in a 3D environment as shown in Fig. 6. All components are neatly packaged into a Ubuntu virtual machine where it is deployed onto 12 physical host machines in the test environment.

5.2 UAVaaS Coordinator Simulation

This component simulates instances of cloud coordinators that provide management and messaging services across multiple geographical locations. Load balancers and DNS services assist UAVs to locate and contact the closest cloud coordinator service available to improve overall network performance.

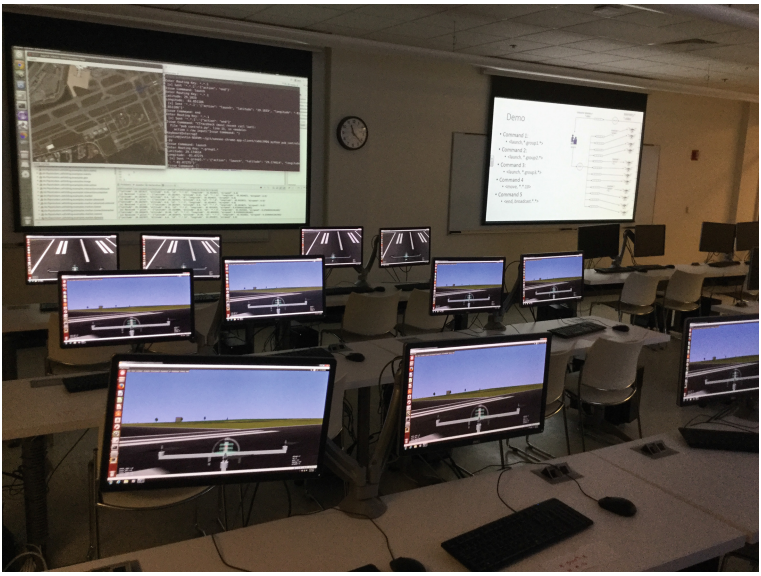

Fig. 6. UAV models running on 12 workstations controlled by the UAVaaS controller

The UAVaaS Coordinator simulation is accomplished using the Java Spring Framework to develop a RESTful web server for the management service and the RabbitMQ API for the messaging service. MongoDB is used as a lightweight NoSQL database to store user authorization and access control policy information. The RESTful web server associates with the messaging service to configure messaging sessions on behalf of users and UAV requests. All components are packaged into a Ubuntu Server virtual machine where it is deployed onto the HP Proliant Server's VMWare hypervisor.

5.3 Client Simulation

This component simulates controllers, spectators, and ground crew personnel using UAVaaS services by passing messages. Client simulation generates requests and pass and retrieve messages to and from UAVaaS cloud coordinators. These messages simulate authentication, authorization, and administrative tasks. More bandwidth intensive messages may include video streams and live telemetry updates for controllers and authorized spectators.

Simulation is accomplished using native Java libraries to make RESTful requests to the management service. The RabbitMQ API is used to subscribe to various data messages and publish control messages to appropriate UAVs. Java mapping API collects telemetry data and plots UAVs in real time as demonstrated on the overhead projector display in Fig. 6.

6 Concluding Remarks and Future Work

This research has proposed a simulated environment to perform analysis and testing on UAV as a Service integration. This is accomplished using off-the-shelf frameworks such as Flight Gear and Ardupilot's Software In The Loop to simulate real world UAV hardware, as well as web service and messaging API's such as RabbitMQ and Java Spring Framework to simulate UAVaaS cloud coordinator and client functionality. The results of future testing will be used to generate Software Defined Networking (SDN) requirements and uncover underlying network availability issues. SDN will allow for greater scalability and higher reliability of the UAVaaS network. Once implemented, UAVs (conforming to a UAVaaS specification) can be openly shared by stakeholders that wish to utilize UAV services, without the legal and financial overheads associated with traditional upfront capital expenditure.

Using this simulated environment, further study will investigate network performance at scale by simulating multiple instances of UAVaaS components as shown in Fig. 5. Important performance metrics such as bandwidth, latency and throughput will be measured and determine potential bottlenecks that exist within the proposed design. Messages will be prioritized based on type, source, destination, client type and bandwidth consumption needs to generate requirements for future Software Defined Networking integration. Security aspects such as system availability will also be analyzed by determining whether the proposed design may be vulnerable to Denial of Service(DOS) and Distributed Denial of Service (DDOS) attacks.

References

1. Yapp, J.: UAV as a service: providing on-demand access and on-the-fly retasking of multi-tenant UAVs using cloud services. MSc thesis, Embry-Riddle Aeronautical University, Daytona Beach, FL, (2016)
2. Mahmoud, S., Mohamed, N.: Collaborative UAVs Cloud, 1st edn. In: International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, pp. 365–373 (2014)
3. Civil Operations (Non-Governmental) Faa.gov (2016). http://www.faa.gov/uas/civil_operations/. Accessed 02 Apr 2016
4. Public Guidance for Petitions for Exemption Filed Under Section 333, 1st edn. Federal Aviation Administration (2016)
5. Mell, P.M., Grance, T.: SP 800–145. The NIST definition of cloud computing. National Institute of Standards and Technology, Gaithersburg, MD, United States (2011)
6. Fielding, R.T.: Architectural styles and the design of network-based software architectures. Ph.D. dissertation, University of California, Irvine (2000)