2024 20th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT) | 979-8-3503-6944-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/DCOSS-IoT61029.2024.00052

Experimental analysis of Wi-Fi-based remote control of UAVs with concurrent mission traffic

Thierry Arrabal*, Théotime Balaguer*†, Isabelle Guérin Lassous† and Olivier Simonin*
*INSA Lyon, Inria, CITI lab., France.
† Univ Lyon, UCB Lyon 1, ENS Lyon, Inria, CNRS, LIP UMR 5668, France.

***Abstract*—Communicating UAVs (Unmanned Aerial Vehicles) are promising in terms of applications, but require to address numerous research issues. In this paper, we study the impact of the quality of Wi-Fi communications on the behavior of remotely controlled UAVs. To this end, we design an experimental platform composed of PX4 Vision UAVs, a Motion Capture system, and a tool to generate mission traffic from the ground station. By defining different scenarios and different traffic configurations, we conduct a set of experiments. This allows us to analyze how the presence of mission traffic impacts the stability of the reception of the control traffic, and how, in turn, it impacts the behavior of the UAVs.**

I. Introduction

Mobile robots have been a subject of interest for decades. Recently, the use of UAVs (Unmanned Aerial Vehicles) has raised new research challenges. UAVs have many applications such as site surveillance, search and rescue missions, mapping, entertainment, military applications, etc.

With the current technological advancement, most UAVs use at least one wireless technology to communicate with ground controllers and/or with other UAVs. These communications have different purposes, such as the control of the movement of a fleet (providing directions and guidance) or its mission (for instance, the transmission of video streams for site surveillance). The quality of these communications is an essential component in ensuring the efficient deployment and control of a fleet of UAVs.

If several studies seek to evaluate the performance that can be achieved by a given wireless technology, like in [1], [2], or to optimize the network performance [3], as far as we know, no study has looked at the performance obtained when several communication flows compete within a fleet of UAVs. However, this competition is intrinsic as soon as several flows are transmitted through the fleet.

In this paper, we study the impact of the quality of Wi-Fi communications on the behavior of remotely controlled UAVs. Based on real experiments including UAVs and a Motion Capture (MOCAP) system, through two scenarios involving up to two UAVs, we study the impact mission traffic can have on control traffic generated from a remote ground station.

By designing a specific experimental setup, we can synchronize, at the same time scale, the communication packets transmitted within the mission traffic, the control traffic, and the measured poses of the UAVs from the MOCAP. The objective is to analyze the influence of the communication flows on UAV behaviors. To this end, we define different scenarios of UAVs remotely controlled via Wi-Fi technology. We then generate mission traffic with increasing throughput, between a ground station and the UAVs, while analyzing its effects on the UAV behaviors. These experiments allow us to analyze how the presence of a mission traffic impacts the stability of the reception of the control traffic, which, in turn, impacts the behavior of the UAVs.

The paper is organized as follows. Section II discusses studies that relate to our problem. Sections III and IV describe respectively the experimental scenarios and the experimental platform. Section V defines the concurrent traffic profile we generate in experimental scenarios and the different steps of an experimental run. In Section VI, after presenting the metrics, we analyze the results obtained with each scenario. Finally, we draw conclusions and some perspectives in Section VII.

II. Related work

Applications where Unmanned Aerial Vehicles (UAVs) could be used are numerous, from a single UAV taking pictures to a whole fleet coordinated to build a 3D map of an environment or structure such as in [4]. However, as mentioned in Chapter 1.3 of [5], several research challenges are still to be overcome.

In most cases, making several UAVs work together relies on communications between UAVs and a ground station. To perform these communications, the Wi-Fi

2325-2944/24/$31.00 ©2024 IEEE
DOI 10.1109/DCOSS-IoT61029.2024.00052

(IEEE 802.11) technology is a potential candidate. It can achieve a good data transmission rate (several Megabits per second) with a communication range of up to a few hundred meters [1]. Moreover, this technology is already embedded in the majority of the UAV platforms, even the smallest ones.

Many studies have investigated how to make several UAVs work in tandem to achieve complex tasks. For instance, a collaborative high-level architecture for small UAVs is described in [6]. This architecture relies on two *blocks*. The first block, *communication&networking*, is responsible for maintaining connectivity, routing and scheduling, and communication link models. The second block, *coordination* is responsible for allocating tasks and planning paths. However, no extensive studies have been done on how different types of traffic can affect each other and impact the realization of a given task.

In [7], the authors present the utilization of a fleet of UAVs for rescue missions on disaster sites where *"UAVs have to exchange flight data regularly to coordinate themselves"*, but no study has been led regarding coexisting traffic.

Few of these papers are interested in the communication capabilities of UAVs and how the communications required for the application can affect the traffic required for the control of the UAVs.

Bettstetter et al. led extensive experimental studies on UAV usage and radio communications with UAVs. Among others, [1] and [2] give experimental results on achievable performances on an 802.11 link involving one or multiple UAVs. Telecommunications metrics such as throughput and signal strength are measured for several scenarios. In [8], the authors experimentally compare the delay performance of a LTE link between a user and a LTE base station with a LTE+Wi-Fi link in which a UAV is a relay between the same user and the LTE base station.

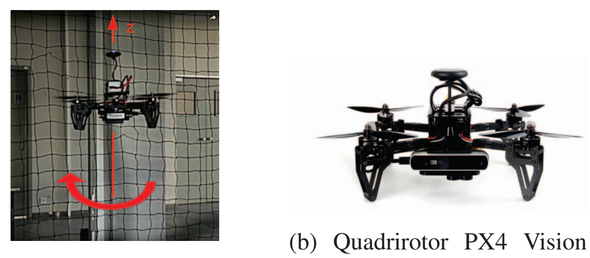
The survey [9] presents a variety of methods for mobile robot teleoperation (including UAVs). The delay, and its stability over time, is one of the main interests to consider in maintaining a functioning system. The delay might come from many different parts of the network between the robot(s) and the teleoperator(s).

While having a strong interest in the networking capabilities of a fleet of UAVs these articles do not consider the coexistence of two types of traffic. [10] mentions that specific routing algorithms are needed to bring reliability to the commands and control of a UAV network. In the current paper, we focus on studying the impact of mission traffic on command traffic. This impact is measured from a networking point of view (*e.g.,* arrival time of command packets) and also from a mission point of view (*e.g.,* yaw control accuracy).

Finally, the authors of [11] highlight the lack of an experimental dataset containing network capture files provided for research purposes. In order to participate in this scientific approach, we provide all the network data captured during the experiments presented in this paper[1], along with the code that was used to fly the drones and record the data[2]

III. Experimental Scenarios

Two scenarios, namely *simple carousel* and *double carousel*, have been experimentally studied. A carousel task consists of sending yaw values to a UAV to make it rotate around its vertical axis ($\vec{z}$ axis, see. Fig. 1a). The carousel represents a real application for a UAV like surveillance of a given area, detection of various events, cartography, etc.



(a) Carousel task illustration.

(b) Quadrirotor PX4 Vision platform.

Fig. 1: Task and UAV platform

In these scenarios, commands are sent from a ground computer to remotely pilot the UAV(s). The commands can be considered "high level" and consist of positions that the drone must follow. The low-level commands are assured by the flight controller of UAVs. In parallel, concurrent traffic is sent from the ground computer to a UAV. This traffic represents data that could be exchanged during a real mission, like a video flow, sensor data, a map to explore, etc. This concurrent traffic may affect the command traffic, necessary to remotely control the UAVs. The goal of these two scenarios is to study to which extent the concurrent traffic may impact the control of UAVs.

A. Single carousel

The first scenario is called single carousel. Commands are sent to a unique UAV triggering a rotation on its $\vec{z}$ axis while hovering (*i.e.,* no translation on any axis). In

[1]https://doi.org/10.57745/UGUWRI
[2]https://github.com/Chroma-CITI/carousel-uav-experiments

parallel to the command traffic, concurrent traffic is sent from the ground computer to the UAV, as depicted in the left part of Fig. 2.

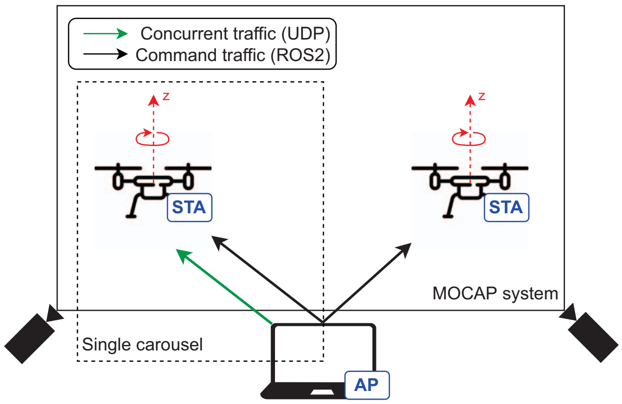


Fig. 2: The single carousel (left, dotted rectangle) and double carousel scenarios.

B. Double carousel

In the *single carousel* it might be difficult to know if the observed deviation from the nominal behavior (*i.e.,* the behavior obtained when no concurrent traffic is activated and commands are received as they are sent) comes from networking resource contention, *e.g.*, on buffers or channel bandwidth, or from the lack of computational capabilities of the UAV's onboard computer. For this second point, the mission flow consumes system resources on the UAV, which may disturb the reception of the commands.

To identify the roots of the possible perturbation on the UAV control, a second scenario, named *double carousel*, is studied. In this scenario, the commands are sent to two UAVs, and the concurrent traffic is only sent to one UAV as the destination (see Figure 2). The two drones are identical in terms of hardware and software. The control commands are identical to the ones used in the single carousel scenario, meaning that each UAV will rotate on its $\vec{z}$ axis when receiving these commands. In that scenario, only one UAV will receive and decode the two types of traffic, while the other UAV will only decode the command traffic.

IV. Experimental Platform

The UAVs used for the experiments are two PX4 Vision[3], see Fig. 1b. Each UAV is composed of two computing devices with different capabilities and functionalities. First, the onboard computer, an *UP CORE* with an *Intel® Atom™ x5-z8350* processor, runs Ubuntu 20.04. It is connected to a Wi-Fi network on which it receives setpoints (yaw values, *i.e.,* angle of rotation) from the ground control computer via ROS2 messages. These are called *commands*. The platform is equipped with small external antennas.

The second device that composes the UAVs is a *Pixhawk 4* flight controller (ESP8266-based), a board specifically designed to run the *PX4 Autopilot* open-source software. It is in charge of sensor fusion, transforming setpoints in low-level motor commands, and other tasks related to the flight. The onboard computer and the flight controller exchange data over a wired, serial connection.

The PX4 Vision is an open platform that allows different configurations. For the experiments described and analyzed in this paper, the GPS is not used in the Extended Kalman Filter (EKF2) to fly indoors. However, a MOtion-CAPture (MOCAP) system is mounted on the aviary to get precise measures on the UAV's positions. The MOCAP is composed of 8 Qualisys Track Manager Miqus cameras. All the data concerning the MOCAP go through a wired network, hence does not interfere with the Wi-Fi network used during the experiments. With this MOCAP system, the precise orientation (yaw) of the UAVs is recorded.

As shown in Fig. 2, the used network infrastructure relies on a ground computer configured to take the role of a Wi-Fi Access-Point (AP). The UAVs are associated with the AP as stations. The Wi-Fi standard used is the most recent available on the PX4 Vision, *i.e.,* 802.11g on channel 2 (no other Wi-Fi network was present on this channel during the experiments).

The commands are sent from the ground computer using ROS2[4]. ROS2 is a framework offering a publisher/subscriber architecture built over the Data Distribution Service (DDS) protocol.

V. Configurations with Concurrent Traffic

A. Experimental configuration

An experimental configuration is the combination of a scenario (namely *single carousel*, designed as S_x, or *double carousel*, designed as D_x) and a given throughput of concurrent traffic. The concurrent traffic is generated using a Constant Bit Rate (CBR) application[5] sending messages of length 1000 bytes using UDP.

For the *single carousel*, the throughput of the concurrent traffic varies from 0 Bps (no concurrent traffic) to

[3]https://docs.px4.io/main/en/complete_vehicles/px4_vision_kit.html

[4]https://docs.ros.org/en/humble/index.html

[5]https://github.com/Chroma-CITI/carousel-uav-experiments

Exp. config.	S_0	S_1	S_2	S_3	S_4	S_5	D_0	D_1
Conc. traf. throughput	0	0.25	0.5	1	10	1000	0	1000

TABLE I: Throughput of concurrent traffic in MBps for each experimental configuration

1 GBps. The throughput of concurrent traffic represents the amount of data generated at the application layer. Six throughputs have been tested corresponding to six experimental configurations with the *single carousel* scenario. For the *double carousel*, two different throughputs of concurrent traffic are studied, 0 Bps and 1 GBps. This information is summarized in Table I.

The experiments with no concurrent traffic are used as a control group. As a comparison, a Netflix video requires a throughput between 5 and 15 MBps.

B. Experimental run

An experimental run is one experiment with a given experimental configuration. An experimental run can be divided into four different phases:

- Phase 0: Taking off, starting rotation
- Phase 1: Hovering, stable rotation
- Phase 2: Transmission of concurrent traffic, rotation continues (with possible perturbations)
- Phase 3: End of the experiment, landing

The commands are sent during Phases 0, 1, and 2 with a constant rate of 10 Hz. The concurrent traffic is sent only during Phase 2 using our CBR generator.

To cope with the randomness of Wi-Fi, the dynamics of the radio medium, and real-world experimentation, three experimental runs have been realized for each given experimental configuration. A data acquisition issue lowered the number of runs to two for the double carousel. These experiments have been carried out on the experimental platform described in Sec. IV.

VI. Metrics and Result Analysis

A. Metrics

Three metrics are considered to analyze the perturbation effects that may arise due to the concurrent traffic. The first two metrics relate to the communication quality of the command traffic and are the command loss rate and the command delay. The third metric measures the quality of the UAV's behavior: the yaw rate (or rotation speed). All these metrics are explained in the following sub-sections.

Exp. config.	S_0	S_1	S_2	S_3	S_4	S_5	D_0	D_1
Loss rate	8.3	0.0	8.7	8.3	8.3	6.1	3.8	7.6

TABLE II: Command loss rates in *per thousand* for each experimental configuration

1) Command loss rate: The loss rate L is computed using Eq. 1 where l is the number of lost commands (*i.e.*, commands not received by the UAV) and r is the number of the commands correctly received at the UAV. The loss rate is averaged over 3 experimental runs of a given experimental configuration.

$$L = \frac{l}{l + r} \tag{1}$$

For all the experiments, Tab. II shows that the loss rate of the command traffic is very low for all the experimental configurations, not depending on the concurrent traffic. The command loss rate is more likely to be caused by wireless channel errors. Note that some frames containing commands may be lost or not received correctly at the UAV(s), but thanks to the Wi-Fi retransmissions, almost no command message is lost. Hereafter, we assume that the majority of the deviation from the nominal behavior is not explained only by this low loss rate.

2) Command delay: During the entire duration of each experimental run, all the packets reaching the UAV's onboard computer are saved using *wireshark*[6]. From these captures, the time separating the reception of two commands is computed using the Wireshark timestamps, resulting in an instantaneous command delay. Since the commands are sent at 10Hz from the remote ground station, this time should be, ideally, 0.1s. However, due to the Wi-Fi channel and the possible Wi-Fi retransmissions, the observed value is never perfectly stable.

The command traffic and the concurrent traffic share the network resources on the sender (*e.g.,* buffers) and the channel resources (*e.g.,* bandwidth). Sharing such resources affects the time when commands are received. To ensure a reliable remote control of a UAV, commands should be received similarly as they were produced by the remote application. In other words, the time separating the reception of two commands should be as close as possible to the time separating the creation of these commands at the application layer.

[6]https://www.wireshark.org/

3) Yaw rate ($\vec{z}$ rotation speed): The rotation speed indicates the quality of the mission execution. The remote controller sends commands to the UAV that should cause a rotation on the $\vec{z}$ axis at a constant speed of 0.4 radian/s (3.82 complete rotations per minute). The MOCAP captures the precise pose (location and orientation) of the UAV at 100Hz. From the pose measures, the yaw rate ($\vec{z}$ rotation speed) is computed using Eq. 2, where $\omega(t)$ is the instantaneous yaw rate at time t, θ_t is the yaw angle at time t and $\Delta T = 0.1s$

$$\omega(t) = \frac{\theta_t - \theta_{t-1}}{\Delta T} \tag{2}$$

In optimal conditions (corresponding to the nominal behavior), the yaw rate would be constant as the UAV receives commands at the same pace they are sent. However, the presence of concurrent traffic disturbs the reception of the command traffic.

Section VI-C focuses on analyzing how the instability of the yaw rate is correlated to the command delays.

B. Example of an experimental run

Fig. 3 gives a visual idea of the behavior of the system during one experimental run with the experimental configuration labeled S_5 (see Tab. I). The figure shows, on the same time scale, the measured yaw angle (blue line) and the command delay (orange line). In the beginning, from $t_0 = 0s$ to $t_1 = 15s$, the command delay suffers very few disturbances and corresponds to the sending rate of 10Hz and the yaw value describes a smooth rotation (ie. steady increase of the yaw). Starting from $t_2 = 15s$ and until $t_3 = 30s$ the concurrent traffic is sent in parallel to the command traffic. One can see the command delay is impacted which causes fits and starts on the UAV behavior, those can be observed on the yaw value. Finally, after $t_4 = 30s$ the concurrent traffic is stopped and the system returns to its normal behavior.

C. Results analysis

As shown in Fig. 3, the presence of concurrent traffic has a clear impact on the drone rotation. The results correspond to one experimental run with the experimental configuration S_5. It shows the disturbance caused by command delay on the yaw when the concurrent traffic occurs (from $t = 15s$ to $t = 30s$). On the other hand, the nominal behavior is shown in the *No congestion* scenarios. For instance, the *No congestion* scenario of Fig. 5 shows that the achieved yaw rate is very close to the command.

The observed behavior can be explained by analyzing the command delays. The variations of the delay are measured using the standard deviation. For each experimental configuration, the average command delay and the corresponding standard deviation are reported in Tab. III for *single carousel* scenarios and in Tab. IV for *double carousel* scenarios. Both averages and standard deviation are computed for Phase 1 and Phase 2 (see Sec. V-B).

Regarding all experiments, the standard deviation of command delays in Phase 1 remains low, while during Phase 2, it increases with the throughput of the concurrent traffic. It goes up to 0.306 in the S_5 experimental configuration, which represents a rise of more than 300% when compared to the worst case reached in Phase 1.

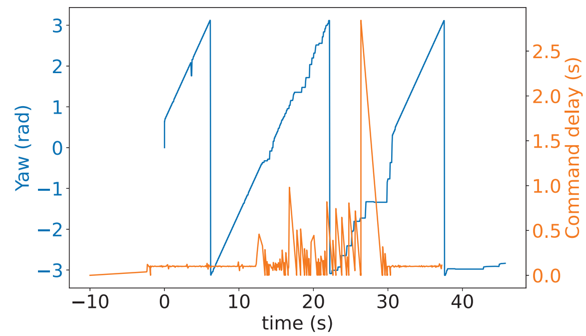


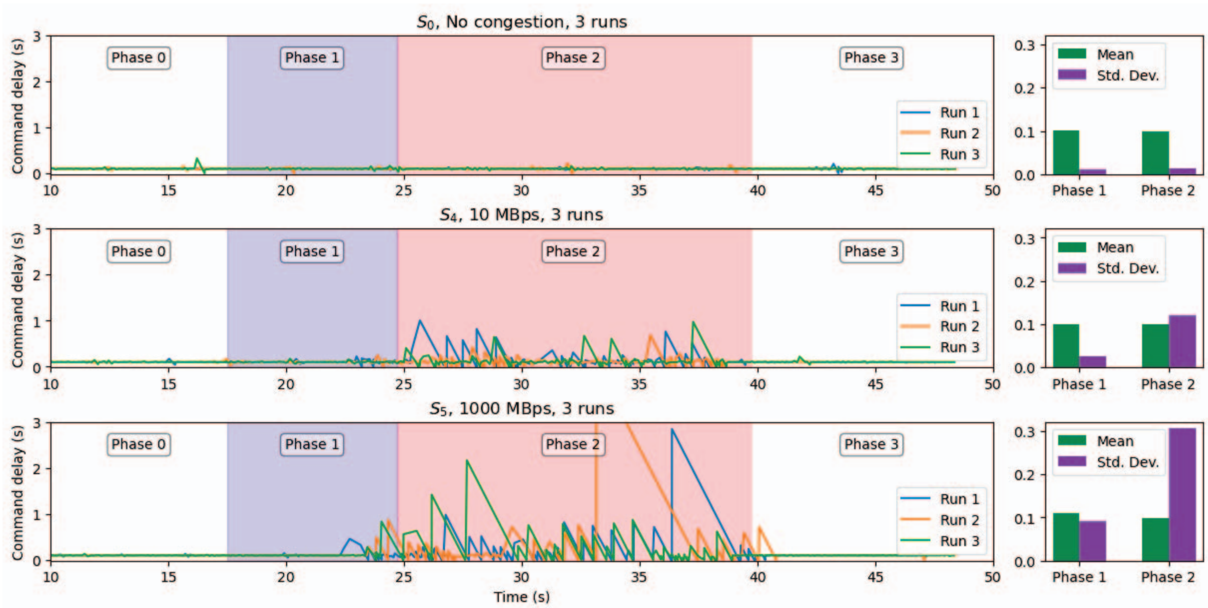
Fig. 3: Yaw value (in blue) with corresponding command delays (in orange) for one experimental run with the experimental configuration S_5 with perturbation between t=15s and t=30s.

	Phase 1		Phase 2	
	mean	std	mean	std
S_0	0.100	0.012	0.100	0.014
S_1	0.102	0.024	0.100	0.066
S_2	0.100	0.016	0.098	0.075
S_3	0.100	0.015	0.010	0.097
S_4	0.100	0.025	0.100	0.120
S_5	0.111	0.091	0.100	0.306

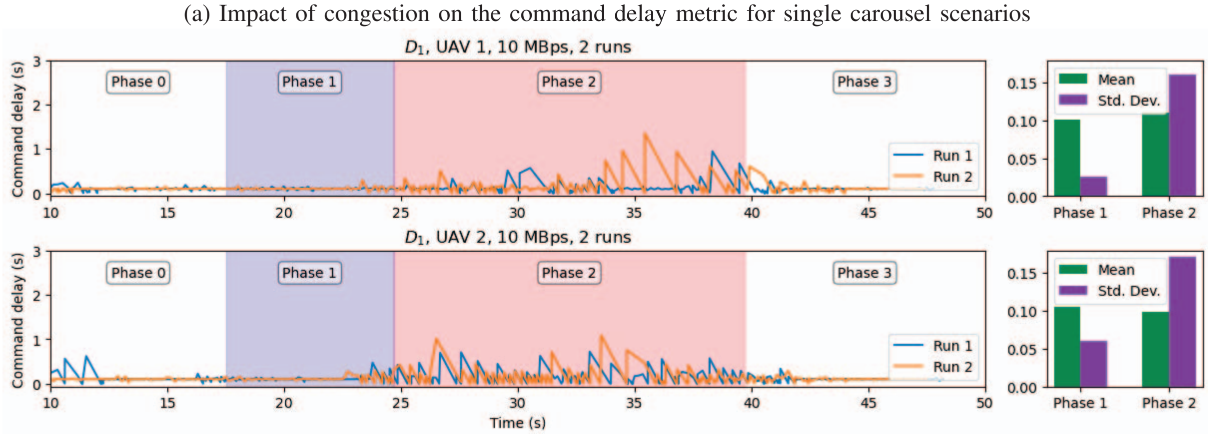
TABLE III: Single carousel scenario: average command delays and standard deviations in seconds

		Phase1		Phase 2	
		mean	std	mean	std
D_0	UAV 1	0.100	0.011	0.100	0.024
	UAV 2	0.099	0.017	0.100	0.026
D_1	UAV 1	0.105	0.048	0.106	0.142
	UAV 2	0.111	0.093	0.097	0.188

TABLE IV: Double carousel scenario: average command delays and standard deviations in seconds



(a) Impact of congestion on the command delay metric for single carousel scenarios



(b) Impact of congestion on the command delay metric for double carousel scenarios

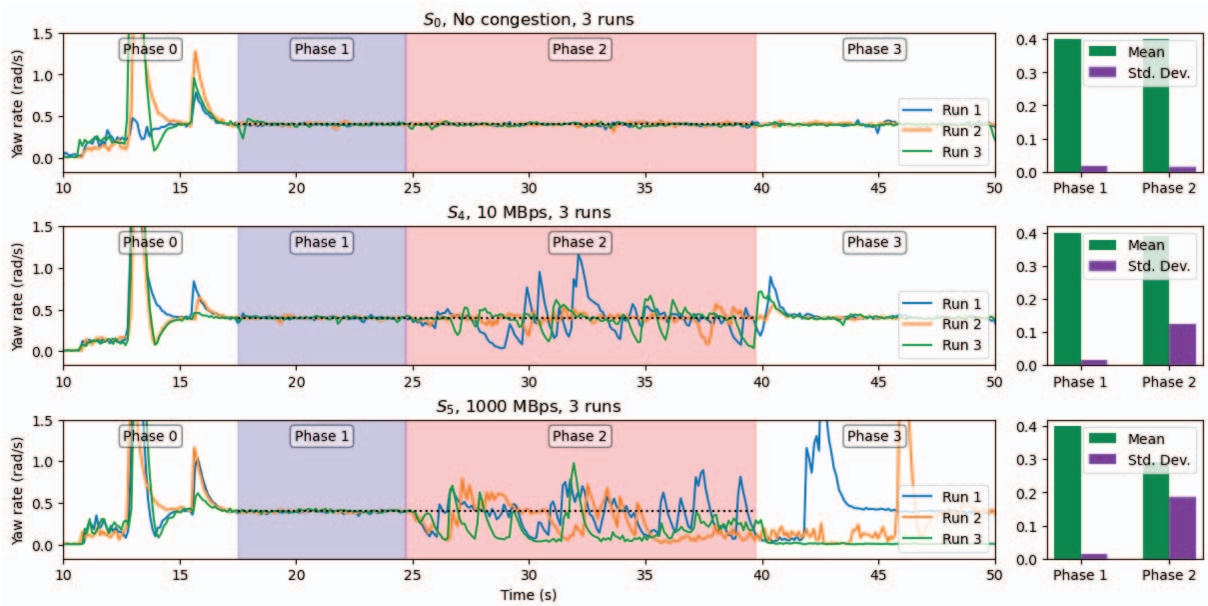
Fig. 4: Command delays

Fig. 4 gives a visual overview of the results obtained for command delay with both scenarios. Fig. 4.a shows 3 runs of the *single carousel* scenario for 3 different concurrent traffic throughput (S_0, S_4 and S_5). Fig. 4.b shows 2 runs of the two UAVs for the *double carousel scenario* (D_1). These figures also show the mean delay and standard deviation of the delay during the two interesting phases (*i.e.,* Phase 1 w/o congestion, Phase 2 with congestion).

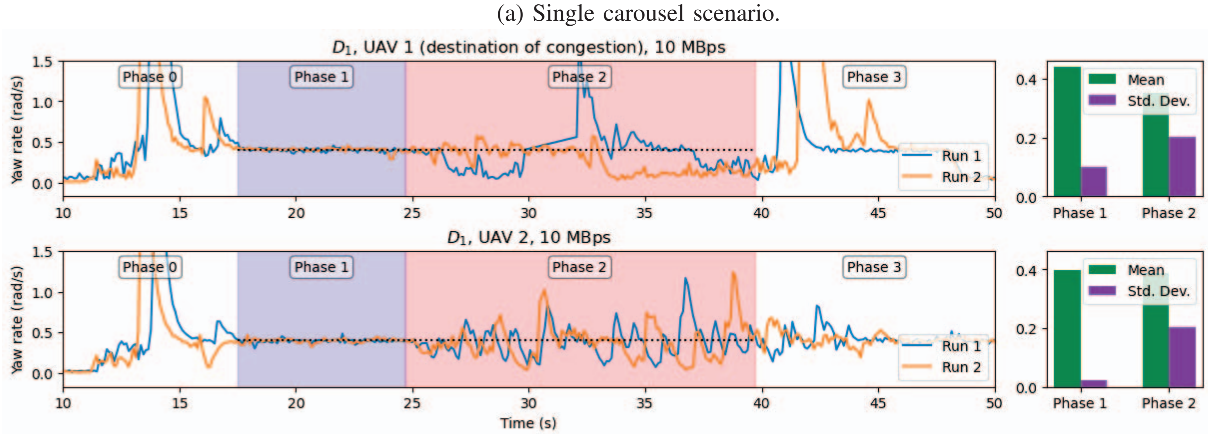
The stability of the mean delay can be explained by the loss rate that is very low (cf. Tab. II). Intuitively, if (almost) every command reaches the UAV, for the total duration of an experimental run, the average command delay matches the expected 0.1s. Despite the stability of the mean value, the concurrent traffic causes irregularities in the arrival time of commands as shown in Fig. 4.

These irregularities have a direct impact on the mission execution. Not receiving the commands at a regular pace causes irregularities in the UAV rotation speed, decreasing the quality of the mission execution. This phenomenon can be observed both in Fig. 5 and Tables V and VI for *single* and *double carousel* scenarios.

In Fig. 5b, a straight segment is observed in Run 1 of UAV 1. This is due to a capture error. Hopefully, this does not have a significant effect on the result of exploitation.



(a) Single carousel scenario.



(b) Double carousel scenario.

Fig. 5: Yaw rate

As mentioned in Sec. VI-A3, the UAV should have a stable rotation speed of 0.4 rad. per second. In Fig. 5, during Phase 1, the yaw speed is stable around 0.4 rad. per second with a standard deviation going from 0.012 to 0.034. With no concurrent traffic, the mission can be correctly executed. Starting Phase 2, the yaw rate becomes more erratic. During this phase, we can visually observe the UAV doing some fits and starts and not rotating at constant speed anymore. The standard deviation of the yaw rate during Phase 2 goes up to 0.204 in the D_1 experimental configuration, which represents a rise of 600%. Two videos, illustrating this behavior during the experiments are provided: one here[7] for the *simple carousel* and a second one here[8] for *double carousel*

During the experimental runs D_1, there is no clear difference between UAV 1 and UAV 2 although only UAV 1 is the destination of the concurrent traffic. Both UAVs are impacted similarly, either considering the command delay (Fig. 4b) or the yaw rate (Fig. 5b). This observation indicates that the concurrent traffic can deteriorate the execution of the mission of any UAV present on the network. Based on this experiment, the

[7] https://team.inria.fr/chroma/files/2024/02/singleB-3.mov
[8] https://team.inria.fr/chroma/files/2024/02/doubleB-1.mov

	Phase 1		Phase 2	
	mean	std	mean	std
S_0	0.400	0.019	0.399	0.016
S_1	0.407	0.034	0.389	0.115
S_2	0.399	0.014	0.409	0.125
S_3	0.397	0.018	0.394	0.118
S_4	0.399	0.015	0.389	0.125
S_5	0.399	0.012	0.291	0.185

TABLE V: Single carousel scenario: average yaw rates and standard deviations in rad/s

		Phase1		Phase 2	
		mean	std	mean	std
D_0	UAV 1	0.412	0.040	0.402	0.040
	UAV 2	0.413	0.051	0.404	0.036
D_1	UAV 1	0.441	0.101	0.353	0.204
	UAV 2	0.397	0.024	0.386	0.202

TABLE VI: Double carousel scenario: average yaw rates and standard deviations in rad/s

limiting resources seem to be network resources (buffer at the remote controller and/or channel bandwidth) rather than computational resources on the UAV receiving the concurrent traffic.

VII. Conclusion

We conducted experimental work and studied the impact of networking conditions on remotely controlled UAVs. The concurrent traffic has a direct impact on the execution of a simple choreography ($\vec{z}$ axis rotation). From a communication point of view, the presence of mission traffic concurrent with the command traffic causes clear delays in the command's receptions. These delays cause instability of the yaw rate. Using the comparison between two scenarios (*single* and *double* carousel), we concluded that the limiting resources are network resources rather than computational capabilities when sending commands and concurrent traffic through the same network. However, there exists a large variety of UAV hardware, and hence, the lack of resources may also come from computational capabilities.

In future works, we plan to study traffic congestion in more complex tasks such as video transfer, sensor data collection, and 3D mapping. We aim to examine communication limits in a scenario involving data exchanges between a ground station and UAV(s) and also between UAVs. Moreover, we are also interested in studying the impact of various communications protocols on drone behavior.

VIII. Acknowledgements

This work was funded by CONCERTO (ANR-20-ASTR-0003). We thank Isabelle Fantoni, Kojo Anyinam-Boateng (ARMEN team, LS2N Lab in Nantes, France), and Sonia Ortega Espin who helped with experiments.

References

[1] E. Yanmaz, R. Kuschnig, and C. Bettstetter, "Channel measurements over 802.11 a-based uav-to-ground links," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*. IEEE, 2011, pp. 1280–1284.

[2] S. Hayat, E. Yanmaz, and C. Bettstetter, "Experimental analysis of multipoint-to-point uav communications with ieee 802.11 n and 802.11 ac," in *2015 IEEE 26th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*. IEEE, 2015, pp. 1991–1996.

[3] R. Grünblatt, I. Guérin Lassous, and O. Simonin, "Leveraging antenna orientation to optimize network performance of fleets of uavs," in *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2020, pp. 253–260.

[4] A. Renzaglia, J. Dibangoye, V. Le Doze, and O. Simonin, "A common optimization framework for multi-robot exploration and coverage in 3d environments," *Journal of Intelligent & Robotic Systems*, vol. 100, pp. 1453–1468, 2020.

[5] W. Saad, M. Bennis, M. Mozaffari, and X. Lin, *Wireless communications and networking for unmanned aerial vehicles*. Cambridge University Press, 2020.

[6] E. Yanmaz, M. Quaritsch, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, "Communication and coordination for drone networks," in *Ad Hoc Networks: 8th International Conference, ADHOCNETS 2016, Ottawa, Canada, September 26-27, 2016, Revised Selected Papers*. Springer, 2017, pp. 79–91.

[7] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner, "Networked uavs as aerial sensor network for disaster management applications," *e & i Elektrotechnik und Informationstechnik*, vol. 127, no. 3, pp. 56–63, 2010.

[8] I. Donevski, C. Raffelsberger, M. Sende, A. Fakhreddine, and J. J. Nielsen, "An experimental analysis on drone-mounted access points for improved latency-reliability," in *Proceedings of the 7th Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, ser. Dronet'21. New York, NY, USA: Association for Computing Machinery, 2021, p. 31–36. [Online]. Available: https://doi.org/10.1145/3469259.3470489

[9] M. Moniruzzaman, A. Rassau, D. Chai, and S. M. S. Islam, "Teleoperation methods and enhancement techniques for mobile robots: A comprehensive survey," *Robotics and Autonomous Systems*, vol. 150, p. 103973, 2022.

[10] O. K. Sahingoz, "Networking models in flying ad-hoc networks (fanets): Concepts and challenges," *Journal of Intelligent & Robotic Systems*, vol. 74, pp. 513–527, 2014.

[11] T. Arrabal, M. Stojanova, I. G. Lassous, and J. Picot, "Experiment-driven platform for link quality estimation in ieee 802.11 wlans," in *2023 IEEE 24th International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 2023.