Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 81 (2024) 195–204

Transportation Research Procedia

www.elsevier.com/locate/procedia

13th International Conference on Air Transport – INAIR 2024, Challenging the Status Quo in Aviation

Analysis of API-Based Communication Performance for drone's operation in U-Space

Neno Ruseno[a], Fabio Suim Chagas[a], Miguel-Angel Fas-Millán[b], Aurilla Aurelie Arntzen Bechina[a,*]

[a]*Faculty of Technology, Natural Sciences and Maritime Sciences Department of Science and Industry systems University of South-Eastern Norway, Campus Kongsberg, 3616, Norway*

[b]*Institute of Flight Guidance, German Aerospace Center (DLR), Braunschweig, Germany*

Abstract

The increasing number of drone's flights is posing several challenges such as airspace safety and particularly in urban and restricted areas. Safety could be enabled by considering several types of geo-zones allowing or preventing drone's operation. In addition, ensuring reliable and efficient communication between drone operators and the U-space service providers represents an important challenge to address and more specifically in situations of unauthorized intrusion into restricted airspace. This paper presents some preliminary results of a SESAR funded Project AI4HyDrop project (An AI-based Holistic Dynamic Framework for safe Drone Operations in restricted and urban areas) which aims to integrate drones safely into controlled airspace.

The research study aims to analyze the latency and throughput in API-based communication to send intrusion alerts, using a developed algorithm to simulate the sending and receiving of these notifications. The results show that shorter time intervals (10 ms) significantly impact latency and throughput, suggesting that the system begins to deteriorate near the limit. However, the effect of message payload size and multiple systems broadcasting warnings was minimal. The overall finding suggests that API-based communication system can transmit drone detection warnings with sufficiently low latency as required in the current requirements of drone operations.

Peer-review under responsibility of the scientific committee of the 13th International Conference on Air Transport – INAIR 2024, Challenging the Status Quo in Aviation

Keywords: Drone detection; Latency Analysis; Throughput Analysis; Airspace Security; UAV intrusion Detection; API-based communication

* Corresponding author. Tel.: +4731008910.
E-mail address: Aurilla.Aurelie.Arntzen@usn.no

Peer-review under responsibility of the scientific committee of the 13th International Conference on Air Transport – INAIR 2024, Challenging the Status Quo in Aviation
10.1016/j.trpro.2024.11.021

1. Introduction

The increasing use of drones, or Unmanned Aerial Vehicles (UAVs), in various commercial, recreational, and public safety activities has brought new challenges related to airspace security, particularly in urban and restricted areas. The AI4HyDrop project (An AI-based Holistic Dynamic Framework for safe Drone Operations in restricted and urban areas) emerges as a response to these challenges, aiming to safely and efficiently integrate drones into controlled airspace. Among the project's objectives are the assessment and quantification of safety levels in U-space operations, the definition of a safety framework for strategic and tactical operations, the determination of the acceptable level of automation and AI integration, the optimization of airspace usage, and the establishment of a rational flight approval process (SESAR 3 JU, 2023).

In this context, efficient communication between drone detection and U-Space Service Provider (USSP) systems, who oversee drone operators, is crucial to achieving the AI4HyDrop objectives, ensuring that operations in sensitive areas are conducted safely and reliably. In particular, the rapid and accurate detection of drone intrusions into restricted airspace, followed by immediate communication to the operator, is essential to mitigate potential risks.

There are several research addresses the communication used in drone operations. Raffelsberger et al. (2019) developed a tool for evaluating the performance of drone communications over 4G cellular networks, focusing on measuring throughput and latency. This study provided valuable insights into communication performance in real-world environments, essential for developing communication protocols in multi-drone systems. While Cruz et al. (2024) explored secure communication in the context of the Internet of Drones, using elliptic curve cryptography (ECC) to manage keys securely on IoT devices, including drones. Implementing ECC in drones proved to be a cost-effective and time-efficient solution, offering secure and reliable communication in IoT networks.

Kagawa et al. (2017) investigated a latency-guaranteed multi-hop wireless communication system specialized for the remote control and telemetry of drones and robots in beyond-line-of-sight (BLOS) operations. The research identified that wireless LANs, commonly used for robot control, must be more suitable for BLOS operations due to unstable transmission latency and interference issues. The authors developed a prototype system that ensures reliable and low-latency communication to mitigate these limitations, as demonstrated through field tests. This approach is particularly appropriate in scenarios where fast and stable communication is critical for the safe control of drones in disaster response missions, infrastructure inspection, and logistics. However, Ruseno and Lin (2023) evaluated a communication of transferring the network Remote ID data from USSP system that provides network identification service to the UTM application used by drone operator using API-based communication via internet. The authors conducted a flight test to analyse the performance of the system and algorithm.

Cropf et al. (2022) introduces a conceptual implementation of a System Wide Information Management (SWIM) architecture within a proposed Unmanned Aircraft System (UAS) Traffic Management (UTM) framework for the U.S. National Airspace System (NAS). The system collects messages from various data producers via FAA SWIM, extracts relevant attributes, stores them in a database, and provides flexible geospatial, temporal, and attribute-based filters for consistently correlated SWIM data retrieval. The data includes (but is not limited to) airspace constraints, traffic advisories, weather, and low-altitude hazards.

Shivakoti et al. (2021) explored a model for reliable communication between multiple drones and a web-based Air Traffic Control (ATC) system. Through simulations of various scenarios involving multiple drones flying in an urban environment, the study showed that the Message Queuing Telemetry Transport (MQTT) protocol holds significant potential for enabling communication among multiple drones within a network. In these simulations, four drones communicated over a single network using limited hardware, specifically Raspberry Pi devices integrated with ThingsBoard.

These studies demonstrate the diversity of approaches to addressing communication challenges in drone operations, from developing new protocols that enhance performance to creating specific communication systems for critical operations requiring low latency and high reliability. Within the project, the drone detection system considers cooperation and non-cooperation drones which could violate a restricted airspace. Cooperative drones are assumed to be equipped with a broadcast Remote ID that transmits its identification, operator, mission, and position via Bluetooth or Wi-Fi technology that can be received by devices that have Bluetooth or Wi-Fi capabilities in the vicinity that could be varied in range maybe 800 feet, maybe a mile, or it might be five miles depends on the

hardware used and the situations. While non-cooperative drones do not broadcast any of their information, but only can be detected from their signatures such as visual, thermal, audio, or radio frequency.

In this paper, the scope is on the communication channel between the drone detection service and the USSP system, and only cooperative drones are considered that enable the communication to the drone operator via the USSP. The other systems shown in the drone detection framework are assumed to be in place and support this communication. Also, internet communication is assumed to use the current internet protocol that is widely used in our daily life based on TCP/IP protocol.

The main challenges related to communication between systems are maintaining consistent, reliable communication performance and avoiding its degradation. Communication performance can suffer interference due to buildings and other types of signals that travel in the same airspace, leading to an increase in latency and a decrease in throughput (Zreikat and Mathew, 2024). Communication degradation can occur in adverse situations, such as unfavorable weather conditions and in rural areas, due to an infrastructure lack (Jesús-Azabal et al., 2023). In these situations, data loss and disconnection problems may occur, leading to aircraft loss of control.

The study's overall objective is to analyze the communication performance between drone detection system and USSP system ensuring that intrusion notifications are transmitted securely and efficiently. IMore specifically, within AI4HyDrop project, the German Aerospace Center (DLR) U-Space Research Environment (DURE) is used as the USSP system. Moreover, the result will be compared to the other research and the requirement of regulation to check that the communication performance meets the requirement of U-Space operations. To this end, an algorithm was developed and tested that simulates the sending and receiving of drone intrusion alerts in restricted airspace. It was designed to measure latency and throughput as the system's ability performance to handle multiple simultaneous requests. This work's contributions include presenting a methodology to evaluate the latency and throughput of communication systems in critical air security scenarios. Additionally, the results obtained can serve as a basis for developing standards and best practices for implementing communication systems in operations involving drones, ensuring that quick and effective responses can be provided in intrusion situations.

The remainder of this paper is organized as follows: Chapter 2 describes the adopted methodology, detailing the drone detection framework in AI4HyDrop project and the development of algorithm used to simulate and evaluate communication between drone detection system and USSP system. Chapter 3 presents the experimental results obtained, including the analysis of latency and throughput. Chapter 4 discusses the key findings from these preliminary experimental results. Finally, Chapter 5 presents the conclusion, highlighting the main contributions of the work and suggesting directions for future research.

2. Methodology

2.1. Drone Detection Framework

The proposed framework of drone detection in the AI4HyDrop project can detect cooperative and non-cooperative drones using AI-based detection algorithms as shown in Figure 1. The drone detection service employs various sensors such as a camera, microphone array, and radio frequency antenna to detect all drones near the restricted airspace as the input for the AI-based detection algorithms. Since the algorithms used are normally the deep learning type, they required a model trained using a large dataset to accommodate a variety of drone types.

In the proposed framework, the drone detection service could be part of U-Space Service Provider (USSP)'s services or other third-party services that are connected to a USSP. USSP is an entity in the U-Space system that provides services as mentioned in the regulation to UAS operators to support their operations as stated in Barrado et al. (2020). There is also another U-Space entity mentioned in this framework, the Common Information Service Provider (CISP). In our study, the extended CISP terminology is used because the current regulation does not include the service of finding a drone flight plan (U-Plan) as part of CISP capabilities as described in EASA (2024).

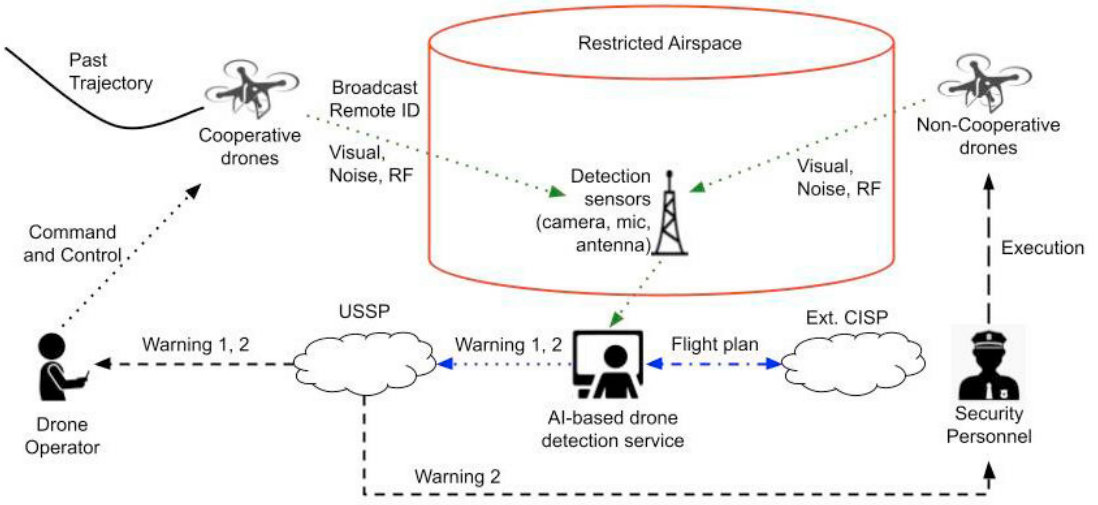


Fig. 1. Drone detection framework.

There are three case studies of drone detection identified in this research that consists of:

1. Cooperative drone that is authorized to fly into restricted airspace:
   - A drone is detected by sensors flying near a restricted airspace and its location is estimated.
   - The broadcast remote ID data is received including its ID and location.
   - The detection system connects to the Extended CISP to get the flight plan authorization data.
   - The data received confirms that the drone is authorized to fly in restricted airspace and the case is closed.
2. Cooperative drone that is not authorized to fly into restricted airspace:
   - The drone is detected by sensors flying near a restricted airspace and its location is estimated.
   - The broadcast remote ID data is received including its ID and location.
   - The detection system connects to the Extended CISP to get the flight plan authorization data.
   - The data received confirms that the drone is NOT authorized to fly to the restricted airspace
   - A warning level 1 (including drone ID and its location) is sent to the USSP to notify the operator to stay away from the restricted airspace.
   - The drone operator commands the drone to return to its planned trajectory and the case is closed.
   - If the drone continues to be nearer to the restricted airspace, it becomes a non-cooperative drone (case 3).
3. Non-cooperative drone that flies into restricted airspace:
   - A drone is detected by sensors flying near a restricted airspace and its location is estimated.
   - No broadcast remote ID data is received or the non-cooperative drone case from case 2.
   - A warning level 2 (including drone location) is sent to the USSP to notify all the operators in the area, trigger any tactical deconfliction measure and warn the security personnel to take the necessary actions.

2.2. API-based Communication

The mechanism of transferring warning information to the USSP has a significant role in the drone detection framework because the information should be transferred as quickly as possible to the related parties to take necessary action to avoid a further catastrophic event. In this research, it is assumed that the detection system and USSP are connected by internet communication that is available in most countries using Application Programming Interface (API) to exchange the information such as warning from drone detection system. An API describes how

users (also called clients) requests information of actions to a server. These requests usually need to determine the values of some parameters, for instance, the date when booking a flight or username and password.

The proposed warning message submitted through the API is in the form of JavaScipt Object Notation (JSON) format as shown in Figure 2. JSON is a widely used format to gather and send the aforementioned pairs of names of parameters and its values in a readable way. In our case, the warning message sent from the drone detection system to the USSP server contains the parameters timestamp, representing the moment the drone was detected; warning, to identify the type of warning; droneID, when it was possible to identify the drone, this value represent its plate or unique identifier; warning level, it is used to determine the severity of the situation used for instance to determine how to display the warning in the pilot interface; lon, lan, and alt_rel which represent the detected drone position; reason to explain more detail on the warning situation; and token that represent the security key to access the USSP system.


```
{
    "droneId": "123456789ABCDE",
    "timestamp": "2021-04-27T16:48:05+02:00",
    "warning": "drone_detection",
    "warning_level": 1,
    "lon": -7.460771,
    "lat":  43.113822,
    "alt_rel": 50.0,
    "reason": "Violation of NFZ X. Exit the area immediately.",
    "token": "eyJhbGciJIUzI1...AzFUD7SvMmSA"
}
```


Fig. 2. JSON format of the warning message.

To evaluate the performance of the communication protocol, two key parameters are considered: latency and throughput. Latency refers to the time taken from when a client initiates a request to when the client receives the complete response. While throughput measures the number of requests that a server can process at any given moment. Throughput assessments generally aim to identify the maximum throughput, which is the highest number of requests a server can manage simultaneously without experiencing timeouts.

Furthermore, in this research the ratio of throughput is introduced to compare the calculated throughput with the intended throughput for the same time duration. Generally, these two parameters (latency and throughput) are related to each other. As the load approaches maximum throughput, latency will rise as explained in Bermbach and Wittern (2020).

In this study, the API is used to transfer the warning from drone detection system to a USSP system. The algorithm in the form of pseudo code to evaluate the latency and throughput is shown in Figure 3. It starts with importing necessary libraries of Python programming, then define the url and payload of the API. The pseudo code consists of two functions which are RUN_TEST as the main function of the test and SEND_REQUEST as the secondary function to send the API. The main function starts with defining intervals and message sizes used in the experiment, then creating the log files to record the result for throughput and latency calculations. It continues to use ThreadPoolExecutor function to run asynchronous task of sending and receiving messages and then record the starting time as reference for throughput calculation.

The sending of the warning message is conducted by executing the SEND_REQUEST function 100 times in a loop for each case study with the defined intervals. The latency calculation is conducted in this function based on the sending and response timestamp. After each loop, the resulted data is saved into a log file with a filename according to the interval and payload size.

```
Algorithm API Request and Logging Process
Start
Import necessary libraries
Define url and payload
function SEND_REQUEST (index, payload_size, interval)
    Capture sending timestamp
    Send POST request to the url
    Capture response timestamp, status, and response text
    Calculate latency between sending and response
    return (index, sending_timestamp, response_timestamp, latency, status_code, response_text)
end function
function RUN_TESTS()
    Define intervals and payload sizes used in the experiment
    Create log files for throughput and latency data
    Use ThreadPoolExecutor to execute tasks concurrently
    Record start time of the test
    for each request do
        Submit SEND_REQUEST(index, payload_size, interval) for asynchronous execution
        Wait for an interval between sends
    end for
    Record end time of the test
    Calculate throughput as test time divided by number of messages sent
    Update log for latency data
    Update log for throughput data
    Print "Finished testing for defined interval and payload size"
end function
Run RUN_TESTS()
Finish
```

Fig. 3. An algorithm to evaluate the API-based communication performance.

2.3. Experimental Setup

To evaluate the communication system for drone detection, the sender of information is from an office computer using Windows OS running a Python script in the University of South-Eastern Norway in Kongsberg campus and the receiver of information is the DLR U-Space Research Environment (DURE), hosted in Amazon Web Service (AWS) cloud servers located in Frankfurt, Germany, running on an Amazon Linuz server instance. In the experiment, the latency and throughput are calculated for the combination of message intervals (1000ms, 500ms, 100ms, and 10ms), and message payload size by varying the reason text (small (62 bytes), medium (234 bytes), and large (1,844 bytes)). The message intervals and sizes are selected that represent the most possible setting in the drone operations without trying to get the limit of server capability. The system sends 100 messages for each case to avoid the experiment to be considered as Distributed Denial of Service (DDoS) attack. Furthermore, there could be multiple drone detection systems that operated in a U-Space airspace. To simulate this condition, an experiment with two computers sending messages at the same time (twin system) is conducted to evaluate the communication performance. The experiment for single system is conducted twice before and after the experiment of twin system. Thus, the result presented in the next chapter comes from two sets of experiments.

The assumptions considered during the experiment are the drone detection system exists and able to generate the required message immediately, the communication used is the Hypertext Transfer Protocol (HTTP) and always available, the USSP system as the receiver of the message able to send a receipt message immediately, and there is no connectivity problem during the transfer of data.

Python programming language is used to implement the algorithms to evaluate the API protocol for the warning of drone detection because it supports the API protocol and asynchronous process for handling the sending and receiving messages separately. Several modules of Python programming are employed to support this implementation such as REQUESTS to send messages via HTTP using POST method, TIME to apply the interval of sending message, CSV to create log file, DATETIME to record time stamp when sending and receiving message in

order to calculate latency, CONCURRENT to implement asynchronous process between sending and receiving message in parallel, QUEUE to orderly sort the receiving message to their sending index.

2.4. Statistical Data Analysis

Once the experiments were conducted and data were collected, a statistical analysis will be performed to evaluate the results. The statistical parameters such as mean, median, minimum, maximum, standard deviation, and outlier will be calculated and presented in the result chapter. Then, a statistical analysis will be performed to the data using the appropriate method that suitable for the collected data. Analysis of Variance (ANOVA) is a widely used statistical method when comparing the means of multiple groups and testing whether specific independent variables (factors) significantly impact a dependent variable. However, whether ANOVA is suitable for the data depends on several factors and the nature of the data itself such as the dependent variables are continuous, the independent variables (factors) are categorical, the residuals (differences between observed and predicted values) are normally distributed, and the variance of residuals should be roughly equal across groups as explained by Nwobi & Akanno, (2021).

To check those assumptions, a normality test such as The Shapiro-Wilk test is often used to test if the residuals from the model follow a normal distribution and a homogeneity of variances test such as Levene's test can be used to check if the variances across the groups are equal. When the assumptions for ANOVA test are violated, a non-parametric test should be used to test the statistical data such as the Kruskal-Wallis test. It is a non-parametric alternative to the ANOVA assumes that observations in each group come from a population with the same shape of the distribution as described by Nwobi & Akanno, (2021). The statistical analysis will be conducted using Python programming language with SCIPY module which has Kruskal-Wallis, Shapiro-Wilk, and Levene's tests.

3. Result and Analysis

3.1. Latency Analysis

This chapter presents the latency results for two scenarios: a single system sending messages and two systems sending messages in parallel (twin system). Latency statistics for selected intervals and payload sizes are illustrated in Figure 4 with the red diamond indicates the mean value and the black line in the middle of boxplot indicates the median value. The results indicate similar latency values for intervals of 100 ms, 500 ms, and 1000 ms. However, a significant increase in latency is observed at a 10 ms interval, suggesting that this interval is approaching the limit of the USSP system's capability to receive warning messages, as system performance begins to degrade. While the comparison between the single and twin systems reveals a slightly higher latency in the twin system.

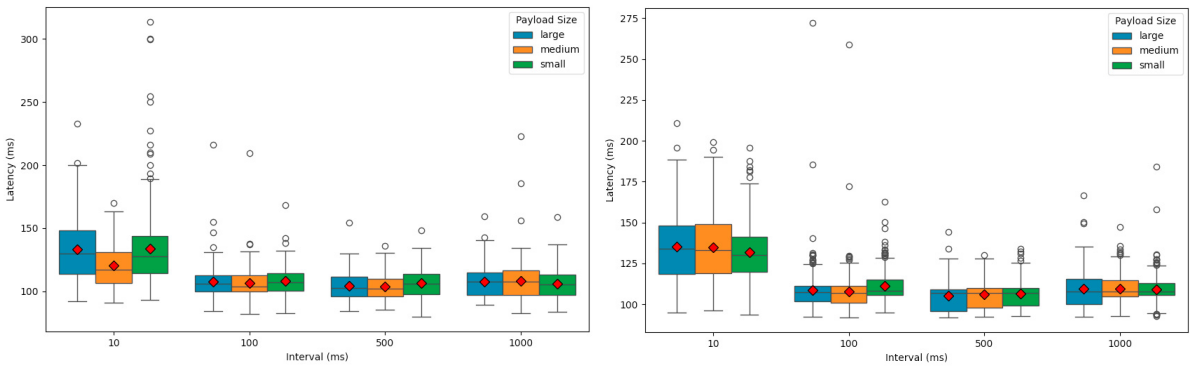
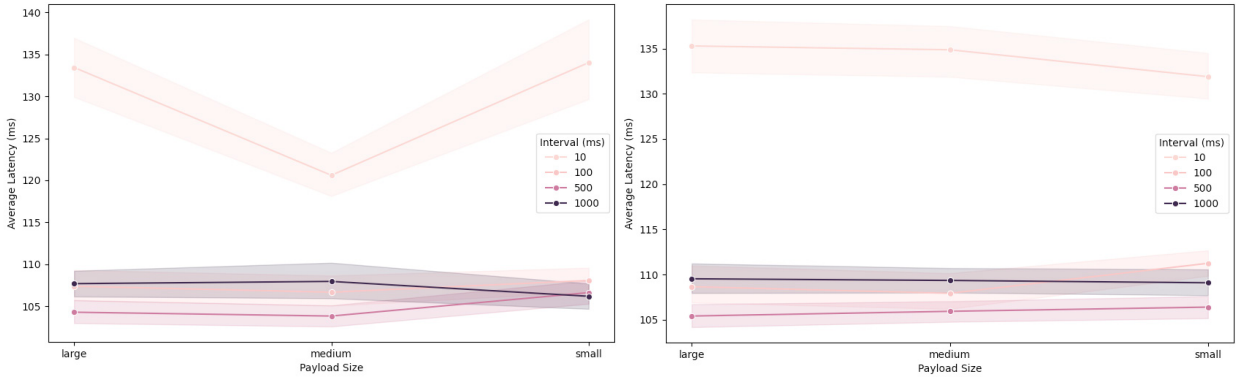


Fig. 4. Statistics of latency result for: (a) single system and (b) twin systems.

Figure 5 displays the mean latency and its standard deviation. The results show no significant effect of message payload size on latency, except at the 10 ms interval. This suggests that the payload sizes used are within the system's handling capacity. The findings further support the observed impact of the small 10 ms interval on system performance. Additionally, Figure 5 shows a minor difference in latency between the single and twin systems, with the twin system exhibiting a slightly higher latency of 2 ms, likely due to the increased number of messages being processed simultaneously.

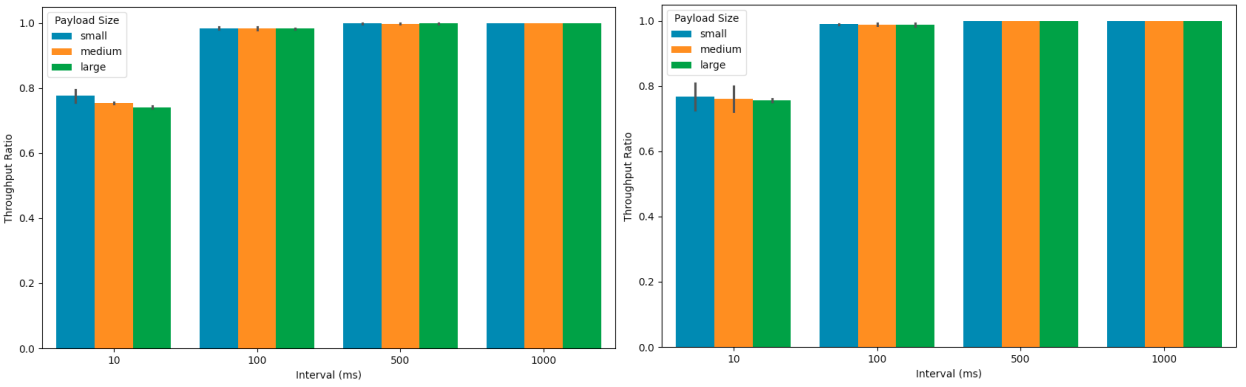


(a) (b)

Fig. 5. Average latency result for: (a) single system and (b) twin system.

3.2. Throughput Analysis

Figure 6 plots the throughput ratio of the API for various intervals and message sizes. The results clearly demonstrate that message interval affects throughput, with the most significant impact occurring at the smallest interval of 10 ms, which drastically reduces throughput compared to other intervals. Payload size also negatively impacts throughput, and the twin system exhibits a slightly lower throughput compared to the single system.



(a) (b)

Fig. 6. Throughput ratio result for: (a) single system and (b) twin system.

3.3. Statistical Analysis

The result for two assumption tests of data: the Shapiro-Wilk test for normality and the Levene's test for homogeneity of variances are shown in Table 1. In both cases for latency and throughput ratio, the p-values of

Shapiro-Wilk test are extremely small (far below a common threshold like 0.05), which means that the residuals for both latency and throughput ratio do not follow a normal distribution. This violates the normality assumption needed for parametric tests like ANOVA. Also, the p-values of the Levene's test are very small, so this indicates that the variances for both latency and throughput ratio are not homogeneous (i.e., the assumption of equal variances is violated). Since both tests indicate violations of the assumptions in normality and homogeneity, using parametric tests like ANOVA is not suitable. Thus, the non-parametric tests such as the Kruskal-Wallis test is a better approach.

Table 1. Data assumptions test result.

Parameter	Shapiro-Wilk Test		Levene's Test	
	Test Statistic	p-value	Test Statistic	p-value
Latency	0.864606	1.135876e-53	249.915972	1.097448e-150
Throughput Ratio	0.696255	1.140684e-08	11.872204	7.968879e-06

The result of Kruskal-Wallis test for both Latency and Throughput Ratio are shown in Table 2. For latency, the p-value (4.35e-280) is incredibly small, indicating that the differences between the groups (based on message interval and payload size) are highly significant. This suggests that at least one group has a statistically different median latency than the others. While for throughput ratio, the p-value (1.45e-09) is also very small, meaning that the throughput ratio significantly differs across at least one group. Overall, there are statistically significant differences in both latency and throughput ratio between the groups.

Table 2. Kruskal-Wallis test result.

Parameter	Kruskal-Wallis Test	
	Test Statistic	p-value
Latency	1293.220000	4.354683e-280
Throughput Ratio	44.081632	1.450109e-09

4. Discussion

Our findings indicate that the latency of the API-based communication for drone detection warnings ranges from 70 ms to 300 ms, as shown in Figure 4. These values are comparable to those reported by Bermbach and Wittern, (2020), who found a latency range of 50 ms to 350 ms in their study of API latency across seven global regions over a three-month period. Also, our result in the latency is in line with the finding from research of Ruseno and Lin (2023) that their latency average of transferring network identification data is slightly higher around 400 ms. The difference could be due to the distance differences of longer location between Czech Republic and Taiwan compared to our systems location between Norway and Germany.

Also, according to the result of statistical tests, the finding indicates that the independent variables (message interval and payload size) have a significant impact on both latency and throughput ratio. This finding is inline with the result from the research of Serrani & Aliverti (2024) about the latency of communication in wireless communication engine. Their p-value of the Kruskal-Wallis test was significant (<0.05) which indicated the interval parameter has a significant contribution to the communication latency.

Additionally, the latency observed in our analysis is significantly lower than the U-Space traffic information distribution requirement, which mandates that latency should be below 5 seconds at least 99% of the time, as specified in Article 11 of the Easy Access Rules for U-Space from EASA, (2024). This suggests that the API-based communication system meets regulatory requirements, making it suitable for supporting drone operations within the U-Space framework.

5. Conclusion and Recommendation

This study aimed to analyze the performance of API-based communication between a drone detection system and a USSP in transferring warning information when drones violate restricted airspace. An experiment was conducted to measure the latency and throughput of the communication system across various intervals and payload sizes. The results indicate that message interval has a significant impact on latency and throughput, particularly at the very small interval of 10 ms, where the system approaches its performance limit. However, the effect of message payload size and multiple systems broadcasting warnings was minimal, attributed to the high-performance capabilities of the USSP as the receiving system. Those findings are supported by the result from the conducted statistical test of non-parametric method.

In conclusion, the API-based communication system can transmit drone detection warnings with sufficiently low latency, meeting the EASA requirements for drone operations. For future research, it is recommended to evaluate the communication system's performance limits while considering additional factors such as bandwidth reduction and the computational hardware utilized. Another aspect to consider in future tests is the quantity and kind of messages being received by the USSP server. Position reports from the drone to the USSP that should be constantly sent during the flight, could be enough once per second or half second. Not very demanding in terms of frequency, however they trigger much more processing than a warning message.

Acknowledgements

The research is part of AI4HyDrop project and supported by the SESAR 3 Joint Undertaking and its founding members and co-funded by the EU's research and innovation programme Horizon Europe under Grant Agreement no 101114805.

References

Barrado, C., Boyero, M., Brucculeri, L., Ferrara, G., Hately, A., Hullah, P., Martin-Marrero, D., Pastor, E., Rushton, A.P., Volkert, A., 2020. U-space concept of operations: A key enabler for opening airspace to emerging low-altitude operations. Aerospace 7. https://doi.org/10.3390/aerospace7030024

Bermbach, D., Wittern, E., 2020. Benchmarking web API quality – Revisited. Journal of Web Engineering 19, 603–646. https://doi.org/10.13052/jwe1540-9589.19563

Cropf, K., Glaneuski, J., Strout, M., Sheridan, P., Hasson, D., & Flynn, C., 2022. Information Services Architecture within Unmanned Aircraft System (UAS) Traffic Management (UTM). IEEE Aerospace Conference Proceedings, 2022-March. https://doi.org/10.1109/AERO53065.2022.9843765

Cruz, A.D., LoCascio, S., Sekhon, J., Suthar, V., Lim, J., Park, Y., 2024. Secure communication in the internet of drones, in: 2024 IEEE International Conference on Consumer Electronics (ICCE), pp. 1–6. doi:10.1109/ICCE59016.2024.10444267.

EASA, 2024. Easy Access Rules for U-space (Regulation (EU) 2021/664), https://www.easa.europa.eu/en/document-library/easy-access-rules/easy-access-rules-u-space-regulation-eu-2021664 (accessed on 14/08/2024)

Jes´us-Azabal, M., Garc´ıa-Alonso, J., Gal´an-Jim´enez, J., 2023. Communication in isolated rural areas: A comprehensive review of the alternatives to the internet, in: Moguel, E., de Pinho, L.G., Fonseca, C. (Eds.), Gerontechnology V, Springer Nature Switzerland, Cham. pp. 11–21.

Kagawa, T., Ono, F., Shan, L., Takizawa, K., Miura, R., Li, H.B., Kojima, F., Kato, S., 2017. A study on latency-guaranteed multi-hop wireless communication system for control of robots and drones, in: 2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC), pp. 417–421. doi:10.1109/WPMC.2017.8301849.

Nwobi, F. N., & Akanno, F. C., 2021. Power comparison of ANOVA and Kruskal–Wallis tests when error assumptions are violated. Metodoloski Zvezki, 18(2), 53–71. https://doi.org/10.51936/LTGT2135

Raffelsberger, C., Muzaffar, R., Bettstetter, C., 2019. A performance evaluation tool for drone communications in 4g cellular networks, in: 2019 16th International Symposium on Wireless Communication Systems (ISWCS), pp. 218–221. doi:10.1109/ISWCS.2019.8877360.

Ruseno, N., & Lin, C.-Y., 2023. Development of UTM Monitoring System Based on Network Remote ID with Inverted Teardrop Detection Algorithm. Unmanned Systems, 1–16. https://doi.org/10.1142/s2301385025500074.

SESAR 3 JU, 2023. AI4HyDrop. Retrieved April 11, 2024, from https://ai4hydrop.eu/

Shivakoti, S., Aurilla, A., Bechina, A., Güldal, S., & Cabañas, E. N., 2021. Drone Operations and Communications in an Urban Environment. The Twelfth International Conference on Sensor Device Technologies and Applications, IARIA.

Zreikat, A.I., Mathew, S., 2024. Performance evaluation and analysis of urban-suburban 5g cellular networks. Computers 13. URL: https://www.mdpi.com/2073-431X/13/4/108, doi:10.3390/computers13040108.