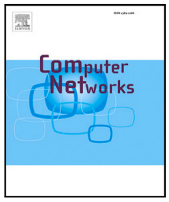
Computer Networks 237 (2023) 110089

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



A quality of experience model for live video in first-person-view drone control in cellular networks

N. González *, M. Solera, F. Ruiz, C. Gijón, M. Toril

Telecommunication Research Institute (TELMA), Universidad de Málaga, Louis Pasteur 35, Málaga, Spain

ARTICLE INFO

Keywords:
Unmanned-aerial-vehicle
First-person-view
Beyond-visual-line-of-sight
Quality-of-experience
Video-quality-assessment

ABSTRACT

Several upcoming 5G and 6G services will rely on unmanned aerial vehicles (UAV) sending live information to remote terminals. Thus, understanding the traffic flows that might influence end-user experience in these services is key for cellular network operators. One of these UAV-based services is first person view (FPV) drone control, consisting on the remote control of the UAV in Beyond Visual Line of Sight scenarios using only the live video visualized in a ground control station. This work focuses on the networking aspects of this service by presenting the assembly, integration and evaluation methodology of an UAV quadrotor teleoperated via FPV through a Long Term Evolution (LTE) network or WiFi radio access link. To assess system performance, three different connectivity schemes between UAV and ground control station are tested, namely server-based connection via LTE, direct LTE, and peer-to-peer WiFi connection. Then, several experiments are carried out in the testbed to characterize telemetry, control and video traffic for FPV service in the above schemes. Later, a methodology is defined to estimate Quality of Experience (QoE) for FPV service based on image quality and video latency measurements collected at network and application level. Results show that the QoE model for live video introduced in this work can be the basis of more sophisticated models for cellular FPV services.

1. Introduction

Unmanned Aerial Vehicles (UAVs) have an outstanding high market potential that is growing promptly. Its ease of deployment, high mobility and autonomous operation provide engaging solutions in a wide variety of applications, such as search and rescue, agriculture, situation awareness and scientific data gathering [1]. Early UAVs were controlled in direct eye contact via short-range communication networks (e.g., WiFi and Bluetooth), which is valid for consumer applications. However, industrial applications, such as public safety or UAV traffic management, require reliable UAV control and wide-area connectivity, which can only be achieved if UAVs are operated Beyond Visual Line of Sight (BVLOS). Cellular networks can provide this connectivity in a cost-effective manner [2,3].

A leading service related to drones is first person view (FPV) drone control (a.k.a. video piloting), where the UAV is driven remotely by a human pilot relying on a live video stream captured on-board [4]. Note that, even if drone control is automated, a person is still needed in the control loop as losing control of the UAV endangers third parties. In this application, user satisfaction depends on stringent performance requirements comprising high data rates, low latency and precise positioning, which can only be achieved by prioritizing UAV traffic [1]. For this purpose, ultra-Reliable and Low-Latency Communications (uRLLC) is considered in Beyond 5th generation (B5G) systems for this mission-critical service [5,6].

As cellular-based UAV services grow in popularity, operators must understand UAV traffic, comprising three traffic flows with very different requirements: (a) command and control (for drone piloting), (b) telemetry data (for monitoring drone state), and (c) payload traffic (i.e., video, sensor data, etc.). The two former traffic flows are often transmitted by Micro Air Vehicle Link (MAVLink) protocol over Transmission Control Protocol (TCP) [7]. In contrast, the protocol used to transmit video traffic depends on the application. For instance, unlike other video streaming services such as YouTube or Netflix, video packets for FPV service are often transmitted by using Real Transport Protocol (RTP), working over User Datagram Protocol (UDP). In congested scenarios, where the achievable throughput per user may be lower than the required video bitrate, UDP may introduce packet losses, producing visual impairments. Conversely, TCP may result in stalling of the video stream due to rebuffering, but does not degrade video quality [8].

In the above context, monitoring user satisfaction as a service performance metric (a.k.a. Quality of Experience, QoE) is key for

* Corresponding author.
E-mail addresses: nuriag@ic.uma.es (N. González), msolera@ic.uma.es (M. Solera), ferv@ic.uma.es (F. Ruiz), cgm@ic.uma.es (C. Gijón), mtoril@ic.uma.es (M. Toril).

https://doi.org/10.1016/j.comnet.2023.110089
Received 15 April 2023; Received in revised form 13 September 2023; Accepted 3 November 2023
Available online 10 November 2023

assigning radio resources, prioritizing service requests or adjusting network settings.

Ideally, the QoE of a service should be assessed with customer surveys, which is expensive and time consuming. Alternatively, QoE models estimate user satisfaction from objective metrics collected in different layers of the protocol stack [9]. Such models can be classified into signal-based and parametric-based [10]. Signal-based QoE models require access to application data to infer user experience (e.g., video content to estimate image quality). In contrast, parametric QoE models rely on network level metrics aggregated per session (i.e., packet loss ratio, latency, bitrate, etc.).

In the literature, several works have proposed simple parametric models for estimating QoE per session in TCP-based offline and live video streaming [11,12]. Likewise, the impact of cellular network latency in live immersive video streaming has been recently evaluated in [13]. However, the above models fail to consider artifacts degrading video quality when video streaming is based on UDP.

This work presents a comprehensive analysis of control, telemetry and video traffic flows for cellular-based FPV service. Based on this analysis, a hybrid parametric and signal-based model for the live video data flow is proposed to estimate the QoE perceived by drone pilots in FPV service. The proposed model jointly considers image quality and video latency. The former is estimated by Video Multimethod Assessment Fusion (VMAF), an advanced full-reference method combining multiple image quality metrics [14]. Model assessment is performed with a real testbed, consisting of a drone connected to a wireless network with LTE/WiFi radio interfaces and driven by a pilot from a Ground Control Station (GCS) has been built. The main contributions of this work are: (a) a discussion of the test system (component selection, assembly, network connection and set-up); (b) an analysis of network performance for different traffic flows in FPV service under different connection settings; (c) generic QoE model for FPV drone piloting, applicable for different radio access technologies; and (d) the assessment of the above model under different UAV-GCS connectivity schemes and radio environments.

This paper is an extended version of a conference paper, [15]. In this version, a more comprehensive review of literature, experimental platform, measurement collection and results are given.

The rest of the paper is structured as follows. Section 2 presents related work. Section 3 describes the experimental platform. Section 4 details the proposed QoE model. Section 5 presents the experiments carried out to test the model. Finally, Section 7 summarizes the main conclusions of this work.

2. Related work

In the literature, several surveys cover generic aspects in the delivery of UAV services in cellular networks. A table summarizing related works that have provided inspiration and support for our work has been included in 1. Regulatory, technical and safety issues raised by standardization bodies to serve aerial users are presented in [16]. Network architecture to support UAV management and operation in 5G is discussed in [17,18]. UAV applications are covered in [19]. It is envisaged there that the integration of UAVs into cellular networks will rely on three scenarios: (a) cellular-connected UAVs, where UAVs become new aerial user equipment coexisting with terrestrial users and accessing the cellular network infrastructure [21], (b) UAV-assisted (a.k.a. UAV-based or UAV-enabled) cellular communications, where UAVs become flying base stations that can be smartly relocated to improve coverage, spectral efficiency and capacity of the existing terrestrial wireless communication system [23], and (c) UAV-UAV communications, where several UAVs are connected directly to each other to facilitate autonomous flight behaviors, cooperation in a UAV swarm, collision avoidance or ad-hoc networking (a.k.a. flying ad-hoc network) [24]. This section focuses on cellular-connected UAVs (i.e., aerial users).

The challenges of serving aerial users in cellular networks from the radio perspective have been extensively covered [20–22]. First works focus on propagation issues, updating legacy propagation models for ground base stations with 3D models for aerial base stations [25]. Later works evaluate the impact of UAV altitude on coverage and throughput metrics [26–28]. The interference caused by UAVs operating above the normal height of ground UEs is identified as a critical issue [24]. Throughput degradation experienced by ground users in the presence of UAVs is evaluated in [29]. Nonetheless, field tests in [32] shows that current LTE networks can provide continuous coverage to UAVs, allowing for uninterrupted telemetry and video streaming, despite being optimized for ground user, and without significant impact on other users.

One of the most common applications in cellular-connected UAVs is real-time video streaming. For conventional (i.e., non-real time) videostreaming, a regression analysis can be performed to estimate the QoE from service performance indicators at the application layer, such as initial reproduction delay, stalling frequency and stalling duration [33]. Alternatively, a QoE can be estimated from the client buffer level [34]. The impact of different transport protocols on the QoE of legacy videostreaming schemes is evaluated in [8]. With application-level measurements, the relationship between network-level impairments and QoE is derived for User Datagram Protocol (UDP) and Transmission Control Protocol (TCP). With the advent of adaptive video streaming, current approaches add image quality metrics (e.g., average/standard deviation of image quality or frequency of switches between quality levels) to estimate QoE [36–38]. More recently, a parametric bitstream-based QoE model for long adaptive streaming session (a.k.a. ITU-T P.1203.1) is proposed in [39]. QoE estimates for video streaming can be improved by combining classical QoE models with machine learning models [40]. For cellular networks, a parametric QoE model for TCP-based live videostreaming based on network-level metrics is presented in [12]. Likewise, an implementation of the ITU-T P.1203 for live video streaming over LTE is tested for two different video transport protocols [41]. The impact of the uplink on latency in TCP-based 360 live video streaming in LTE is analyzed in [13]. Complementarily, several works propose novel schemes to improve videostreaming with cellular-connected UAVs by updating different layers of the protocol stack. For the application layer, video adaptation for UAV-based surveillance is studied in [42,43]. For the network layer, an experimental study of live multicast video streaming from multiple drones is carried out in [44]. For the network management plane, in [45], a self-tuning scheme improves the QoE of live videostreaming with multiple drones and an aerial base station by adjusting UAV location and transmission power with deep reinforcement learning. It should be pointed that most of the QoE models used in the above works are conceived for video delivery over TCP.

UAV teleoperation relies on the above-mentioned real-time video streaming. A first group of works on teleoperation checks pilot requirements. For instance, [46] defines user needs for teleoperating connected and automated vehicles in cellular networks. To address such needs, operator workload is reduced with advanced human–machine interfaces (e.g., haptics, immersive, virtual augmented reality, . . .) [47] and functionalities (e.g., obstacle detection [48]). Other works focus on cellular network performance for vehicle teleoperation. In [49], the feasibility of teleoperated car driving based on LTE networks is shown in a testbed. The requirements for 5G UAV use cases are presented in [50]. Likewise, the potential and limitations of new 5G features, such as massive antennas or mmWaves, for UAV communications are covered in [51]. For teleoperated UAVs, a system-level simulator is used in [30] to evaluate video quality and latency in real time video streaming and control for FPV in LTE. In [52], flight tests with different UAV altitudes are carried out to compare latency in different radio technologies. In [35], an FPV drone flight simulator in a cloud-gaming platform is used to check the influence of video settings (image resolution, frame rates and bitrate) on perceived video quality under ideal

Table 1
Related work and contributions.

References	Summary	Contribution
[16–22]	Surveys and general aspects of UAVs cellular communications.	–
[21,23,24]	UAV connection topologies: UAV as end-user, UAV as base station and UAV-to-UAV connection	UAV as end user, design, assembly and test
[24–31]	Issues description and/or simulations to check the impact of propagation, interference, altitude on coverage, throughput and latency	Identify main issues related to FPV drone control and introduce a QoE model based in them
[32]	Real coverage tests	Real QoE tests
[8,33–35]	QoE UAV videostreaming	A new QoE model specifically for FPV drone controlled by videostream
[12,36–45]	Videostreaming quality	A key component of the proposed QoE model is videostreaming quality
[13]	Latency videostreaming	A key component of the proposed QoE model is video latency
[46–49]	Demand for QoE models in remotely controlled activities	Due to this demand, a new QoE model is proposed
[50,50,51]	Potential and limitations of 5G UAV cases	Forecasts and guidelines for future testing with 5G
[52–54]	Latency and/or throughput flight tests	Latency and throughput fulfillment tests

network conditions (low video and control latency, no packet loss) with real users.

Most of the above works evaluating QoE of cellular teleoperated UAVs rely on analytical or simulation models. Very few works conduct measurement campaigns in real testbeds. In [52], flight tests with different UAV altitudes are carried out to compare signal-level, latency and jitter in different radio technologies. Such signal-level measurements can then be used to estimate latency via simulation for command-and-control traffic when serving multiple drones in a commercial LTE network [31]. In [53], a prototype of a LTE-based UAV communications system is used to measure latency, handover and signal strength. Likewise, in [54], an indoor testbed for cellular-connected UAV is used to evaluate throughput, end-to-end delay and reliability of command-and-control and real-time video streaming.

However, to the authors' knowledge, none of these works has proposed an analytical QoE model for FPV drone control in a BV-LOS scenario based on image quality and video latency measurements collected in a real testbed.

3. Experimental platform

This section details the experimental platform used in this work, illustrated in 1 . The platform consists of a UAV connected to a GCS through a wireless system. For clarity, UAV components are first explained. Then, the configuration of the wireless radio link is described. Next, the different options to connect UAV and GCS at the application layer are introduced. Finally, protocols in UAV-GCS interface are outlined.

3.1. UAV components

Fig. 2 shows the quadrotor used as UAV. Components have been chosen to minimize payload weight to save battery. The structure consists of a frame, propellers, power module, motors and Electronic Speed Controllers (ESCs). As usual, the four rotors are placed at the edges in X-shaped form. To prevent the helicopter from tilting in relation to its axis of orientation, two propellers rotate in one direction and the other two in the opposite direction. Motors are controlled (i.e., synchronized and balanced) by ESCs, which translate data from the flight controller, modulated with Pulse Width Modulation (PWM), into motor control actions. The power module, also controlled by ESCs, supplies power from battery to motors. A LiPo battery of 4000 mAh and 14.8 V is selected. A Flight Controller (FC) is also included, where two control algorithms are implemented [1]. A first controller adjusts the attitude to obtain a desired position and speed. In an autonomous flight, these actions are driven by sensors, whereas in manual flights such information is provided by the remote pilot. Then, a second controller computes the required motor speed to achieve the targeted attitude and send it to the motor controllers. The selected FC is Pixhawk 2.1 Standard Set. This set contains the Cube Orange and the standard Automatic Dependent Surveillance Broadcast (ADS-B) carrier board. The Cube Orange has multiple sensors known as Inertial Measurement Units (IMUs), reporting linear acceleration and rotational rate. There are three IMUs for the accelerometer and three sensors for the gyroscope (i.e., a sensor per pitch, roll and yaw axes). Likewise, the ADS-B broadcasts UAV position, attitude and speed to enable tracking from the GCS by using the ArduPilot software [55]. A Global Navigation Satellite System (GNSS) receiver is also included to provide precise information about the UAV position. Finally, C920 Logitech camera is installed to offer live video streaming for FPV control.

3.2. Radio connection

The UAV is connected to a LTE network by using the solution provided by XBStation [56]. As shown in Fig. 1, three main components can be distinguished for *LTE server-based* connection: XBCloud, XBLink and XBStationPilotApp.

XBCloud is a management and real-time monitoring center.

XBLink transmits telemetry, command/control and video between UAV and GCS passing through XBCloud. It is structured in three parts:

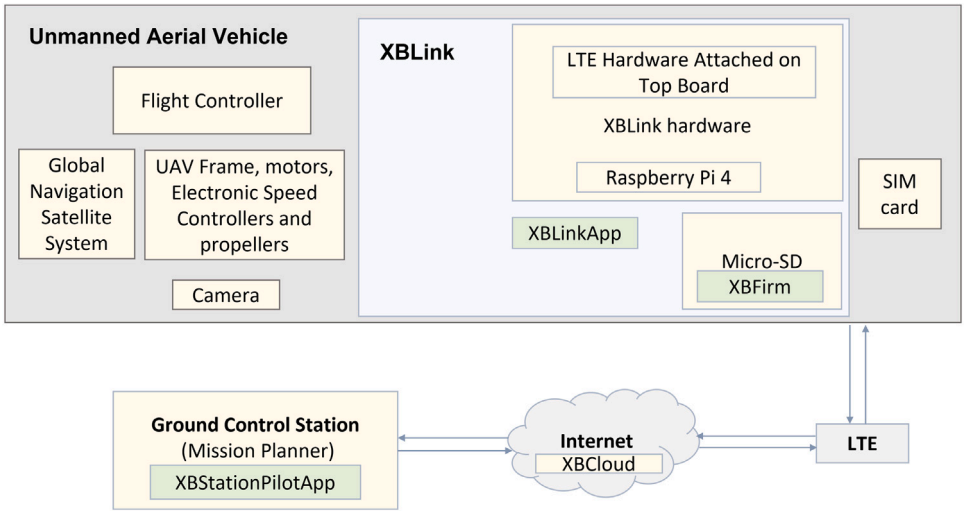


Fig. 1. LTE server-based platform. The yellow boxes represent hardware components and the green boxes represent software components.

Fig. 2. Unmanned aerial vehicle built for this work.

XBLink hardware, XBFirm and XBLinkApp. XBLink hardware, illustrated in Fig. 1, consists of a Raspberry Pi 4 and a Hardware Attached on Top (HAT) board providing connection to the LTE system and the FC [57]. XBFirm is a Linux-based software installed in a Micro-SD card inserted in the Raspberry Pi with the code used to enable connection and data transmission. XBLinkApp is a WiFi hotspot for controlling XBLink remotely for configuration and monitoring tasks. Finally, XBStationPilotApp is a software installed in the control station to interchange telemetry and video data between XBLink and an analog software in the GCS.

3.3. UAV-GCS connection

The GCS is a laptop equipped with Mission Planner software [58]. Such a software allows (a) the configuration and tuning of UAV parameters, and (b) saving and loading missions for autonomous flights. For the latter purpose, it stores monitoring and surveillance data in telemetry log (*.tlog*) files that can be downloaded offline. These files are essential for the traffic and network performance analysis carried out in this work.

The connection between UAV and GCS is defined at the application layer. Fig. 3 shows the three different UAV-GCS connection schemes tested here. In the first scheme, referred to as *LTE server-based*, the UAV accesses the Internet through a private pilot LTE network. Information exchanged between UAV and GCS goes through the XBStation software installed in the Raspberry Pi. In this scenario, telemetry and control traffic is carried over TCP, whereas video traffic is transported over UDP. Fig. 1 shows the first configuration, referred to as *LTE server-based*. In the second scheme, denoted as *LTE direct*, UAV and GCS are also connected by LTE, but information does not pass through the XBCloud server, but it is directly transmitted to the GCS. To avoid the need for a server, MAVProxy software Linux package is used in the Raspberry Pi to send telemetry messages directly from UAV to GCS. The difference between *LTE server-based* and *LTE direct* in the way the data is sent and received. Therefore, the expected parameter difference between the two configurations is the end-to-end delay. In the server-based configuration, the data is transmitted and passes through a proprietary server and arrives at the GCS. Due to the distance to the server, end-to-end latency increases. In the direct configuration, a pipeline is created where data is sent from the transmitter to the receiver without the need for an intermediate server to process and retransmit the data. Likewise, ffmpeg commands [59] are used to create a UDP pipeline to transmit video encoded with a H.264 codec from UAV to GCS. Finally, in a third scheme, referred to as *WiFi direct*, UAV and GCS are connected with a peer-to-peer link via WiFi, using the same mechanisms as in LTE direct scheme.

3.4. Application-level protocols

The FC is based in the popular open-source ArduPilot software. At the root of ArduPilot, Micro Air Vehicle Link protocol (MAVLink protocol) is used at the application layer, encapsulated over TCP [7]. MAVLink is a protocol for telemetry data that allows transmission and reception of short messages over any serial connection regardless of the underlying transport technology. MAVLink messages can be classified into three types:

- Presence messages: once the communication is opened, both UAV and GCS send one heartbeat message per second.
- Status message: sent from the UAV to the GCS to provide system status information, such as location, speed and attitude.
- Command messages: sent from the GCS to the UAV to execute some actions or missions, e.g. take-off, landing at some specified point or following a predefined path.

These messages are recorded in *.tlog* files created by Mission Planner, where all their fields can be visualized to identify message type.

Likewise, video packets are transmitted using RTP protocol [60], widely used for real-time transmission of video and audio. RTP is typically encapsulated in UDP to minimize delays. Every instantiation of RTP requires specifying a profile defining the codecs used to encode the payload data. Then, RTP messages include in their header a *payload type* filed with payload format, and sequence number and timestamp, to compute packet losses and delay.

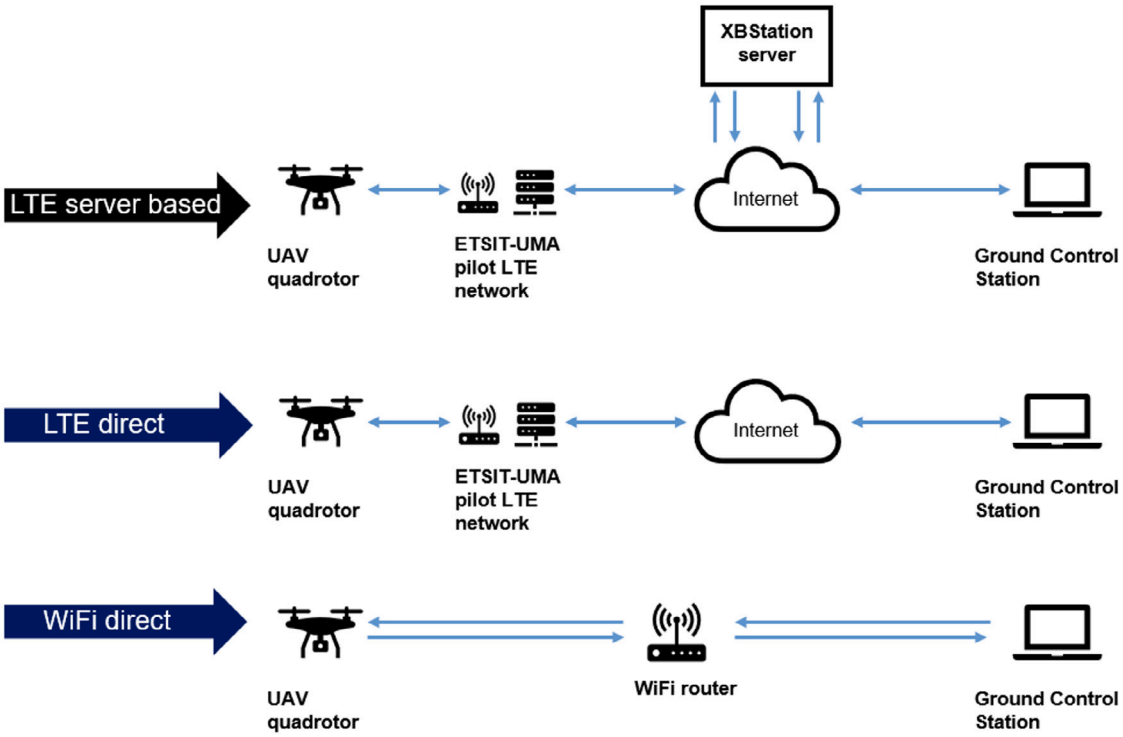


Fig. 3. UAV-GCS connection schemes.

4. QoE model for FPV service

This section presents an analytical model to measure QoE perceived by users demanding FPV services. For clarity, the rationale of the model is first introduced by revising service requirements and video quality assessment methodologies.

4.1. FPV service requirements

In this work, UAV traffic consists of telemetry, command/control and video traffic. The former two data flows are needed in autonomous flying, when the UAV follows a pre-scheduled flight plan, but a connection between UAV and GCS is maintained for monitoring and route updates when necessary. The latter flow is only needed in FPV control, when the UAV is driven by human person.

For long-range missions with no direct view to the UAV, a low-latency and highly reliable bidirectional connection between UAV and GCS must be established. Only thus can all telemetry of the drone be available in near real time with high level of accuracy (i.e., time/space resolution), while commands sent by the GCS are received by the UAV in due time. Additionally, in FPV service, a low-latency and high-throughput connection is needed to transport live video content displayed by the pilot.

Table 2 presents connection performance requirements defined by 3GPP for the UAV use case [61]. These are 4/9/30 Mbps for 720p/1080p/4K and 100 ms end-to-end latency for live video data rate, 1 kbps and 360 ms end-to-end latency for service control data rate, and 12 kbps and 1 s of end-to-end latency for telemetry. Latency values are end-to-end, including both uplink and downlink when the GCS is connected via radio. From these values, it is clear that live video is the most demanding data flow. Its performance cannot be guaranteed in the presence of signal quality and capacity issues, often common in radio links of a cellular networks. These problems justify the need for assessing pilot experience for users demanding FPV video services.

Table 2
Connection performance requirements defined by 3GPP for the UAV use case [61].

Traffic Type for C2	Bandwidth	Latency
Command and Control	0.001 Mbps	VLOS: 10 ms BVLOS: 360 ms
Telemetry	0.012 Mbps w/o video	1 s
Video Streaming	4 Mbps for 720p video 9 Mbps for 1080p video [30 Mbps for 4K Video]: optional	100 ms
Situation Aware Report	1 Mbps	10–100 ms

4.2. Objective video quality assessment

Objective video quality can be evaluated by media-based, bitstream-based or parametric methodologies [10]. Media-based methods decode the video content, whereas bitstream-based methods rely on the video elementary stream. Both require accessing the application layer, so they are suitable for service providers with access to one side of the link. As an alternative for network operators, parametric packet-level methods analyze protocol messages to identify the different stages of the session. From such information, service key performance indicators (e.g., stalling metrics) are obtained, which are then mapped to Mean Opinion Score (MOS) values through formulas derived in subjective tests. An example of parametric scheme is the U-vMOS model, developed by Huawei, providing a MOS value for the QoE experienced by video-on-demand users from initial reproduction delay and stalling statistics [62]. Unfortunately, if video is encrypted, service performance can only be roughly estimated from high-level network performance metrics (e.g., average session throughput). Moreover, the resulting network-level models are only valid for the transport protocol of they were conceived (in most cases, TCP).

Legacy parametric QoE models rely on simple regression formulas to compute MOS values from a few objective metrics selected a priori. With the latest advances in big data analytics, more complex

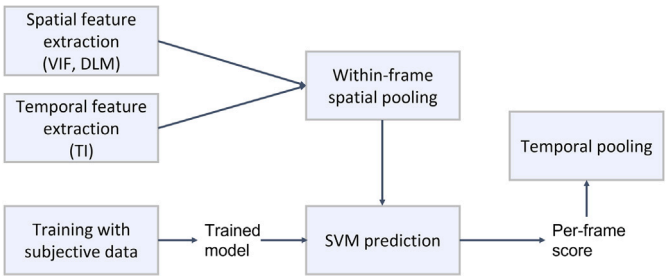


Fig. 4. VMAF video quality assessment methodology [14].

algorithms are used to isolate the indicators that better reflect user experience [63]. An example is VMAF methodology [64] developed by Netflix, whose flow diagram is illustrated in Fig. 4.

VMAF is an objective full-reference video quality assessment method that estimates QoE from the comparison of the original and degraded video sequences. For this purpose, it uses a support vector machine regressor to combine different quality metrics considering spatial and temporal features, namely Visual Information Fidelity (VIF), Detail Loss Measure (DLM), and Temporal Information (TI). VIF is an image quality index based on natural scene statistics, which measures loss of fidelity in four scales based on characteristics of the human visual system [65]. DLM quantifies the visual impairment due to loss of details that affects content visibility, while considering redundant impairment distracting viewer attention [66]. Finally, TI reflects time properties by measuring the average of the difference between consecutive frames [64]. VMAF output is a score value in the range of 0–100 per video frame (the higher value, the better video quality). By combining multiple metrics, VMAF output correlates better with subjective measures than legacy full-reference methods, such as PSNR and SSIM [67–69].

4.3. QoE model

In FPV service, video content is transmitted in real time for drone teleoperation through a unreliable connection that may experience packet losses and delays. The absence of retransmission mechanisms in UDP results in image artifacts and video freezes. Moreover, the use of gateway servers may lead to a high video latency. All these issues must be considered when evaluating the QoE of live video in FPV drones.

Fig. 5 shows the structure of the proposed model to estimate the satisfaction of FPV pilot users. In the model, the QoE experienced of a user demanding FPV service, QoE_{FPV}, is computed from image quality and video latency as

$$QoE_{FPV} = 1 + \frac{(QoE_{ImgQual} - 1)(QoE_{VidLat} - 1)}{4}, \quad (1)$$

where $QoE_{ImgQual}$ and QoE_{VidLat} are MOS values for image quality and delay of video frames between transmitter and receiver, respectively.

The former indicator is defined as

$$QoE_{ImgQual} = 1 + \frac{(maxQoE_{ImgQual} - 1)(QoE_{VMAF} - 1)}{4}, \quad (2)$$

where $maxQoE_{ImgQual}$ denotes the maximum MOS for image quality, limited by the quality of the factors in transmission (codec, resolution, bit rate, frame rate and display rate) as defined in [62] and showed in Table 3. On the other hand, QoE_{VMAF} is a MOS value reflecting the objective video quality of the video session estimated with the above-described VMAF methodology. Thus, unlike U-vMOS, the proposed QoE model considers the impact of image artifacts due to packet losses on image quality.

The second indicator, QoE_{VidLat}, is a MOS value reflecting the loss of interaction caused by video delay between transmit and receive side,

Table 3
Maximum MOS value due to image quality depending on video resolution and screen size [62].

Resolution	Screen size		
	4.5-inch	7-inch	9.7-inch
4K	4.90	4.86	4.82
1080p	4.62	4.52	4.44
720p	4.32	4.17	4.05
360p	3.49	3.25	3.06

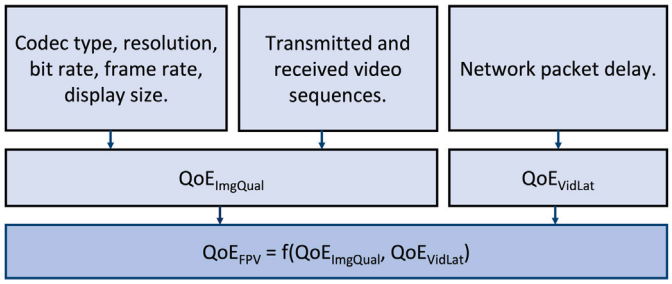


Fig. 5. QoE model for first person view service.

computed as

$$QoE_{VidLat} = \begin{cases} 5, & AvgFrameLat < 0.1 \\ 1, & AvgFrameLat > 0.5 \\ -10 \cdot AvgFrameLat + 6, & \text{otherwise} \end{cases} \quad (3)$$

where $AvgFrameLat$ is the average video frame delay (in seconds), defined as the time it takes to receive a complete video frame at the receiver side from the GCS during the session. In (3), it is assumed that video latency above 500 ms are unacceptable for FPV service, while video latency below 100 ms are excellent for FPV service [61].

In this work, $AvgFrameLat$ is estimated from network measurements as the product of the average number of packets per frame and average packet-level network delay as follows

$$AvgFrameLat = AvgNoPacketsPerFrame \cdot AvgPacket\ Delay \quad (4)$$

AvgNoPacketsPerFrame can easily computed by dividing the total number of packets by the number of frames in the FPV session.

5. Performance assessment

This section describes experiments carried out to analyze the traffic generated by FPV service and assess the QoE model presented in Section 4 over the testbed described in Section 3. For clarity, the measurement collection process is first described, experiments are then explained and results are finally analyzed.

5.1. Measurement collection

Measurements collected during experiments include network-level statistics gathered by network probes and application-level measurements captured on the extremes of the communication. The former are used to characterize traffic generated by FPV service, while the latter allow to check service performance.

5.1.1. Network-level measurements

Packet-level statistics are collected in both UAV and GCS for traffic analysis, including UDP/IP metrics for video traffic and TCP/IP metrics for telemetry and control traffic. For this purpose, *Tcpdump*, a Linux-based package installed in the Raspberry Pi, is used in the UAV. In the GCS, information is gathered in packet capture (PCAP) files generated by Wireshark. The following statistics are considered:

- Packet size [bytes].
- Inter-arrival packet time [s], defined as the time between received packets.
- Packet rate [packets/s], defined as the number of packets received per second.
- Bitrate [bits/s], defined as the data rate at the receiving side.
- Packet delay [ms], defined as the time since a given packet is transmitted by the source, measured at the communication interface, until it is successfully received at the destination.
- Dropped packet ratio [%], defined as the percentage of packets lost in the communication.

The average packet rate is used as a proxy of quality improvement, because a higher packet rate usually indicates that the image is transmitted with more detail or higher resolution. In fact, it is a parameter considered for the value of max $maxQoE_{ImgQual}$ in the proposed QoE model. In contrast, when a sudden increase of instantaneous packet rate is detected occasionally, it may be an indication of repetition of lost packets. On the other hand, the visualization of packet inter-arrival times eases the detection of received packets and packet size. For instance, small inter-packet times might indicate that packets are small. The loss of video packets over UDP causes visual artifacts. Thus, packet loss ratio at network layer is a proxy of video quality.

5.1.2. Application-level measurements

Application data is also registered to compute key service performance indicators impacting user experience VMAF methodology is applied to assess image quality. For this purpose, both the video sent by the UAV and that received by the GCS are stored. The video transmitted by the UAV cannot be gathered while the camera port is transmitting. To circumvent this issue, virtual video devices are created in the Raspberry Pi to capture the video content via Linux commands. Two virtual video devices are created: (a) */dev/video2*, used to store the transmitted video, and (b) */dev/video3*, used to transmit the video to the GCS via *ffmpeg* commands. During video transmission, the port of the camera in the drone, */dev/video0*, is used to copy video frames captured to */dev/video2* and */dev/video3* virtual devices. Likewise, the video received in the GCS is gathered in an *.avi* file by Mission Planner.

Once video is captured at both sides of the communication, VMAF score is calculated as described in Section 4. Note that VMAF requires that both the original and degraded videos: (a) have the same encoding properties (e.g., image resolution, frame rate...), which is checked by *ffprobe* command, and (b) have the same duration and are time aligned at frame level. To synchronize both videos, the transmitted video is used as reference, so that the received video is locked to the transmitted video by adding a fixed delay [70]. Synchronization is performed by selecting the delay maximizing the Peak Signal-to-Noise Ratio (PSNR) from the comparison of frames in the original and time-adjusted degraded sequence. Then, a per-frame and average VMAF score is obtained by *ffmpeg* commands. A basic command to calculate it is: *ffmpeg -i distorted.mp4 -i original.mp4 -filter_complex libvmaf -f null* – . The resulting VMAF score, ranging from 0 to 100, is then mapped to 1–5 MOS values.

The validity of the proposed model is justified by the soundness of its constituent elements. The combination of MOS values related to video quality and interaction delay is already used in U-vMOS model in [62]. VMAF method is a well-established methodology for quantifying the impact of video artifacts on viewers [71–73]. Likewise, latency thresholds are taken from 3GPP 22.825 specification [74]. Finally, a linear relationship between video frame latency and user opinion is assumed for simplicity to reflect gradual degradation of service usability with increasing latency reported by professional drone pilots.

5.2. Experiment description

The radio access network used to connect UAV and GCS is the private LTE network described in [13], comprising picocells working at 2.6 GHz connected to a compact network core. Picocells are Huawei's BTS3911B model, designed for indoor environments. The frequency band tested in our experiments is 2.6 GHz. Three experiments are performed:

5.2.1. Experiment 1: Comparison of connectivity solutions

This experiment aims to characterize traffic patterns in video, control and telemetry data flows between UAV and GCS by analyzing packet level traces with different networking schemes. The goal is to present a methodology to estimate the quality of experience of an FPV service. For this purpose, in experiment 1, it is important to characterize the different traffic flows at packet level to have a deeper understanding of network configuration. Since the existing traffic flows were not initially identified, their characterization is part of the work presented in this article. Furthermore, this first experiment serves as a starting point to highlight the type of packets transmitted for telemetry, video and control, and justify the need for tools to process them in the proposed QoE model. To ease comparison, system configuration emulates an FPV use case in a controlled scenario, where the UAV is placed in a fixed static position close to a LTE pico-cell. Note that image motion greatly affects video coding efficiency and packet flow. To ensure that video codecs work in realistic conditions, a laptop playing a landscape video sequence from a real aerial camera captured with 4K quality (i.e., the highest offered by YouTube) is placed in front of the UAV's camera, which retransmits it H.264 encoded at 30 fps and 1080 × 720 resolution. Video, control and telemetry data are carried through the three different network configurations of UAV-GCS link described in Section 3 (LTE server-based, LTE direct and WiFi direct). For simplicity, these configurations will be hereafter denoted as $LTEserver$, $LTEdirect$, $WiFidirect$, respectively.

For each configuration set-up, two video sessions of 2 min are analyzed. In the first session, the video encoder has the default configuration, with a Group of Pictures (GOP) combining I-type (intra) frames, P-type (predicted) and B-type (bidirectional) frames. In the second session, the encoding process is changed to reduce the video processing delay by configuring *ffmpeg* zero-latency tune option, where differential coding is disabled and only I frames are used. Hereafter, these configurations of the encoder are referred to as normal and low-latency modes, respectively. Unless stated otherwise, experiments are carried out with the default configuration (normal). In all tests, packet size, data rate and network delay are captured in both UAV and GCS.

5.2.2. Experiment 2: QoE assessment for static UAV demanding FPV

The aim of this experiment is to illustrate the behavior of the proposed QoE model by evaluating how FPV video is affected by different types of link degradation, namely packet loss and packet delay. Experiment 2 is a first contact that emulates drone flight in controlled laboratory conditions. In this experiment, the proposed QoE model is applied to estimate the quality that the user would perceive in extreme radio network conditions. To aid analysis, this experiment is carried out by introducing controlled impairments with a network emulator in a lab environment with optimal radio conditions. Thus, the number of uncontrollable variables is reduced. No human pilot is needed, since UAV is static on the ground. Such an analysis is carried out by introducing controlled impairments in a lab environment with optimal radio conditions (i.e., UAV in a static position on the ground, absence of control commands, good signal quality from picocell, negligible video motion and low video resolution requiring few radio resources). Again, UAV transmits video with H.264 encoding at 30 fps, with 1080 × 720 resolution. Note that, since the aim here is to assess the impact of radio link issues on QoE_{FPV}, it is desirable to experience a high QoE_{FPV} in the ideal (i.e., without impairments)

scenario. Impairments are introduced by NetEm emulator [75] located at the output UAV interface. Selected packet loss ratios are 0.01, 0.025, 0.1, 0.2, 0.3, 0.5 or 1.0%. Likewise, tested packet delays are 25 ms, 50 ms, 100 ms, 200 ms, 300 ms or 500 ms. The emulated network conditions do not combine different types of impairments. Even if both issues compromise drone control, the underlying mechanisms are different. Since transport protocol for video is UDP, packet losses introduce image artifacts degrading image quality. In contrast, latency issues do not affect image quality, but increase pilot reaction time. For each emulator setting, 3 tests of 3 min are run, for a total of 39 tests.

5.2.3. Experiment 3: QoE assessment for flying UAVs demanding FPV

The aim of this experiment is to assess the QoE of users demanding FPV service with a moving UAV in a real scenario. In experiment 3, real flight tests are performed with a pilot controlling the drone based on the displayed video. The aim is to assess the QoE of users demanding FPV service with a moving UAV in a real scenario. As in experiment 3, network degradations are also introduced with the network emulator to obtain a wider range of impairments in a controlled manner. However, unlike experiment 2, this experiment is influenced by external environmental conditions and human factors (e.g., weather conditions and pilot expertise, affecting the amount of movement in the video sequence captured by the camera). To this end, the UAV flies in an outdoor BVLOS scenario. Link degradations are generated by a combination of radio channel conditions and network emulator. The former ensure more realistic performance fluctuations (due to, e.g., fast signal fading), while the latter is used for a wider range of impairments in a controlled manner. A pilot remotely controls the UAV helped by the video displayed in the GCS. To isolate the impact of the selected effects on QoE, tests are performed on days with weather conditions within state air safety regulations and having as little influence as possible in radio link conditions.

For readability, traffic transmitted from UAV to GCS and vice-versa are referred to as upstream and downstream traffic, respectively. Thus, telemetry and video data flows are upstream traffic, whereas control traffic is downstream traffic. Note that experiment 1 covers the three traffic components (telemetry, video and control), whereas experiments 2 and 3 focus on the video data flow (i.e., the one with the highest impact on QoE for FPV service) for an LTE connection.

6. Results

This subsection details results obtained in the assessment process. For clarity, results from each experiment are presented separately.

6.1. Experiment 1

Fig. 6 depicts the Cumulative Distribution Function (CDF) of bitrates for telemetry (orange lines), control (blue lines) and video (green lines) flows in the *LTE server* scheme with the normal video encoding configuration. Dashed curves correspond to measurements the link between UAV and XBStation server (from .pcap files), whereas solid curves correspond to measurements captured in the link between the server and the PC with the GCS. Curves for video traffic in the server-GCS interface, not shown for a better visualization, are very similar to those from the UAV-server interface.

In Fig. 6, it can be observed that the drone transmits video at approximately 1 Mbps. Likewise, control traffic consumes 200 bps in both upstream and downstream. Such a rate would be higher when a joystick is connected to the GCS, since joystick commands would be sent via GCS by MAVLink commands. Finally, data rate for telemetry on UAV-server link reaches 300 kbps, whereas the same traffic flow has a bitrate of only 10 kbps on server-GCS link. To understand this unexpected phenomenon, Fig. 7 shows curves similar to Fig. 6 but for packet size. The orange dashed curves reveals that, in the UAV, 70% of telemetry packets are filled with padding until they have the

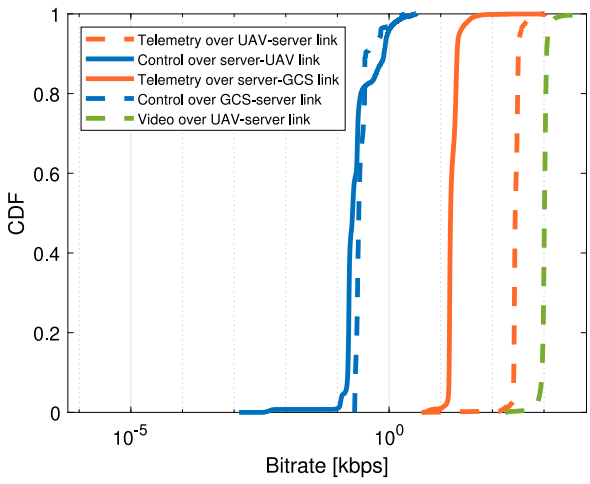


Fig. 6. CDF of bitrate for telemetry, control and video traffic when drone is connected to the ground control station through a LTE network and a cloud server (*LTEserver* scheme).

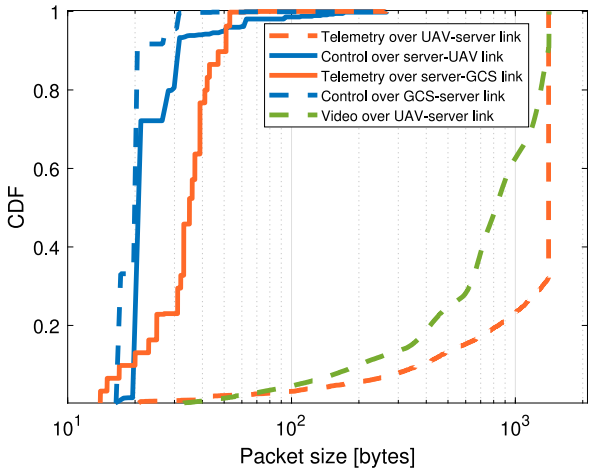


Fig. 7. CDF of network packet size for telemetry, control and video traffic when drone is connected to the ground control station through a cloud server (*LTEserver* scheme).

maximum size allowed by IP (a.k.a. Maximum Transfer Unit, MTU). Consequently, telemetry packets are the heaviest ones over the UAV-server (i.e., 1000 bytes on average). In the server, these packets are repacked and padding removed, drastically reducing packet size (30–40 bytes) and bitrate in the server-GCS link. Regarding control data flow, packet size CDFs are roughly similar in the two links. Specifically, 70% of packets comprise consist of 20 bytes, which is the size of the *HEARTBEAT* message sent by the GCS to the UAV every second to update connection status. Finally, video packets have a varying size ranging from approximately 100 to 1000 bytes. Specifically, 70% of packets consist of 20 bytes, which is the size of the *HEARTBEAT* message sent by the GCS to the UAV every second to update connection status. Finally, video packets have a varying size ranging from approximately 100 to 1000 bytes.

Likewise, Fig. 8 illustrates the CDFs of inter-arrival packet time. As expected, packet rate is higher for video traffic than for telemetry and control traffic, with a packet inter-arrival time of 10 ms for most video packets, 30 ms for most telemetry packets and 1 s for most control packets. More important, curves of both UAV-server and server-GCS links are similar for each data flow. This behavior confirms that the same telemetry packets are sent over the two interfaces, being the above-mentioned padding the only cause for the different telemetry bitrates

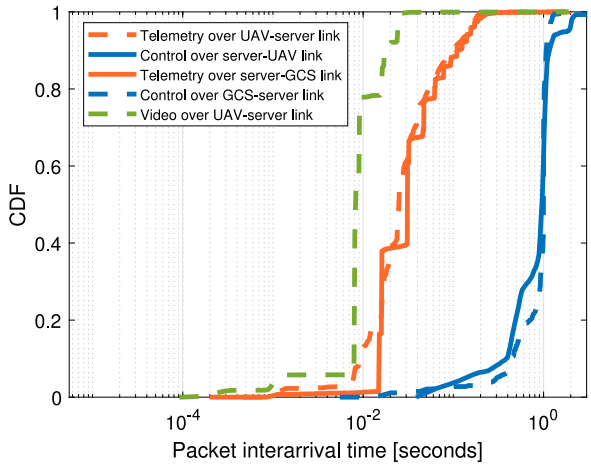


Fig. 8. CDF of packet inter-arrival time for telemetry, control and video traffic when drone is connected to the ground control station through a cloud server (*LTEserver* scheme).

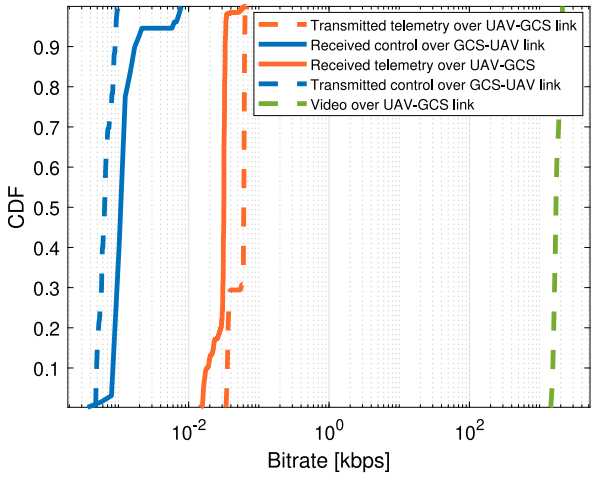


Fig. 9. CDF of network bitrate for telemetry, control and video traffic when drone is connected to the ground control station through a LTE network without a server (*LTEdirect* scheme).

shown in Fig. 6. The reason for this padding (e.g., error protection, packet tagging, data enrichment, ...) could not be identified, as it is not documented.

To check the impact of the selected connection scheme on service performance, Fig. 9 depicts the CDF of bitrate for video, telemetry and control data flows when UAV and GCS are connected directly (*LTEdirect*) with the default video encoding configuration. It can be observed that, unlike *LTEserver* scheme, the bitrate of telemetry traffic is similar in both links, revealing the absence of aggressive padding in the UAV. The same behavior is obtained with *WiFidirect* configuration (bitrate CDFs for this set-up have not been included here for brevity), suggesting that aggressive padding observed in *LTEserver* scheme might be introduced by XBStation software.

To analyze latency performance, Fig. 10 presents the CDF of end-to-end packet delay experienced by video traffic in the six configurations tested (three connectivity schemes and two video encoder settings) tested. Solid and dashed lines show results obtained with the encoder in normal and low-latency modes, respectively. It is observed that *LTEserver* scheme has the largest packet delays, while *WiFidirect* offers the lowest delays. Specifically, for the default video encoding scheme (worst case), the average packet delay is 500 ms for *LTEserver*, 42 ms for *LTEdirect* and 4 ms for *WiFidirect*. This fact suggests that

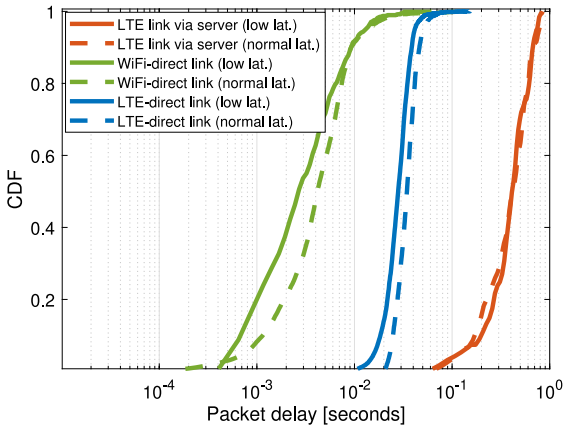


Fig. 10. CDF of packet delay for video traffic with different connectivity schemes and coding settings.

the predominant cause of packet delays for this set-up is the cloud connection. It has been checked that the used server is located in Iowa (USA), 7300 km away from the lab where experiments were carried out. Moving the server closer would lead to a reduction in delay and energy consumption [76]. Likewise, it is observed that enabling low-latency encoding decreases packet delay by 10% for *LTEdirect* and *WiFidirect* schemes. For instance, packet delay in *LTEdirect* is 42 ms with default encoding and 30 ms in low-latency mode. In contrast, in *LTEserver* configuration, a negligible difference is observed, as packet delay is dominated by the distance to xBCloud server. Note that low-latency mode forces that the video encoder only sends I (intra-coded) frames, eliminating differential inter-frame coding and reducing video processing delay. However, this should not affect latency at packet level.

A more comprehensive analysis of the packet arrival process in normal and low latency modes has been carried out. It was observed that, in normal mode, video chunks transmitted by the source encoder arrive every 2–3 s. Each chunk is further divided into short bursts, the first of which is much larger than the rest, corresponding to information that the codec delivers at its output at the same time. The bitrate, calculated as the total data volume divided by session duration, is 385 kbps. In contrast, in low latency mode, chunks also arrive every 2–3 s, but these are divided into small short bursts very similar in size and evenly spaced in time. The bitrate is higher, 597 kbps. This is consistent to the fact that the video encoder only uses intraframe encoding (I-frames only) to avoid the delay of storing previous frames and reordering frames in inter-frame prediction (P-frames and B-frames). The absence of large bursts of data arriving at the transmitter buffer, and the faster coding since frames are coded as they are captured, reduces latency. The price to be paid is that frame size (and bitrate) needs to be larger for the same image quality, since video coding is less effective as temporal redundancy is not eliminated.

From the above results, it can be concluded that WiFi-based connectivity solutions allows to offer FPV service with lower delays than LTE-based solutions, since connection is direct. However, WiFi has stringent coverage area restrictions and high sensitivity to interference and jamming, which prevents its use for industrial (i.e., non-recreational) BVLOS drone communications. Among solutions based on LTE, *LTEdirect* is the preferred option, since it leads to less bitrate and provides a lower latency. For this reason, *LTEdirect* configuration is selected for the remaining experiments.

6.2. Experiment 2 and 3

Figs. 11 and 12 summarize the impact of impairments introduced by the network emulator on QoE estimates obtained with the proposed

model. To ease comparison, both figures combine results of experiments 2 (static drone) and 3 (flying drone). In both figures, red curves corresponds to static drone and blue curve to flying drone. We combine the results of both experiments in the same figures to better understand the differences. The results are similar and should be so, since experiment 2 is a previous experiment and validation experiment to experiment 3. The point to keep in mind is that in any scope of implementation of a given system, a methodology should be followed in which step by step the promised scenario is achieved.

In Fig. 11, it is observed that packet loss has a strong impact on QoE for FPV service for both static and flying drone. Recall that, since video traffic is carried by the RTP protocol over UDP, there is no packet correction or retransmission mechanism. As expected, the best QoE is obtained without packet losses. The maximum MOS value is below 5, because the original image resolution is not the best and a residual latency is introduced by elements other than the network emulator (detailed later). More interestingly, the maximum MOS when drone is static is 0.5 larger than when flying. Such a difference suggests that image quality estimated by VMAF is worse for flying drones. This is mainly due to a higher image complexity introduced by the real drone movement in experiment 3 compared to the sequence selected for experiment 2, which makes image varies more frequently. The increased temporal entropy degrades the effectiveness of motion-compensated differential video coding, causing blocking artifacts in the presence of bitrate limitations [10]. Nonetheless, the evolution of QoE_{FPV} with packet loss ratio is similar for static and flying drone. The average MOS value of 3 (i.e., acceptable QoE) is obtained when packet loss ratio is 0.2%. However, MOS quickly decreases for packet loss ratios above 0.2%. Such a large sensitivity is due to the fact that video packets are encapsulated over UDP, a protocol without a loss recovery mechanism.

Similarly, Fig. 12 illustrates the QoE for different packet delays added by the network emulator. To aid the interpretation of results, the average video frame latency for flying drone is included on the secondary axis. Note that the total latency is the sum of the residual latency without NetEm plus the added latency introduced by NetEm. In principle, the residual latency might vary with radio link and core network conditions. However, in practice, the residual latency in experiments proved to be similar in all cases. This is the reason why the total latency looks like a straight line. A closer analysis shows that the residual latency varies from 45 and 65 ms across 3-min tests (53 ms on average). For clarity of the results visualization, it should be noted that this secondary curve has not been added for the static drone (experiment 2). It can be seen that, starting from the system without adding latency (impairments), we would have a latency of about 53 ms and a good quality of experience. The higher the total average latency, the lower the quality of experience of the system. It can be seen how from about 300 ms total latency, the quality of the system is considered bad. By comparing Figs. 11 and 12, it can be deduced that user experience perceived for video frame latency of 300 ms is similar to that experienced with a packet loss of 0.3%.

By using (1)–(3), measurements in Figs. 11 and 12 can be combined to infer the joint influence of packet losses and video frame latency on pilot satisfaction, provided that video frame latency does not affect image quality. Fig. 13 shows the resulting surface showing the joint impact of packet loss ratio and average video frame latency on $QoE_{ImgQual}$. It is observed that the maximum value is 3.443 corresponding to $maxQoE_{ImgQual}$ = 4.32 due to the limited video resolution (1080 × 720 pixels) and screen size (4.5 inch).

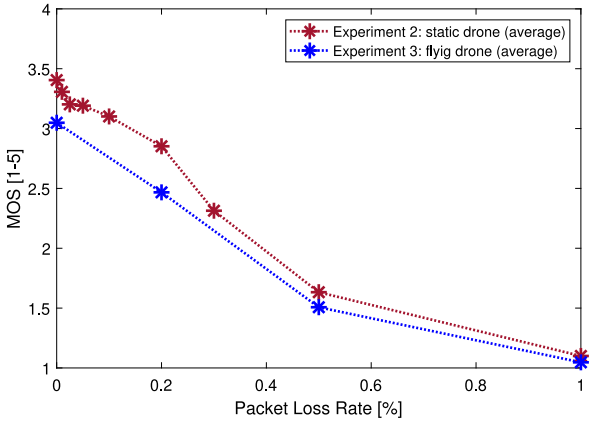


Fig. 11. Average MOS value estimated by the proposed QoE_{FPV} model with static or flying UAV for different packet loss ratios (experiments 2 and 3).

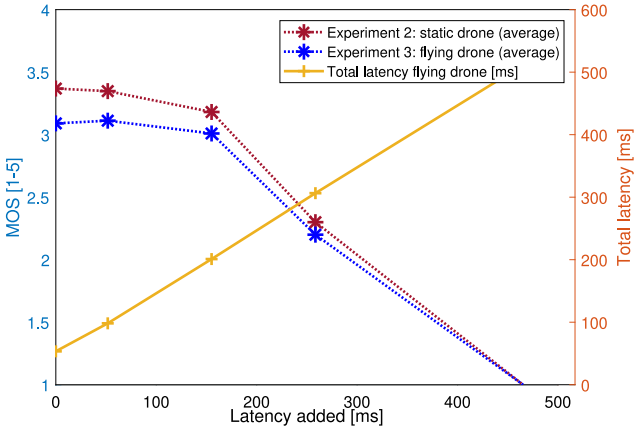


Fig. 12. Average MOS value estimated by the proposed QoE_{FPV} model with static or flying UAV for different added packet delays (experiments 2 and 3).

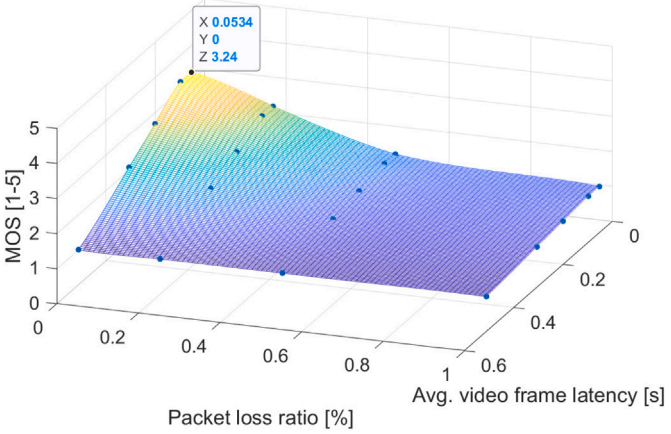


Fig. 13. Average MOS value estimated by the proposed QoE_{FPV} model with combination of packet loss and latency added impairments.

7. Conclusions

Several upcoming 5G services will rely on UAVs acting as terminals in beyond visual line-of-sight scenarios. One of these services is first person view drone control. This work has presented the choice and assembly of the components to set-up a drone for FPV service. Different LTE-based and WiFi-based connectivity solutions between the UAV and the ground control station have been compared. Likewise, two compression settings related to video latency have been tested. Then, control, telemetry and video traffic flows have been analyzed in a real testbed based on a pilot LTE network. Moreover, an analytical QoE model has been proposed for FPV service relying on video image quality and latency. The model has been assessed over the above-mentioned testbed in different UAV mobility and network conditions.

A first experiment has been performed to understand the protocols and the characteristics of the packets transmitted in FPV services considering different connections set-ups. Network measurements have shown that the server-based scheme, despite being flexible and easy to use, has the largest latency, since the server may be located far from the pilot LTE network accessed by the UAV. Moreover, with this scheme, telemetry packets are filled with redundant data, increasing the required bitrate. When a serverless UAV-GCS connection is established based on LTE or WiFi, latency and telemetry packet size are reduced, with a WiFi peer-to-peer scheme showing the lowest latency. The main advantage of the server-based scheme is that it is easier to use and has more flexibility in terms of functionality, since it has its own application for service configuration and monitoring. Then, the following experiments have allowed to understand the sensitivity of the system to different channel conditions. Specifically, tests have proved the large impact of packet loss and latency on FPV service usability, suggesting that QoE_{FPV} can only be achieved with a packet ratio below 0.2% and end-to-end-latency below 250 ms.

This work is a first step towards the future definition of a quality of experience methodology covering different use cases of cellular-based UAV service. It is left for future work the validation of the proposed QoE model with surveys from pilots in flight sessions. It would be interesting to check if the different traffic components change in real flights. In particular, it should be checked the linear degradation of MOS with increasing latency and how channel variability affects results. Likewise, it should be checked to which extent image quality indexes obtained with the latest video quality assessment methods reflect motion sickness, which is critical in teleoperated services [35]. More sophisticated models should also take into account the QoE of command/control and telemetry traffic flow. Likewise, tests should be repeated in a pilot 5G network guaranteeing high bandwidth and ultra-low latency.

Experiments presented here were performed with LTE and WiFi 802.11ac radio interfaces in the absence of a 5G/6G network in the lab. Low latency (< 1 ms) and high bandwidth (> 100 Mbps) requirements in the latter pave the way for latency-critical missions such as BVLOS drone control. It is left for future work the comparison of drone FPV control performance in terms of QoE between LTE and 5G systems. It is expected that the combination of new radio functionalities for Industrial Internet of Things and URLLC support (e.g., mini-slots, packet delay budget scheduling, frequent monitoring of control channel, new control channel formats...) [77], together with enhanced mobile edge computing (e.g., edge relocation) [78] and low-latency video codecs (e.g., mezzanine compression) [79], should ensure latency values below 10 ms required by human–machine interfaces for teleoperation services.

CRediT authorship contribution statement

N. González: Data curation, Formal analysis, Investigation, Resources, Software, Validation, Visualization, Writing – original draft. **M. Solera:** Conceptualization, Formal analysis, Methodology, Writing – review & editing. **F. Ruiz:** Resources, Supervision, Software, Validation, Writing – review & editing. **C. Gijón:** Data curation, Investigation, Validation, Visualization, Writing – original draft. **M. Toril:** Conceptualization, Funding acquisition, Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

None.

Data availability

The data that has been used is confidential.

Acknowledgments

This work has been funded by the Spanish Ministry of Science, Innovation and Universities, Spain (PID2021-122217OB-I00 and RTI2018-099148-B-I00).

References

[1] N. Mehta, K. Clement, T. Gareau, 5G!drones 857031 D1.1 use case specifications and requirements, 2019, Online. https://5gdrones.eu/wp-content/uploads/2020/05/D1.1-Use-case-specifications-and-requirements-v1.0.pdf. (Accessed: 12 Jul. 2021).

[2] L. Bertizzolo, T.X. Tran, J. Buczek, B. Balasubramanian, R. Jana, Y. Zhou, T. Melodia, Streaming from the air: Enabling drone-sourced video streaming applications on 5G open-RAN architectures, IEEE Trans. Mob. Comput. (2021).

[3] 3GPP, 3GPP TS 22.261, service requirements for the 5G system, 2020, https://www.etsi.org/deliver/etsi_ts/122200_122299/122261/16.14.00_60/ts_122261v161400p.pdf. (Accessed 8 Mar. 2022).

[4] N. Smolyanskiy, M. Gonzalez-Franco, Stereoscopic first person view system for drone navigation, Front. Robot. Artif. Intell. 4 (2017).

[5] A. Masaracchia, Y. Li, K.K. Nguyen, C. Yin, S.R. Khosravirad, D.B. Da Costa, T.Q. Duong, UAV-enabled ultra-reliable low-latency communications for 6G: A comprehensive survey, IEEE Access 9 (2021) 137338–137352.

[6] C. She, C. Liu, T.Q. Quek, C. Yang, Y. Li, Ultra-reliable and low-latency communications in unmanned aerial vehicle communication systems, IEEE Trans. Commun. 67 (5) (2019) 3768–3781.

[7] A. Koubâa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, M. Khalgui, Micro air vehicle link (mavlink) in a nutshell: A survey, IEEE Access 7 (2019) 87658–87680, http://dx.doi.org/10.1109/ACCESS.2019.2924410.

[8] T. Hoßfeld, R. Schatz, U.R. Krieger, QoE of YouTube video streaming for current internet transport protocols, in: International Conference on Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance, Springer, 2014, pp. 136–150.

[9] ITU-T, Subjective video quality assessment methods for multimedia applications, 2022, Rec. P.910.

[10] A. Raake, J. Gustafsson, S. Argyropoulos, M.-N. Garcia, D. Lindegren, G. Heikkilä, M. Pettersson, P. List, B. Feiten, IP-based mobile and fixed network audiovisual media services, IEEE Signal Process. Mag. 28 (6) (2011) 68–79.

[11] N. Barman, M.G. Martini, QoE modeling for HTTP adaptive video streaming–a survey and open challenges, Ieee Access 7 (2019) 30831–30859.

[12] L.R. Jiménez, M. Solera, M. Toril, A network-layer QoE model for YouTube live in wireless networks, IEEE Access 7 (2019) 70237–70252.

[13] L.R. Jiménez, M. Solera, M. Toril, S. Luna-Ramírez, J.L. Bejarano-Luque, The upstream matters: Impact of uplink performance on YouTube 360 live video streaming in LTE, IEEE Access 9 (2021) 123245–123259.

[14] GitHub, Measuring perceptual video quality with VMAF, 2017, https://github.com/Netflix/vmaf/blob/master/resource/doc/presentations/VMAF_ICIP17.pdf. (Online. Accessed 25 Jan. 2022).

[15] N. González Serrato, M. Solera Delgado, F. Ruiz Vega, C. Gijón Martín, M. Toril Genovés, A quality of experience evaluation methodology for first-person-view drone control in cellular networks, in: Proceedings of the 19th ACM International Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks, 2022, pp. 59–66.

[16] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L.G. Giordano, A. Garcia-Rodriguez, J. Yuan, Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges, IEEE Commun. Surv. Tutor. 21 (4) (2019) 3417–3442.

[17] S. Si-Mohammed, M. Bouaziz, H. Hellaoui, O. Bekkouche, A. Ksentini, T. Taleb, L. Tomaszewski, T. Lutz, G. Srinivasan, T. Jarvet, P. Montowtt, Supporting unmanned aerial vehicle services in 5G networks: New high-level architecture integrating 5G with U-space, IEEE Veh. Technol. Mag. 16 (1) (2021) 57–65, http://dx.doi.org/10.1109/MVT.2020.3036374.

[18] T. Taleb, A. Ksentini, H. Hellaoui, O. Bekkouche, On supporting UAV based services in 5G and beyond mobile systems, IEEE Netw. 35 (4) (2021) 220–227, http://dx.doi.org/10.1109/MNET.021.2000358.

[19] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, M. Debbah, A tutorial on UAVs for wireless networks: Applications, challenges, and open problems, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2334–2360.

[20] R. Amorim, H. Nguyen, J. Wigard, I.Z. Kovács, T.B. Sørensen, D.Z. Biro, M. Sørensen, P. Mogensen, Measured uplink interference caused by aerial vehicles in LTE cellular networks, IEEE Wirel. Commun. Lett. 7 (6) (2018) 958–961.

[21] Y. Zeng, J. Lyu, R. Zhang, Cellular-connected UAV: Potential, challenges, and promising technologies, IEEE Wirel. Commun. 26 (1) (2018) 120–127.

[22] Y. Zeng, Q. Wu, R. Zhang, Accessing from the sky: A tutorial on UAV communications for 5G and beyond, Proc. IEEE 107 (12) (2019) 2327–2375.

[23] A. Chakraborty, E. Chai, K. Sundaresan, A. Khojastepour, S. Rangarajan, SkyRAN: A self-organizing LTE ran in the sky, in: Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies, 2018, pp. 280–292.

[24] D. Mishra, E. Natalizio, A survey on cellular-connected UAVs: Design challenges, enabling 5G/B5G innovations, and experimental advancements, Comput. Netw. 182 (2020) 107451–107483.

[25] W. Khawaja, I. Guvenc, D.W. Matolak, U.-C. Fiebig, N. Schneckenburger, A survey of air-to-ground propagation channel modeling for unmanned aerial vehicles, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2361–2391.

[26] L. Ferranti, F. Cuomo, S. Colonnese, T. Melodia, Drone cellular networks: Enhancing the quality of experience of video streaming applications, Ad Hoc Netw. 80 (2018) 130–141.

[27] X. Lin, V. Yajnanarayana, S.D. Muruganathan, S. Gao, H. Asplund, H.-L. Maattanen, M. Bergstrom, S. Euler, Y.-P.E. Wang, The sky is not the limit: LTE for unmanned aerial vehicles, IEEE Commun. Mag. 56 (4) (2018) 204–210.

[28] M.M. Azari, F. Rosas, S. Pollin, Cellular connectivity for UAVs: Network modeling, performance analysis, and design guidelines, IEEE Trans. Wireless Commun. 18 (7) (2019) 3366–3381.

[29] L. Bertizzolo, T.X. Tran, B. Amento, B. Balasubramanian, R. Jana, H. Purdy, Y. Zhou, T. Melodia, Live and let live: Flying UAVs without affecting terrestrial UEs, in: Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications, 2020, pp. 21–26.

[30] A. Stornig, A. Fakhreddine, H. Hellwagner, P. Popovski, C. Bettstetter, Video quality and latency for UAV teleoperation over LTE: A study with ns3, in: 2021 IEEE 93rd Vehicular Technology Conference, VTC2021-Spring, IEEE, 2021, pp. 1–7.

[31] X. Lin, R. Wiren, S. Euler, A. Sadam, H.-L. Määttänen, S. Muruganathan, S. Gao, Y.-P.E. Wang, J. Kauppi, Z. Zou, et al., Mobile network-connected drones: Field trials, simulations, and design insights, IEEE Veh. Technol. Mag. 14 (3) (2019) 115–125.

[32] Vodafone, Vodafone beyond visual line of sight drone trial report, 2019, https://www.vodafone.com/content/dam/vodcom/images/vdf_images_2019/what-we-do/technology/connected-drones/phase1.pdf. (Accessed 7 Mar. 2022).

[33] R.K. Mok, E.W. Chan, R.K. Chang, Measuring the quality of experience of HTTP video streaming, in: IFIP/IEEE International Symposium on Integrated Network Management, IM, 2011, pp. 485–492.

[34] F. Wamser, P. Casas, M. Seufert, C. Moldovan, P. Tran-Gia, T. Hossfeld, Modeling the YouTube stack: From packets to quality of experience, Comput. Netw. 109 (2016) 211–224.

[35] M. Silic, M. Suznjevic, L. Skorin-Kapov, Qoe assessment of FPV drone control in a cloud gaming based simulation, in: 2021 13th International Conference on Quality of Multimedia Experience, QoMEX, 2021, pp. 175–180, http://dx.doi.org/10.1109/QoMEX51781.2021.9465385.

[36] J. De Vriendt, D. De Vleeschauwer, D. Robinson, Model for estimating QoE of video delivered using HTTP adaptive streaming, in: 2013 IFIP/IEEE International Symposium on Integrated Network Management, IM 2013, IEEE, 2013, pp. 1288–1293.

[37] M. Claeys, S. Latre, J. Famaey, F. De Turck, Design and evaluation of a self-learning HTTP adaptive video streaming client, IEEE Commun. Lett. 18 (4) (2014) 716–719.

[38] W. Huang, Y. Zhou, X. Xie, D. Wu, M. Chen, E. Ngai, Buffer state is enough: Simplifying the design of QoE-aware HTTP adaptive video streaming, IEEE Trans. Broadcast. 64 (2) (2018) 590–601.

[39] A. Raake, M.-N. Garcia, W. Robitza, P. List, S. Göring, B. Feiten, A bitstream-based, scalable video-quality model for HTTP adaptive streaming: ITU-T P. 1203.1, in: 2017 Ninth International Conference on Quality of Multimedia Experience, QoMEX, IEEE, 2017, pp. 1–6.

[40] W. Robitza, S. Göring, A. Raake, D. Lindegren, G. Heikkilä, J. Gustafsson, P. List, B. Feiten, U. Wüstenhagen, M.-N. Garcia, et al., HTTP adaptive streaming QoE estimation with ITU-T rec. P. 1203: Open databases and software, in: Proceedings of the 9th ACM Multimedia Systems Conference, 2018, pp. 466–471.

[41] H.-F. Bermudez, J.-M. Martinez-Caro, R. Sanchez-Iborra, J.L. Arciniegas, M.-D. Cano, Live video-streaming evaluation using the ITU-T P. 1203 QoE model in LTE networks, Comput. Netw. 165 (2019) 106967.

[42] M. Naveed, S. Qazi, S.M. Atif, B.A. Khawaja, M. Mustaqim, SCRAS server-based crosslayer rate-adaptive video streaming over 4G-LTE for UAV-based surveillance applications, Electronics 8 (8) (2019) 910.

[43] Y. Zhang, C. Xu, I.A. Hemadeh, M. El-Hajjar, L. Hanzo, Near-instantaneously adaptive multi-set space-time shift keying for UAV-aided video surveillance, IEEE Trans. Veh. Technol. 69 (11) (2020) 12843–12856.

[44] R. Muzaffar, E. Yanmaz, C. Raffelsberger, C. Bettstetter, A. Cavallaro, Live multicast video streaming from drones: An experimental study, Auton. Robot. 44 (1) (2020) 75–91.

[45] L.A. binti Burhanuddin, X. Liu, Y. Deng, U. Challita, A. Zahemszky, QoE optimization for live video streaming in UAV-to-UAV communications via deep reinforcement learning, IEEE Trans. Veh. Technol. 71 (5) (2022) 5358–5370.

[46] S. Li, Y. Zhang, S. Edwards, P.T. Blythe, Exploration into the needs and requirements of the remote driver when teleoperating the 5G-enabled level 4 automated vehicle in the real world: A case study of 5G connected and automated logistics, Sensors 23 (2) (2023) http://dx.doi.org/10.3390/s23020820, URL https://www.mdpi.com/1424-8220/23/2/820.

[47] B. Huang, D. Bayazit, D. Ullman, N. Gopalan, S. Tellex, Flight, camera, action! using natural language and mixed reality to control a drone, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 6949–6956.

[48] H. Courtois, N. Aouf, K. Ahiska, M. Cecotti, OAST: Obstacle avoidance system for teleoperation of UAVs, IEEE Trans. Hum.-Mach. Syst. 52 (2) (2022) 157–168, http://dx.doi.org/10.1109/THMS.2022.3142107.

[49] R. Inam, N. Schrammar, K. Wang, A. Karapantelakis, L. Mokrushin, A.V. Feljan, E. Fersman, Feasibility assessment to realise vehicle teleoperation using cellular networks, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems, ITSC, IEEE, 2016, pp. 2254–2260.

[50] A. Patil, H. Sawant, Technical specification group services and system aspects IP multimedia subsystem (IMS), Int. J. Electron. Commun. Comput. Eng. 3 (2) (2012) 234–238.

[51] G. Geraci, A. Garcia-Rodriguez, M.M. Azari, A. Lozano, M. Mezzavilla, S. Chatzinotas, Y. Chen, S. Rangan, M. Di Renzo, What will the future of UAV cellular communications be? A flight from 5G to 6G, 2021, arXiv preprint arXiv:2105.04842.

[52] L. Afonso, N. Souto, P. Sebastiao, M. Ribeiro, T. Tavares, R. Marinheiro, Cellular for the skies: Exploiting mobile network infrastructure for low altitude air-to-ground communications, IEEE Aerosp. Electron. Syst. Mag. 31 (8) (2016) 4–11.

[53] M.A. Zulkifley, M. Behjati, R. Nordin, M.S. Zakaria, Mobile network performance and technical feasibility of LTE-powered unmanned aerial vehicle, Sensors 21 (8) (2021) 2848.

[54] H. Zhou, F. Hu, M. Juras, A.B. Mehta, Y. Deng, Real-time video streaming and control of cellular-connected uav system: Prototype and performance evaluation, IEEE Wirel. Commun. Lett. (2021).

[55] RC Innovations, Pixhawk 2.1 standard set the cube orange (receptor ADS-B), 2021, Online. https://rc-innovations.es/pixhawk-2.1-standard-set-the-cube-orange-receptor-ads-b-hx4-06100-hex-technology. (Accessed 12 Jul. 2021).

[56] XBStation, Xbstation platform for internet drone base on real time 4G/5G connectivity, 2021, Online. https://xbstation.com. (Accessed: 12 Jul. 2021).

[57] XBStation, Xbstation 4G LTE base module, 2021, Online. https://store.xbstation.com/xbstation-4g-lte-base-module. (Accessed: 12 Jul. 2021).

[58] ArduPilot, Mission planner home, 2021, Online. https://ardupilot.org/planner/. (Accessed: 12 Jul. 2021).

[59] FFmpeg Corp, FFmpeg, 2023, Online. http://ffmpeg.org. (Accessed: 11 Ap. 2023).

[60] IETF RFC 3550, RTP: A transport protocol for real-time applications, 2003, (Accessed 8 Mar. 2022).

[61] 3GPP, 33Gpp TS 22.829, technical specification group services and system aspects; enhancements for unmanned aerial vehicles, 2019, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3557. (Accessed 8 Mar. 2022).

[62] H. iLab, Video experience-based bearer network technical white paper, 2016, Online. https://www.huawei.com/~/media/CORPORATE/PDF/white%20paper/video-experience-based-bearer-network-technical-whitepaper. (Accessed 12 Jul. 2021).

[63] L. Skorin-Kapov, M. Varela, T. Hoßfeld, K.-T. Chen, A survey of emerging concepts and challenges for QoE management of multimedia services, ACM Trans. Multim. Comput. Commun. Appl. (TOMM) 14 (2s) (2018) 1–29.

[64] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, M. Manohara, Toward a practical perceptual video quality metric, Netflix Tech Blog 6 (2) (2016).

[65] H.R. Sheikh, A.C. Bovik, Image information and visual quality, IEEE Trans. Image Process. 15 (2) (2006) 430–444.

[66] S. Li, F. Zhang, L. Ma, K.N. Ngan, Image quality assessment by separately evaluating detail losses and additive impairments, IEEE Trans. Multimed. 13 (5) (2011) 935–949, http://dx.doi.org/10.1109/TMM.2011.2152382.

[67] J. Li, L. Krasula, Y. Baveye, Z. Li, P. Le Callet, Accann: A new subjective assessment methodology for measuring acceptability and annoyance of quality of experience, IEEE Trans. Multimed. 21 (10) (2019) 2589–2602.

[68] B. García, L. López-Fernández, F. Gortázar, M. Gallego, Practical evaluation of VMAF perceptual video quality for WebRTC applications, Electronics 8 (8) (2019) 854.

[69] F. Zhang, F.M. Moss, R. Baddeley, D.R. Bull, BVI-HD: A video quality database for HEVC compressed and texture synthesized content, IEEE Trans. Multimed. 20 (10) (2018) 2620–2630.

[70] GitHub, Python tool based on ffmpeg and ffprobe to deal with the video pre-procesing required for VMAF inputs, 2019, Online. https://github.com/gdavila/easyVmaf. (Accessed: 7 Mar. 2022).

[71] M. Orduna, C. Díaz, L. Muñoz, P. Pérez, I. Benito, N. García, Video multimethod assessment fusion (VMAF) on 360VR contents, IEEE Trans. Consum. Electron. 66 (1) (2019) 22–31.

[72] J. Benjak, D. Hofman, J. Knezović, M. Žagar, Performance comparison of H. 264 and H. 265 encoders in a 4K FPV drone piloting system, Appl. Sci. 12 (13) (2022) 6386.

[73] R. Rassool, VMAF reproducibility: Validating a perceptual practical video quality metric, in: 2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB, IEEE, 2017, pp. 1–2.

[74] 3GPP, 3GPP TS 22.825, study on remote identification of unmanned aerial systems, 2018, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3527. (Accessed 13 Jul 2023).

[75] Linux Foundation, NetEm, 2021, Online. https://man7.org/linux/man-pages/man8/tc-netem.8.html. (Accessed: 12 Jul. 2021).

[76] C. Caiazza, S. Giordano, V. Luconi, A. Vecchio, Edge computing vs centralized cloud: Impact of communication latency on the energy consumption of LTE terminal nodes, Comput. Commun. 194 (2022) 213–225.
[77] D. Yang, A. Mahmood, S.A. Hassan, M. Gidlund, Guest editorial: Industrial IoT and sensor networks in 5G-and-beyond wireless communication, IEEE Trans. Ind. Inform. 18 (6) (2022) 4118–4121, http://dx.doi.org/10.1109/TII.2022.3142149.
[78] 3GPP, Technical specification (TS) 23.558, architecture for enabling edge applications; (release 18), 2022, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3723. (Accessed 13 Jul. 2023).
[79] J. Žádník, M. Mäkitalo, J. Vanne, P. Jääskeläinen, Image and video coding techniques for ultra-low latency, ACM Comput. Surv. 54 (11s) (2022) http://dx.doi.org/10.1145/3512342.

Nuria González received her B.Sc. degree in Telecommunication Systems Engineering and M.Sc. double degree in Telecommunication Engineering and in Telematics and Telecommunication Networks from the University of Malaga, Spain, in 2018 and 2021, respectively. Currently, she is working as a research assistant at the University of Malaga. Her research interests include radio resource management, machine learning and data analytics.

Marta Solera received the M.Sc. and Ph.D. degrees in telecommunication engineering from the Polytechnic University of Catalonia (UPC), in 1996 and 2006, respectively. She is currently an Associate Professor with the Department of Communication Engineering, Universidad de Malaga (UMA). Since 1996, she has been a Lecturer in several universities, such as UPC, the Universidad Nacional Autonoma de Mexico (UNAM), and UMA. She has been involved in several public funded national research projects in the field of multimedia and mobile communications. Her research interest includes design and performance evaluation of multimedia services over mobile networks.

Fernando Ruiz received the Ingeniero de Telecomunicacion degree from the Universidad de Málaga, Málaga, Spain, in 1994. After working in CETECOM testing laboratory with type-approval test systems for communication equipment, he joined the Departamento de Ingeniería de Comunicaciones, Universidad de Málaga, where, in 2001, he became an Associate Professor. In 1995 he was a Visiting Researcher with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, UK. From 2000 to 2002 he took part in the Nokia Mobile Communications Competence Centre, Málaga, Spain. He received the Ph.D. Degree on channel characterization and his research interests include mobile radio and underwater acoustic channel characterization and emulation.

Carolina Gijón received her B.Sc. degree in Telecommunication Systems Engineering and her M.Sc. Degree in Telecommunication Engineering from the University of Malaga, Spain, in 2016 and 2018, respectively. Currently, she is working towards the Ph.D. degree. Her research interests include self-organizing networks, machine learning and radio resource management.

Matías Toril received the M.S. and Ph.D. degrees in telecommunication engineering from the University of Malaga, Spain, in 1995 and 2007, respectively. Since 1997, he has been a Lecturer with the Communications Engineering Department, University of Malaga, where he is currently an Associate Professor. He has authored more than 100 publications in leading conferences and journals, and he holds three patents owned by Nokia Corporation. His current research interests include self-organizing networks, radio resource management, and graph partitioning.