VR-Based Immersive Service Management in B5G Mobile Systems: A UAV Command and Control Use Case

Tarik Taleb, *Senior Member, IEEE*, Nassim Sehad, Zinelaabidine Nadir, and Jaeseung Song, *Senior Member, IEEE*

***Abstract*—The management of remote services, such as remote surgery, remote sensing, or remote driving, has become increasingly important, especially with the emerging 5G and Beyond 5G technologies. However, the strict network requirements of these remote services represent one of the major challenges that hinder their fast and large-scale deployment in critical infrastructures. This article addresses certain issues inherent in remote and immersive control of virtual reality (VR)-based unmanned aerial vehicles (UAVs), whereby a user remotely controls UAVs, equipped with 360° cameras, using their head-mounted devices (HMD) and their respective controllers. Remote and immersive control services, using 360° video streams, require much lower latency and higher throughput for true immersion and high service reliability. To assess and analyze these requirements, this article introduces a real-life testbed system that leverages different technologies (e.g., VR, 360° video streaming over 4G/5G, and edge computing). In the performance evaluation, different latency types are considered. They are namely: 1) glass-to-glass latency between the 360° camera of a remote UAV and the HMD display; 2) user/pilot's reaction latency; and 3) the command/execution latency. The obtained results indicate that the responsiveness (dubbed Glass-to-Reaction-to-Execution—GRE–latency) of a pilot, using our system, to a sudden event is within an acceptable range, i.e., around 900 ms.**

***Index Terms*—5G and beyond, edge computing, immersive services, mobile networking, unmanned aerial vehicle (UAV), virtual reality (VR).**

I. Introduction

NOWADAYS, we are witnessing very rapid developments toward innovative technologies in the field of telecommunications. Thanks to 4G mobile networks that have been widely used for the past years, and the great efforts devoted to the development of the Internet of Things (IoT), different sectors have been positively impacted [1], [2]. These sectors include health, education, smart cities [3], and agriculture [4], to name a few. In this vein, 5G networks are expected to continue paving the way for enabling different IoT applications [5], [6], particularly in the case of remote immersive services, and/or virtual reality (VR)-based applications, such as telesurgery, self-driving, and UAV-based applications.

VR technology has recently received increasing interest in both academia and industry [7], [8]. It is rapidly progressing toward customers' use to bring an immersive experience to a variety of applications. It is worth noting that remote and highly mobile services, particularly the use of UAVs have gained great interest and will play an important role by becoming an enabling technology in IoT and next-generation networks [9], [10], [11].

However, one key challenge for the current networks to support these immersive services relies on their special peculiarities [12], [13]. These services require high throughput varying from 1.0 Gb/s up to 1.0 Tb/s, and ultralow latency of less than 10 ms [12], [14]. Sensitive and short response times are crucial for new latency-sensitive applications in the IoT, VR and augmented reality (AR), autonomous vehicular and UAV remote control applications, etc. For example, to remotely control a process under changing and critical conditions, very short-term information exchange is required between sensors, controllers, and actuators to achieve real-time high Quality of Control (QoC). Another potential challenge lies in the intensive computing required for capturing and encoding the volumetric scene of a remote location. Among other things, one of the encountered problems is, for instance, the choice of devices that a UAV can carry. In order to effectively control the UAVs, the weight of the onboard equipment must be taken into account. Therefore, a tradeoff between their performance and their lightness must then be considered judiciously.

Advanced 360° cameras are too heavy for a UAV to be able to take off or control accurately. Lightweight 360° cameras, on the other side, come with limited computing resources and capabilities. Thus, enabling edge computing will alleviate the computational burden of these lightweight devices. One of the many improvements that 5G promises, in addition to

Manuscript received 29 June 2021; revised 15 May 2022 and 14 September 2022; accepted 10 November 2022. Date of publication 15 November 2022; date of current version 7 March 2023. This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme through CHARITY Project under Grant 101016509; in part by the Academy of Finland 6Genesis Project under Grant 318927; and in part by the Idea-Mill under Grant 33593. The work of Jaeseung Song was supported by the Technology Innovation Program funded by the Ministry of Trade, Industry and Energy (MOTIE), South Korea, under Grant RS-2022-00154678. *(Corresponding author: Tarik Taleb.)*

Tarik Taleb is with the Faculty of Information Technology and Electrical Engineering, University of Oulu, 90570 Oulu, Finland, and also with the Department of Computer and Information Security, Sejong University, Seoul 15600, South Korea (e-mail: tarik.taleb@oulu.fi).

Nassim Sehad and Zinelaabidine Nadir are with the Department of Communications and Networking, Aalto University, 02600 Espoo, Finland (e-mail: nassim.sehad@aalto.fi; zinelaabidine.1.nadir@aalto.fi).

Jaeseung Song is with the Department of Computer and Information Security, Sejong University, Seoul 15600, South Korea (e-mail: jssong@sejong.ac.kr).

Digital Object Identifier 10.1109/JIOT.2022.3222282

enhancing new radio (NR) and millimeter wave (mmWave) and energy efficiency [15], is to incorporate network automation and micro-services-based architecture that could improve data rates, latency, and reliability [16]. Such an architecture will offer network elasticity and softwarization which will ease the deployment of edge services and enhance their portability in the sense that they are capable of running on any server [17], [18].

Previous studies provide only solutions for immersive UAV tele-existence, without considering the option of remotely controlling the device, using either a stereo camera [19] or an array of cameras [20]. However, these solutions consider only how to efficiently remote control the viewpoint from a head-mounted device (HMD) using a non 360° cameras, whereas the service's reliability and delay analysis are not considered. Furthermore, much of the research up to now deal with immersive 3-D video streaming and more specifically point cloud video delivery that provides a 6-Degree of Freedom (DoF) view experience [21], [22], [23], [24].

Most of the studies on immersive services focus only on leveraging view-port adaptive streaming to reduce bandwidth consumption with no consideration to streaming latency. In tile-based solutions, tiles of the user's Field of View (FoV) are assigned high priority [25]. Moreover, based on the HMD predictions, several versions of the video can be created and therefore, an optimal solution should be able to deliver the version with the highest quality [26].

To increase the quality of experience, the work in [27] created dynamic heatmaps representing the user's view probability of a navigated 360° video while taking network resources into account. Similarly, the experimental setup in [28] assessed the influence of 360° video geometric layouts variation with an analysis of how to incorporate the solution into MPEG DASH streaming [29]. From the perspective of this work, the authors stated that they would investigate how to apply the solution on live video streaming because generating different representations of the 360° video is extremely time consuming and difficult to implement on the go. All of the above-stated works are very promising for minimizing the bandwidth while increasing video quality, except that they are intended for Video on Demand which is not concerned with ultralow latency video delivery requirements.

Nevertheless, few articles have addressed the E2E latency of immersive real-time live media streaming from 360° video cameras, and there has been little quantitative analysis of end-to-end (E2E) delay measurements. In addition, very little consideration has been paid to the glass-to-glass (G2G) latency. For instance, the work in [30] proposed a live omnidirectional video streaming system leveraging view-port prediction and offloading the encoding to reduce bandwidth and latency. However, viewport prediction is time-consuming and its E2E latency exceeds 2 s. The work in [31] presented an efficient transcoding method for the 360° CCTV system, however, their chosen protocols and transcoding parameters lead to a delay reaching 7 s and is not suitable for real-time surveillance. The work in [32] proposed a rate-distortion scalable multicast system for live 360° streaming based on machine learning for viewport prediction, yet their system was not analyzed in terms of E2E latency which is an important parameter for live video. On the other hand, the work in [33] analyzed delays of their proposed 360° testbed that comprised an omnidirectional camera prototype. Their E2E delay results were quite good as of 800 ms, except in case of the low resolution of 1080p, owing to the use of HTTP-based streaming at the client-side.

To sum up, 360° video streaming is bandwidth- and compute-intensive and latency sensitive. Additionally, many articles have studied commercial streaming platforms that allow 360° video streaming, such as Facebook, and YouTube in terms of E2E latency [34], [35]. Most of these studies state that the current commercial platforms suffer from frequent rebuffering and a very high delay as of 25–60 s [35].

However, in the proposed system of this article, a full analysis of the E2E VR-based remote UAV control from a communication perspective is considered, with an average video streaming delay of 700 ms. The aim of this article is to study the behavior of VR-based remote UAV control under different network access technologies, and to provide a suitable software (SW) and hardware (HW) architecture as well as recommendations to achieve high reliability and full immersiveness. In this study, an initial investigation was carried out to choose the SW and HW that allow achieving an immersive service. Then, the architecture was defined with different HW and SW components for remotely controlling the UAV using either wireless or mobile networks. Furthermore, a practical method based on video multimethod assessment fusion (VMAF) [36] is provided to measure the objective video quality metric of the 360° video stream.

Moreover, a particular focus is given to E2E UAV command and control by developing an HW-based measurement tool. This tool measures the video streaming and control command latency which is referred in this article as "glass-to-reaction-to-execution latency" (GRE). GRE is the time from the moment an event (i.e., usually when a UAV approaches an object) has been captured by the UAV camera until the user's reaction has been executed by the UAV to avoid hitting that object. Note that this latency information is essential to be considered when navigating, especially when controlling the UAV remotely in areas requiring maximum responsiveness, such as cramped areas and/or areas with many obstacles. In such cases, the UAV approaching an obstacle must be highly responsive and must exploit the latency and dynamic information of the UAV, in order to avoid colliding with that object.

The core contributions of this article are summarized as follows.

1) Design the system that allows the remote control of a UAV using a suitable real-life HW and SW-based architecture for maximum immersion while allowing optimal and reliable control of UAVs.
2) Provide a detailed analysis of the GRE latency under different conditions, namely: a) mobile networks, such as LTE, 5G, and WiFi; b) user reactions; and c) video qualities. Furthermore, it is worth noting that to optimize the GRE latency and its measurement, an HW-based

technique is proposed to measure precisely the different delays (i.e, in contrast to HW-based technique, SW-based techniques affect the GRE latency).

3) Conduct the experiments in real-life testbed allowing exhaustive testing and validation. It is worth noting that our solution for remotely controlling a UAV with controllers and body movements based on 360° video feedback from a camera on the UAV is successfully implemented and experimented in real life.

The remainder of this article is organized as follows. In Section II, an overview of remote service use cases and their requirements is presented. In Section III, the description of the system architecture is detailed. In Section IV, extensive experiments are conducted alongside their analysis. Finally, this article concludes in Section V by highlighting the research challenges and future research directions.

II. Remote Services: Overview

Remote services are increasingly important enabling technologies for 5G/B5G networks. In fact, they provide the ability to remotely control, with a high rate of safety, heavy and precision-needy machinery, such as surgeries, medical services, or vehicles. A fundamental challenge of these services, for the most guaranteed services, lies in their strict connectivity requirements. Streaming of the remote location is needed to control the remote device, and upon receiving this stream, a user can control the remote device by sending the needed commands (e.g., hand gestures). These requirements allow no late or lost packets, as such an event could lead to a disruption or failure in the planned service. The required network latency is less than 5 ms with network reliability below 99.999 [37], [38].

The objective of remote services is to create a sense of presence between the user and the remote location. These remote services can adopt a 3-D or a traditional 2-D interactive video stream environment that is rendered in real time. Despite the fact that the word VR is associated nowadays with immersive VR, the latter can be also nonimmersive and allow the user to view a remote environment on a desktop or projection screen [39]. The degree of immersiveness depends on how much the user is isolated from the physical environment [40].

A. Nonimmersive Remote Services

Nonimmersive remote services are concretely deployed using fully or partially traditional interaction devices. In the medical field, a vocal cord 5G telesurgery based on robots was recently achieved using 5G and a multiaccess edge computing (MEC)-based core network from a distance of 15 km to the actual site of the surgery [41]. Since VR is now so widely used in daily life, nonimmersive virtual environments are often ignored in the VR category. This technology creates a computer-generated world, thus allowing the user to remain conscious of their physical surroundings and monitor them. VR systems that are not fully interactive rely on a computer or video game console, a monitor, and input devices such as keyboards.

Although nonimmersive services are more accepted by users since they offer less motion-sickness, deploying them provides weak interactivity to 2-D end-user devices (i.e., an equipment with a screen/monitor and simple input devices, such as keyboard and mouse). This results in an increase in latency caused by many extra functionalities such as pose redirection [19]. Alternatively, VR applications provide a full immersion of the users into the remote location. 360° video streaming delivers a more immersive viewing experience to end users. However, it faces tremendous challenges, because of the high resolution and the short response time requirements.

B. Immersive Remote Services

Immersive services provide an intuitive way to control remote robots, thereby transporting the user to a virtual remote space where they can control these robots via natural hand movements or through virtual interactions with them. Immersive technology integrates virtual content in a physical environment so that the user naturally interacts with mixed reality.

Nevertheless, high-quality immersive services require high-throughput, low latency, and reliable connectivity. However, one of the interests of 5G is to provide a higher uplink capacity than 4G, which partially addresses the quality and latency issues encountered in high-resolution streaming at 360°. Therefore, 5G ensures reliable remote control of devices, as well as increased efficiency for low risks in hazardous environments. Undoubtedly, immersive services are now thriving in many industries. The following sections present the different use cases of remote immersive services and their requirements.

1) Use Cases: Nowadays, VR and AR are thriving within some sectors as follows.

Manufacturing: Manufacturing is one of the most promising fields that is expected to benefit from VR. Industry 4.0 is going to automate traditional practices, and the help of IoT and machine-to-machine communications will improve communication and monitoring. Automotive companies use VR to improve the design of their vehicles. On the contrary, creating physical samples of cars to improve and test them has resulted in wastage of significant time and money. Now thanks to VR, manufacturers can make adjustments virtually and in real time. On the other hand, researchers are developing VR-based systems for remote manufacturing inspection and monitoring. For instance, in [42], a cyber–physical system (CPS) architecture, that uses WebVR, was developed to display real-time visual 3-D models based on IoT sensors' data and actuators, leading to optimized production and assistance. In [43], a novel concept was proposed for remote inspection and anomaly detection in manufacturing machines. It relies on 360° video streams with virtual embedded information. However, it uses a low-quality resolution stream (i.e., 720p), and both the camera and HMD were connected to the same computer Moreover, no analysis regarding the video stream or interaction latency was provided.

Training and Education: VR and AR technologies can help in education, especially after the outbreak of COVID-19 and

TABLE I
BENCHMARK OF DIFFERENT STREAMING PROTOCOLS

Test	Camera	Protocol	Resolution	Web support	Average 360° video E2E streaming latency (Seconds)
1	Webcam 2D	RTSP	1280x720 @ 30FPS	No	1s
2	Webcam 2D	RTMP	1280x720 @ 30FPS	No	2s
3	Webcam 2D	WebRTC	1280x720 @ 30FPS	Yes	0.1s
4	360° camera	HLS	4K(2560 x 1280) @ 30FPS	Yes	15s
5	360° camera	RTMP/RTSP over WebSockets	4K(2560 x 1280) @ 30FPS	Yes	7s
6	360° camera	RTMP/RTSP over WebRTC	4K(2560 x 1280) @ 30FPS	Yes	0.7s

the massive reliance on distance learning. This unexpected situation shows the need for interactive distance learning, which can be provided using VR/AR for more interactivity. Yet another example is learning to drive before trying a real car, visiting distant places and interacting with their objects and surroundings. In certain cases, it would be difficult to experience in real-life ocean exploration, or it would take more than half the globe to fly and explore worlds and galaxies. This area is expected to be one of the most affected by VR since it reproduces the user's natural spatial abilities. Feurstein et al. [44] presented the challenges and scenarios of integrating 360° video streaming in higher education for e-learning environments. One of the challenges they encountered is that they cannot offer HMDs to all viewers and, therefore, the solution should be, instead, Web-based using WebVR.

Smart Healthcare: VR has also a major impact on the smart healthcare sector. This interactive environment provides doctors the medium to create 3-D models of human anatomy, based on scans, to practice critical medical procedures. This is expected to improve the performance of the operations and the well-being of the patients [45]. In [46], a low-cost VR system for next-generation rehabilitation was developed. This system recognizes body motions, allowing the user to see his/her movements, and provides health status using the worn medical sensors. In [47], a proof-of-concept for a tele-medicine technology platform was presented. It provides an interactive experience through a 360° video in an augmented virtual world. However, this concept is not suitable for ultralow latency services as one of the top priorities stated in their perspectives is to reduce video delay.

UAV and Vehicle Control: Multimedia and VR technology can be also used to help first responders in natural and man-made disasters, whereby conditions are often extremely difficult for the human to reach the disaster area [48]. In such cases, UAVs can also help to measure damage, establish communications, and allow better visibility to the rescue teams. Erat et al. [49] investigated the possibility of navigating and manipulating an UAV indirectly from an exocentric perspective using drone-augmented human vision. However, their solution used low-resolution streamed images (640 × 480), and implemented all computations onboard the UAV.

C. Requirements and Challenges in 360° Video Streaming

360° applications are bandwidth intensive and, therefore, optimizing their E2E latency through the network is very challenging [12], [13]. Furthermore, E2E video latency depends strongly on the encoder, the target resolution, bit rate, and most importantly the streaming protocol, such as the real-time messaging protocol (RTMP), real-time streaming protocol (RTSP), or HTTP Live Streaming (also known as HLS) to name a few that are supported by most 360° cameras [50], [51], [52]. Moreover, compared to the standard streaming systems, interactive VR applications will push the connectivity requirements to their limits due to their strict requirements in terms of capacity, low latency (less than 50-ms latency), as well as consistent Quality of Service (QoS).

Consequently, the design of the system, presented in this article, has largely been influenced by the current implementation technologies related to immersive services and UAVs. Compared to lightweight cameras, advanced 360° cameras have extra functionalities (high frame rate, 8k streaming, internal RTMP server) but at the cost of high weight. Moreover, UAVs are only able to carry light cameras, and these light cameras are not suitable for such immersive services for various reasons, such as: 1) protocol dependencies, 2) HW and SW limitations; and 3) user-centric considerations. Furthermore, to obtain our optimized 360° video streaming architecture, as in Section III, many experiments were carried out at the beginning of this study to test and analyze various streaming protocols (using different types of media servers as presented in Table I). These protocols were tested on 2-D and 360° cameras. The Experiments were carried out with a perfect bandwidth link by connecting the server and the video client to the same network [53], [54].

Table I shows the results of these experiments. Overall, the Web real-time communication (WebRTC) protocol exhibits much lower G2G latency compared to the other protocols. However, current implementations of 360° cameras do not support streaming using this protocol. From these results, our design choice should follow the setup that supports 360° cameras and provide the lowest latency. Therefore, Experiment 6 represents the best option for building such system, with the following considerations:

Protocol Dependencies: Unfortunately, most of 360° cameras only support streaming via RTMP. These protocols were designed for broadcast purposes and they require the deployment of an RTMP server at the continuum public/edge cloud.

User-Centric Considerations: Due to the high mobility of UAVs and the effect of winds, this may cause instability in the video, and consequently, it may cause dizziness to the user of the VR HMD. Therefore, in such a case, HW or SW images

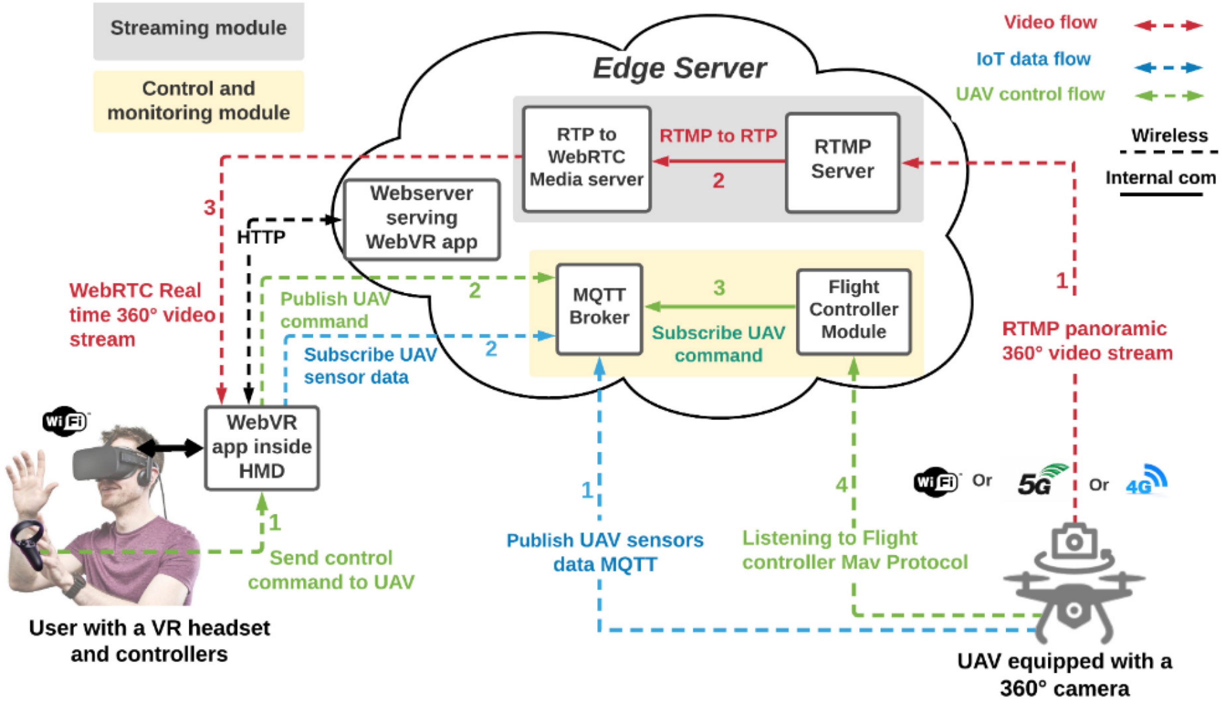

Streaming module
Control and monitoring module
Edge Server
Video flow
IoT data flow
UAV control flow
Wireless
Internal com
RTP to WebRTC Media server
RTMP to RTP
2
RTMP Server
Webserver serving WebVR app
HTTP
3
WebRTC Real time 360° video stream
Publish UAV command
2
Subscribe UAV sensor data
2
MQTT Broker
3
Subscribe UAV command
Flight Controller Module
1
RTMP panoramic 360° video stream
WebVR app inside HMD
1
Send control command to UAV
User with a VR headset and controllers
1
Publish UAV sensors data MQTT
4
Listening to Flight controller Mav Protocol
Or
5G
Or
4G
UAV equipped with a 360° camera

Fig. 1. High-level architecture of the envisioned system.

stabilizer should be used to provide more reliable video stream. However, using the SW stabilizer may contribute to the G2G latency.

HW and SW Limitations: UAVs can only carry lightweight 360° cameras. However, these cameras come with limited resources that provide streaming through a smartphone using WiFi direct. They provide low 4K streaming rates at a maximum of 30 frames per second (FPS). Their battery lasts for 1h and they heat up while streaming, which adds some delays.

Access Control: The issues of access control, security, and privacy are always a big concern for the users. Recently, the blockchain technology has been introduced as a viable solution for access control problems in decentralized systems [55].

Bandwidth: The current generation of 4K 360° videos requires 20–100 Mb/s of upload and download bandwidth, that is approximately 6x the bandwidth of a regular 2-D video streamed at a similar resolution [56]. Although 4G LTE can deliver this bandwidth, for a high-definition (HD) video, LTE needs to provide higher throughput and lower latency below 10 and 1-ms jitter per user [57]. Moreover, most of the approaches widely used in the literature to reduce the download bandwidth are analog to the novel solution, namely, the omnidirectional media format (OMAF) [58], established by the Moving Picture Experts Group (MPEG) [59], [60], [61], [62]. However, this latter is not tailored to low latency since the first MPEG-OMAF standard-compliant end-to-end chain for live VR360 [63] showed high E2E delay results of 6s for a 6K × 4K resolution video. However, the work in [64] proposed a tile-based viewport-dependent solution compliant with MPEG-OMAF and showed an end-to-end delay >1s.

Latency: One of the most critical requirements of immersive services is G2G or MTP motion-to-photon latency [53], [54]. VR/AR applications are the most latency-demanding applications, as the latency between the physical movement of a person's head and the updated photons on the HMD display reaching the person's eye must be very small. This is because human sensory systems can detect very small delays. To avoid this motion-to-photon issue, rendering and display latency at the HMD must be less than 15–20 ms. Furthermore, the overall E2E video delay must be less than 1 s in order to feel smooth interaction with the information [65]. In this work, we are focusing more on the E2E video delay.

Frame Rate: In order to have a smooth immersive experience, the frame rate must be fast (30–60 FPS). When input video sources have a fixed resolution (4K/8K), the video output viewed by the viewer must be within the appropriate range [66].

III. VR-Based UAV Remote Controlling

It is worth noting that our system is a real-life testbed that allows to remotely control a UAV using a VR HMD. As it will be shown, the testbed's latency results regarding the video and UAV control overcome most of the challenges discussed earlier.

A. Description

This section describes the proposed architecture functionalities for the VR-based remote control of a UAV with the aim of providing fully immersive services. Each component is presented alongside the interactions with the other components. Two key architectures are presented: 1) global and 2) detailed architectures (shown in Figs. 1 and 7, respectively). The global system's architecture in Fig. 1 shows the overall components of

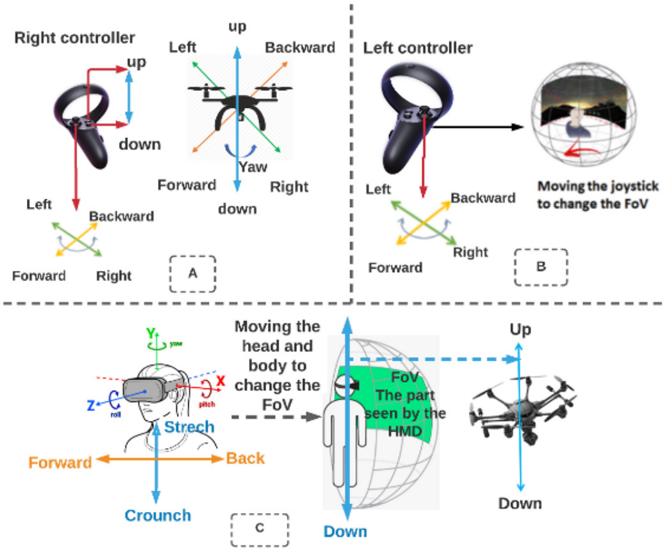


Fig. 2. Operations for the control of the FoV of a UAV. (a) UAV control with joystick. (b) FoV control with joystick. (c) FoV and UAV's altitude control with head and body movements.

the system, whereas the detailed architecture in Fig. 7 provides the measured delays and the complete testbed demonstrating the components' interactions in a chronological order.

Our architecture, which is a real-life implementation, includes microservices-based components at the edge server. Components are containerized in order to provide portability and scalability. The system aims at providing end users, the ability to control remote UAVs based on 360° stream and sensed data from the remote location. At the remote location, the UAV is equipped with a 360° camera that live streams to the user's HMD. On the user side, upon receiving the camera's stream, the user can control the remote UAV using body movements and HMD controllers. The components of the system are described as follows.

Remote UAV: The remote UAV is equipped with: 1) a single-board computer (SBC); 2) a flight controller; 3) IoT sensors; and 4) 360° camera. Both the SBC and the camera communicate with the system through WiFi, 5G, or 4G. The SBC controls the UAV by receiving the user commands from the edge applications and transmits them via USB to the flight controller. The 360° camera live streams an RTMP video to the VR edge application streaming module (FFmepg RTMP server [67]) through the SBC.

VR Users: Upon receiving the 360° video stream, the user views the real-time stream through a Web platform using his HMD and controls the UAV remotely using the right-side HMD's controller [as shown in Fig. 2(a)]. The HMD sends these commands to the UAV edge application. The user can control the FoV as full 6-DoF, using head and body movements [Fig. 2(c)], or using the HMD's left-side controller, [Fig. 2(b)]. Furthermore, with body movements, the user can control the altitude of the UAV, which lowers or increases its altitude if the user crouches or stretches, respectively, as shown in Fig. 3.

Edge Computing: Edge computing is considered to reduce the latency and alleviates the processing burden from the UAV and HMD. The latter would be achieved by offloading heavy tasks. Two edge applications are deployed near the UAV. These applications are, namely, the streaming module that relays video data and the control and monitoring module that transmits IoT and UAV commands data, as shown in Fig. 1.

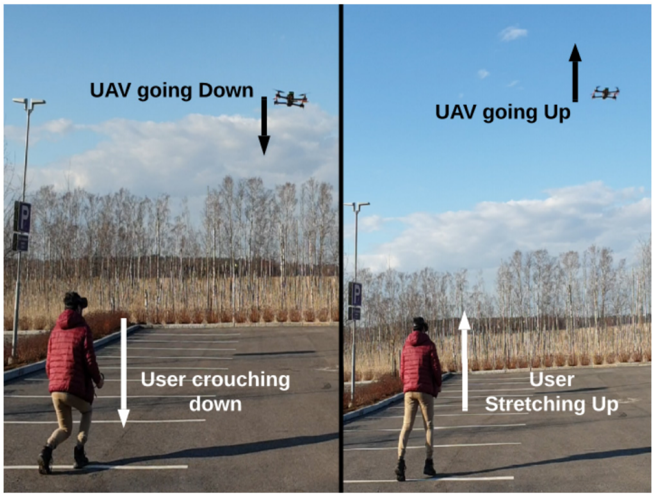


Fig. 3. 6-DoF camera control and UAV control based on body movements.

Streaming Module: The first edge application deployed near the HMD is the streaming module. This application is composed of an RTMP server and a WebRTC proxy allowing to transmit the real-time video stream to the Web application with the lowest latency possible. The streaming module converts the RTMP stream into WebRTC to reduce video latency. We have chosen WebRTC since it is well known for its ultralow latency and Web support [43]; hence, it provides 360° video to any device able to access the Web.

Web Server: The Web server serves the WebVR application. It is the interface to the user to view the information status of the UAV and 360° video stream through an HTML5 video player [68] adapted to play 360° video. It manages the WebSocket stream video from the WebRTC server and synchronizes different video inputs (UAV video streams) with the outputs (video players who are requesting a given stream). The 360° video can be viewed by any device able to access a Web browser. The choice of WebVR was mainly due to allowing an immersive view to any device that has access to a Web browser starting from a simple card box to a HMD.

Control and Monitoring Module: It is composed of a message queuing telemetry transport (MQTT) broker in charge of two functions: 1) forwarding the user control commands from the Web application to the flight controller module via one of the most popular communication protocols for UAVs, namely, the micro air vehicle link (MAVLink) protocol [69] and 2) updating the user about the censorial information of the UAV, such as altitude, latitude, longitude, and speed, as well as LTE and 5G-relevant information from the dongle that is connected to the UAV. This information is integrated within the 360° immersive view of the HMD. It is visualized by clicking on virtual elements within the immersive view, as shown in Fig. 4.

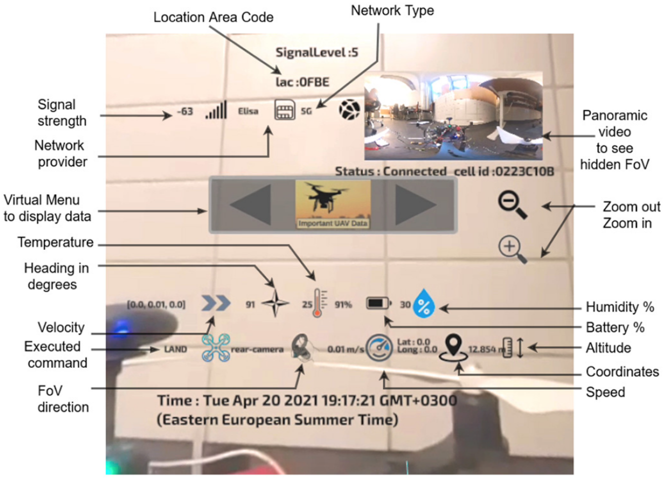


Fig. 4. Displaying the IoT sensed data in the immersive view.

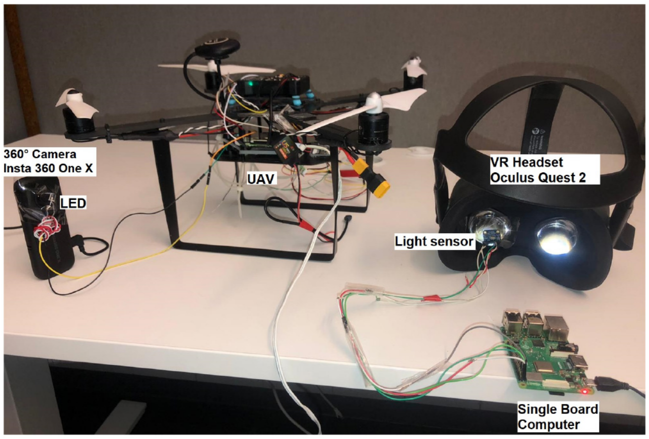


Fig. 5. Testbed's HW components.

TABLE II
TESTBED'S PARAMETERS AND VALUES

Parameter	Value
Wifi (upload/download)	100Mbps/200Mbps
4G (upload/download)	20Mbps/20Mbps
5G (upload/download)	50Mbps/200Mbps
Server CPU	8 cors @ 2.5GHz
Server memory	16Gb
Distance UAV to Server	200m
Distance VR HMD to Server	100m
UAV flight speed during tests	3Km/h

TABLE III
NOTATIONS USED IN THIS ARTICLE

Parameter	Description
$G2G$	Glass-to-Glass latency
GRE	Glass-to-Reaction-to-Execution latency
T_{LED}	The LED blink time at the UAV
T_{LS}	The time a light sensor detects the LED blink from the HMD lenses
T_{UAV_U}	The time a user command is received at the UAV
T_{UAV_S}	The time a light sensor command is received at the UAV
HRL	The human reaction latency (visual stimuli delay)
CTL	The command transmission latency
SRL	The sensor reaction latency

IV. PERFORMANCE ANALYSIS

This section provides a deep analysis of the testbed performance. The performance is analyzed in terms of GRE latency, G2G latency, human reaction latency (HRL), and command transmission latency (CTL). It is worth noting that the tests were not simulated but achieved in real using a real-life implementation. It is important to recall that for efficiently controlling a remote device/UAV, very low video transmission latency is essential. In this article, glass-to-reaction-to-execution (GRE) latency is defined as the time between the moment a motion or an event has been captured by the UAV's 360° camera to the moment a user's reaction to this event has been received and executed by the UAV. GRE is essentially important to determine how the system reacts to predicted events (e.g., approaching an obstacle such as a wall). GRE comprises three delays, namely: 1) glass-to-glass (G2G) latency, 2) HRL, and 3) CTL. These different delays were measured following the detailed architecture in Fig. 7.

This architecture was carried out using the following HW configuration shown in Fig. 5. It is composed of an Oculus Quest 2 as a HMD, a UAV equipped with a flight controller and a SBC, another SBC for G2G latency measurement and an Insta 360 One X as a 360° video camera.

We used a local wireless network as well as the 4G and 5G networks of a Finnish telecom operator, the characteristics of which are summarized in Table II. Though the speeds of 5G show remarkable downfall (as much as 7.5 times) when the receiving devices are not in direct line of sight of the antennas, this remains one of the biggest challenges of 5G [70].

The term G2G is defined in the literature as the delay between the moment an event is captured by a camera, for instance, a 360° camera, till the event is projected on the display (HMD display). The HRL refers to the delay a user takes to perceive a visual event and react to it. Finally, the CTL is the time a user command takes to reach and be executed by the UAV. Table III summarizes the parameters used in this article. Fig. 6 illustrates the analyzed delays.

A. Metrics Measurement

Several techniques have been proposed to measure G2G latency. SW-based techniques encode timestamps within the stream frames and retrieve them at the receiver. G2G is then calculated as the difference between the local system time and the decoded timestamp. Timestamps based on EAN-8 barcodes are the most commonly used, compared to numbers and characters, since they are accurate and easy to decode [71], [72]. However, these techniques are compute-intensive and may strain both sender and receiver resources to encode/decode the timestamps. On the other hand, HW-based techniques [73] do not involve the system resources, but instead use external tools and devices to measure G2G delay. MacCormick [74] and Hill et al. [75] measured the G2G delay using a camera

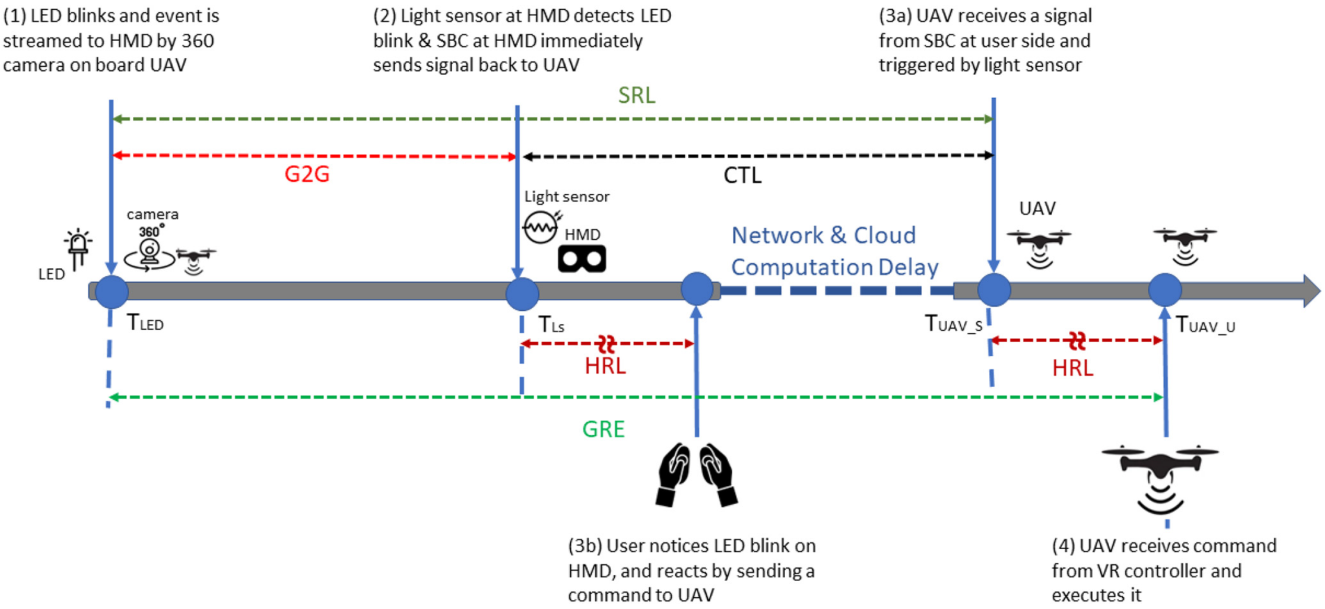

(1) LED blinks and event is streamed to HMD by 360 camera on board UAV
(2) Light sensor at HMD detects LED blink & SBC at HMD immediately sends signal back to UAV
(3a) UAV receives a signal from SBC at user side and triggered by light sensor
SRL
G2G
CTL
camera
360°
LED
Light sensor
HMD
Network & Cloud Computation Delay
UAV
TLED
TLs
TUAV_S
TUAV_U
HRL
HRL
GRE
(3b) User notices LED blink on HMD, and reacts by sending a command to UAV
(4) UAV receives command from VR controller and executes it

Fig. 6. Different delays analyzed in this article.

that films a clock on a computer screen, and its stream in a second screen whereby the difference noticed in the clock between both screens represented the G2G delay. However, the system required manual intervention to compare the clock's images, which does not allow to retrieve many samples of latency results and is thus not highly accurate. Gruen et al. [76] used an almost similar method to the previous one and measured visual latency on video for AR devices with an HW instrumentation-based measurement method. However, their method requires comparing both the event source and the screen of the HMD manually. Xu et al. [77] compared the G2G delay of video conferencing tools, such as Microsoft Skype and Google+ using an analog method to Hill et al. [75]. In their method, the clock's comparison from pictures was not retrieved manually but automatically by using an optical character recognition (OCR) SW. However, OCR cannot be used in panoramic frames such as in 360° video since images are stretched and therefore clock' numbers may not be recognized. In this article, an HW-based technique inspired from [78] and applied to the VR application on HMD is used to measure G2G, GRE, HRL, and CTL.

The method is applied to a head-mounted device (HMD), following the architecture in Fig. 7, and as shown in Fig. 5. It consists of triggering a light source, light-emitting diode (LED), using the SBC that is attached to the UAV, in front of a camera's FoV at T_{LED}. The LED's flashing is captured by the 360° camera and streamed to the HMD display. The light sensor that is connected to the HMD captures the blinks at T_{LS}, and triggers its SBC. The G2G delay is then calculated as the difference between T_{LS} and T_{LED}. Effectively, knowing that both the SBC of the UAV and the one equipped with a light sensor are connected through a wire and that the UAV's SBC notifies the light sensor's SBC when the LED is triggered, this allows us to measure the G2G delay at this latter SBC as follows:

$$G2G = T_{LS} - T_{\text{LED}}. \tag{1}$$

Furthermore, based on the G2G latency measurement tool, to measure GRE latency, the SBC on the UAV, on which a LED is attached is used. This SBC blinks the LED at a constant frequency and is placed in front of the camera to simulate an event, such as an approaching object or obstacle. The user then recognizes this object/obstacle through the HMD display and reacts to this event by pressing a button of the HMD controller (e.g., to turn away from that object). This command reaches the UAV at T_{UAV_U} to be executed then. The GRE is measured then as the delay between the moment the SBC blinks the light-emitting diode (LED), T_{LED}, and the moment the user command is received by the UAV T_{UAV_U}. Simultaneously, since the user wears the HMD with a light sensor, placed on one of the HMD lenses, the same setup can be used to measure the light sensor's detection latency which represents the CTL comprised with the G2G delay (i.e., as illustrated in Fig. 6). Once the light sensor detects the light at T_{LS}, the SBC on which it is attached reacts and sends a command to the UAV. This command reaches the UAV at T_{UAV_S}, and the sensor reaction latency (SRL) can be computed at the SBC of the UAV as follows:

$$\text{SRL} = T_{\text{UAV}\_S} - T_{\text{LED}}. \tag{2}$$

This latency can be also theoretically expressed as follows:

$$\text{SRL} = G2G + \text{CTL} = \text{GRE} - \text{HRL}. \tag{3}$$

CTL represents the commands' E2E transmission latency, excluding the latency needed for the processing and the rendering of the video, as shown in Fig. 7. Starting from when the user triggers the HMD's right controller until the reception of the command by the UAV, this latency consists of two main delays since the transmission involves two protocols: one

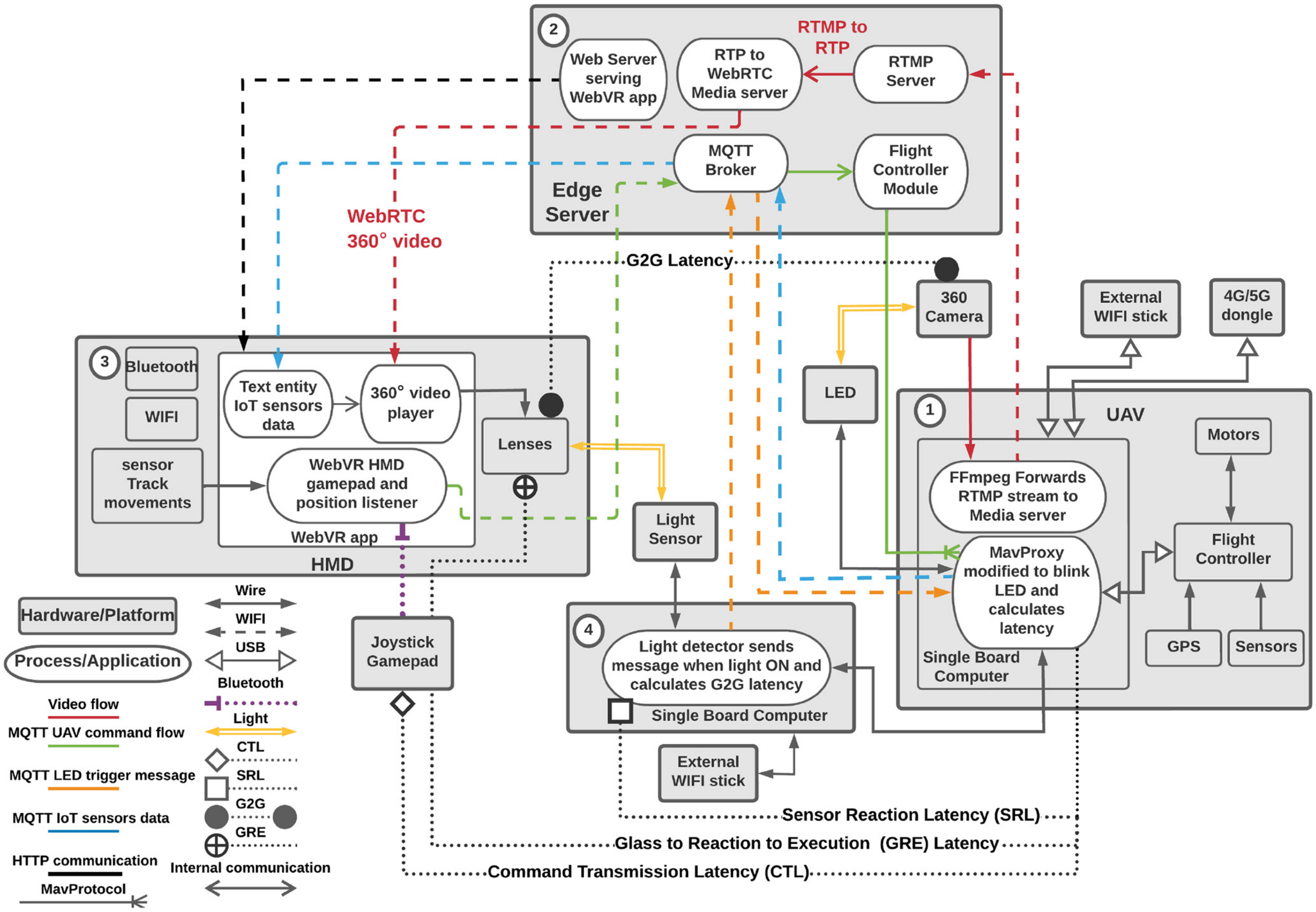


Fig. 7. Testbed's HW and SW components for VR-based UAV control and the measurement of the different considered delays.

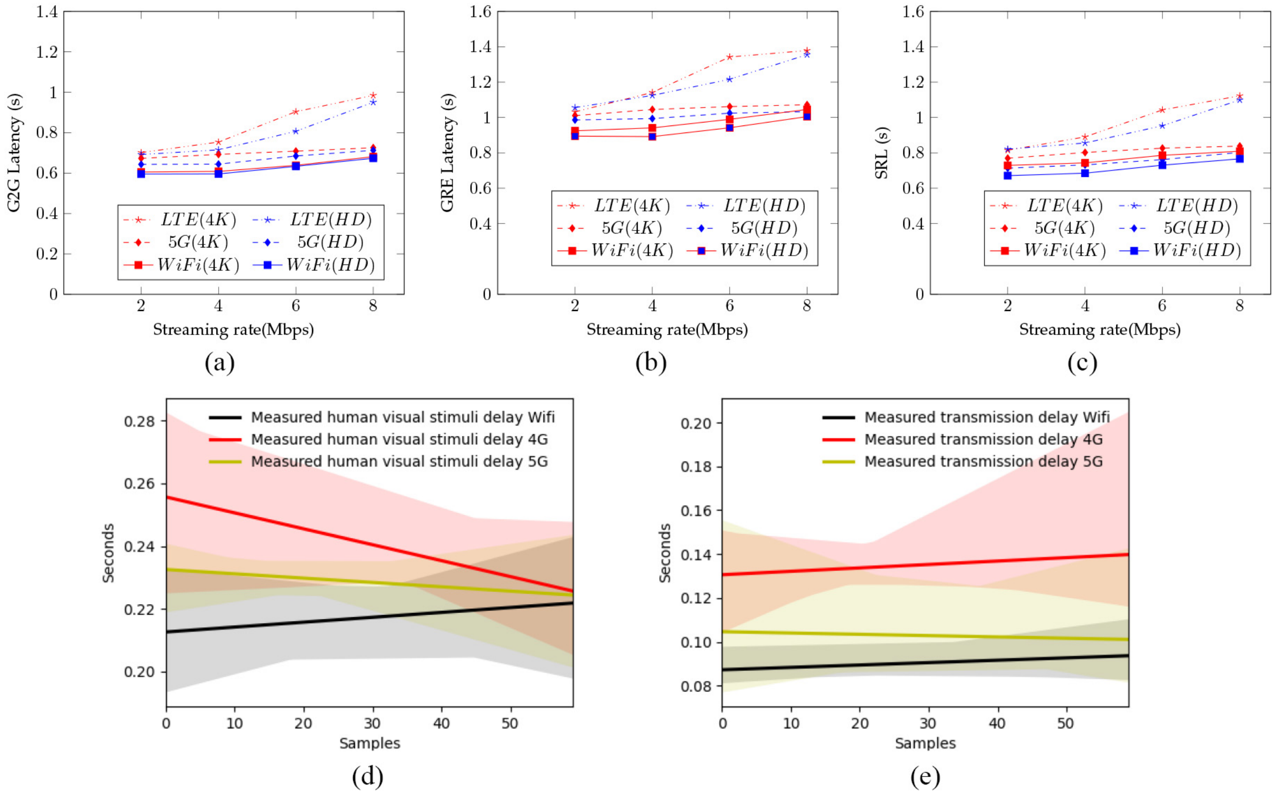
from the user to the edge server which is based on the MQTT protocol and the second from the edge server to the UAV that is based on the MAVLink protocol. For the MQTT protocol, we start a timer at the user when he/she sends a command to the MQTT broker at the edge. Effectively, we configured the subscriber (i.e., flight controller module) to echo a message to the client once it receives a command. The time difference between when the command is sent and when the echo is received represents the round-trip latency of the MQTT communication. Therefore, we consider the MQTT protocol latency as the round-trip latency divided by two. The same is done for the MAVLink protocol latency, as we exploit a Ping library implemented within the telemetry messages and calculate the round-trip latency.

B. Results Analysis

This section presents an analysis of the results obtained through experiments on the system. While the HMD uses WiFi to access services running at the edge and control the remote UAV, the UAV connects to the edge server through WiFi, LTE, or 5G. The 360° camera streams at 30 FPS and encodes video using the H264 codec [79]. Fig. 8 shows the GRE latency for different streaming bit rate (Mb/s), for different access networks of the UAV (i.e., WiFi, LTE, and 5G), and for two different video qualities (i.e., HD – 1280 × 720 – or 4k – 3840 × 1920). Within the same setup, two scenarios are considered, namely, with and without HRL.

Fig. 8(a) plots the measured G2G latency. The results demonstrate that this G2G delay increases as per an increase in the constant bit rate (CBR) encoding. This is intuitively due to network bandwidth limitations and lower throughput when increasing the streaming CBR. Moreover, it is clear that the increase is more noticeable in case of 4G, especially for 4K 360° videos. High network latency and bandwidth limitations in 4G are the main factors beneath the increase in the G2G latency. On the other hand, 5G shows very good results compared to 4G, since the G2G latency obtained in case of 5G is almost similar to the G2G latency experienced when the UAV is connected through a dedicated WiFi connection.

Fig. 8(b) and (c) shows the GRE latency and the SRL, respectively. Both metrics are measured simultaneously, once the LED blink is displayed on the HMD. Both the user and the light sensor react to this blinking. The user reacts by pressing the controller button, whereas the light sensor reacts by sending a command to the UAV. Naturally, the sensor reacts faster than a human being. From these figures, we also observe that the GRE and sensor reaction latency (SRL) increase, in the same fashion as the G2G latency, when the streaming bitrate is increased. This is trivial as both latencies include the G2G delay.



(a) (b) (c)

(d) (e)

Fig. 8. Measured latencies. (a) G2G latency. (b) GRE latency. (c) SRL. (d) Measured HRL. (e) Measured CTL (CTL).

Overall, it is noticeable that the average GRE is 900 ms [Fig. 8(b)]. This latency represents the overall E2E round-trip latency of the system from the moment an action occurs or an event is detected and streamed by the camera till the execution of the user's command at the UAV. As illustrated in Fig. 6, this latency comprises the G2G latency, the HRL latency (200–400 ms) and the CTL latency.

It is worth noting that each sample in the graphs shown in Fig. 8(a)–(c) represents an average of the results obtained from 40 iterations of the experiment described above. Iterations were carried out by different people (i.e., from the research team) and that is in order to minimize the impact of the individual HRL on the overall analysis. Indeed, the HRL, also known as the visual stimuli, differs from one individual to another, but it is known to be perceived after approximately 200 ms from stimulus [80]. We have measured this HRL latency by having different persons react to the LED blinking without wearing the headset. The visual stimuli delay is then the delay of an individual in reacting to a LED blink, without wearing a headset, subtracting from it the command transmission delay.

As shown in Fig. 6, an offset delay is what differentiates the sensor reaction and GRE delays. This offset maps unto HRL. SRL consists of the G2G latency and the CTL. Fig. 8(d) and (e) illustrates the measured HRL and the CTL, respectively. The HRL is independent of network delays and depends only on each person's visual reaction delays. Therefore, we can see that this delay tends to converge toward a constant value which is around 220 ms. For the measured command transmission delay, 5G shows almost identical delays to the WiFi network as a mean of 103 ms for 5G and 88 ms for WiFi. Whereas 4G tends to have a higher delay, namely, an average of 138 ms. This is mostly due to network latency and bandwidth limitations of 4G compared to WiFi and 5G.

Furthermore, to assess the video quality, we measured an objective and a subjective quality assessment metric, namely, the view-port peak signal to noise ratio (VP-PSNR) and VMAF developed by Netflix [36] based on the opensource library FFmpeg360 [81]. VMAF is a full reference (FR) metric that combines multiple secondary metrics using machine learning to offer a good prediction of subjective video quality (human perception) on a scale of 0 to 100. It was designed first to assess 2-D video quality, but its compatibility to work with 360 VR content without any adaptation is validated by [82]. The VP-PSNR is an objective video quality metric used to measure the distortion introduced by encoding at video transmission. To measure the VP-PSNR and VMAF values, we pursued the following steps.

Step 1: Record a 360° equirectangular projection (ERP) video at 4K (3840 by 1920) resolution at 30 FPS using a 360° camera.

Step 2: Generate the view-rendered video from the reference one by applying a filter from the FFmpeg360 library at this latter. The filter allows us to replicate the user's view at a given orientation position of pitch, yaw, and roll that was 0°, 0°, and 90° in our

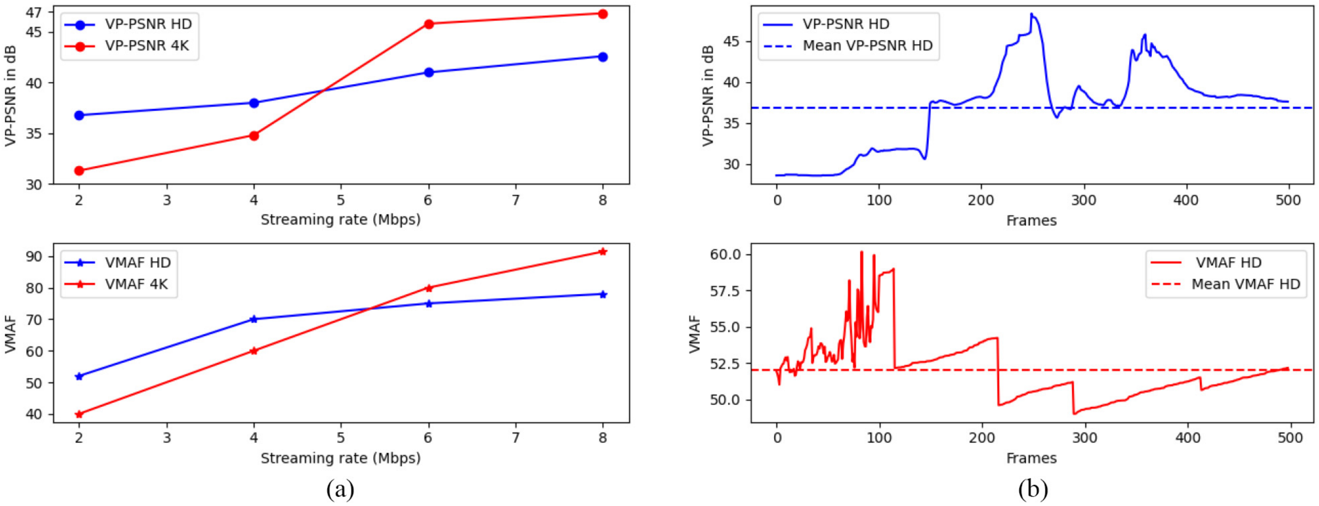


Fig. 9. Video quality evaluation in terms of VP-PSNR and VMAF. (a) VP-PSNR and VMAF values of a 360° video streamed over WiFi at different streaming rates. (b) VP-PSNR and VMAF values of different frames of a 360° HD video streamed over WiFi at a streaming rate of 2 Mb/s.

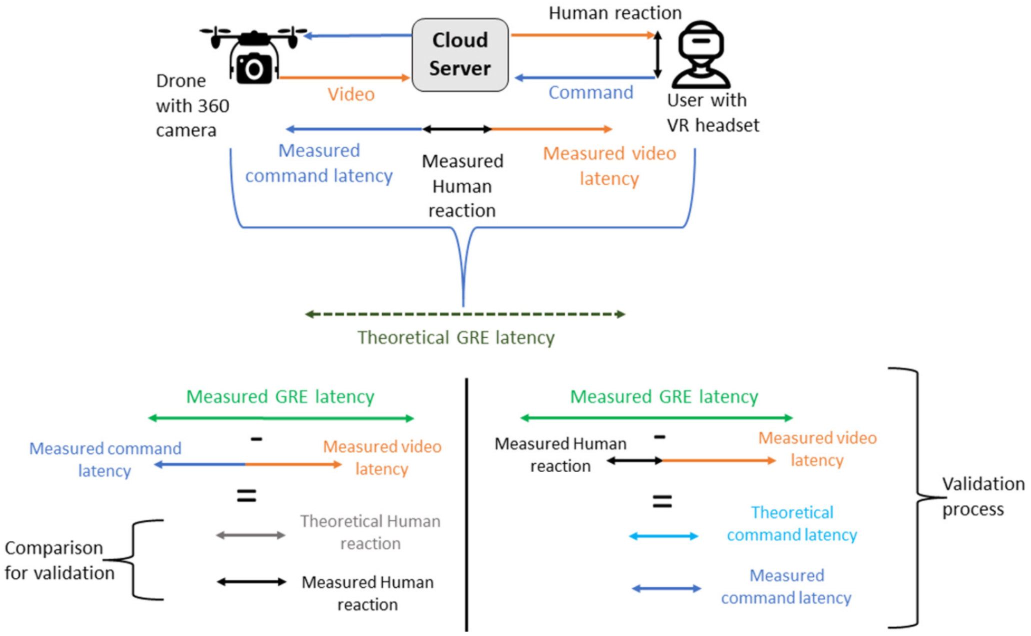


Fig. 10. Results' validation process.

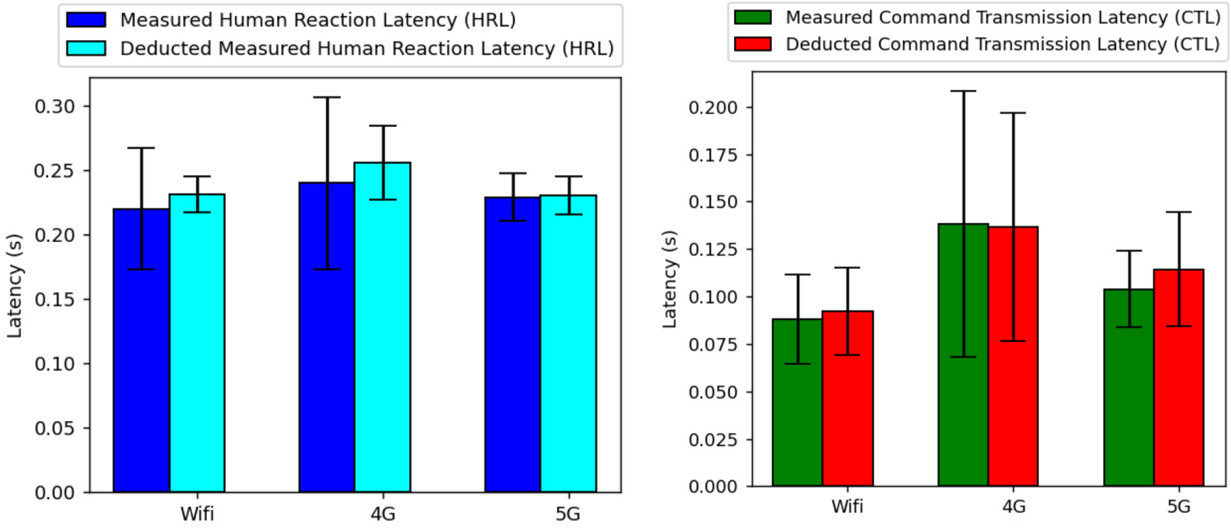
case. The video is referred to as a reference-view video.

Step 3: Stream this reference video through the Internet and our streaming system and record it at the receiving HMD that is viewing at 0°, 0°, 90° angle. We call this recorded video the user-view video.

Step 4: Make the visual quality comparison between the original reference-view and the user-view videos by applying PSNR and VMAF filters provided by FFmpeg.

Step 5: Repeat steps 3 and 4 by changing the streaming rate as well as all of the steps for an HD reference video.

The VP-PSNR and VMAF measurements were done considering different streaming rates as shown in Fig. 9(a). The values represent the mean values of the VP-PSNR and VMAF results for each stream rate of a 500 frames video stream for both 4K and HD quality streamed over WiFi. Fig. 9(b) plots the variation of the VP-PSNR and VMAF values of each frame of the 500 frames of a 360° HD video streamed at a rate of 2Mb/s. From Fig. 9(a), we observe that the values of VMAF and VP-PSNR increase as per the increase in the streaming rate and that is for both HD and 4k video streams. This increase is justified by the fact that the distortion rate is less important when the streaming rate is higher. We also notice that 4K videos get more distorted than HD videos when streamed at low rates, which is due to 4K videos containing more data than HD videos. Overall, the quality assessment of our received streams was satisfactory, since the lowest VMAF value at the lowest streaming rate, 2 Mb/s, is 40 and 50 for both 4k and



(a)

(b)

Fig. 11. Latency validation. (a) Validation of HRL. (b) Validation of CTL.

HD, respectively. In contrast, it reaches 78 and 90 when the streaming rate is at 8 Mb/s. We also observe satisfactory values for VP-PSNR and that is in case of both 4k and HD videos and for all the considered streaming rates. Accordingly, even at low stream rates, our video quality remains acceptable.

C. Results Validation

To validate the results provided in Fig. 8, both CTL and HRL can be calculated as follows:

$$\mathrm{HRL} = \mathrm{GRE} - \mathrm{SRL} \tag{4}$$

$$\mathrm{CTL} = \mathrm{SRL} - G2G. \tag{5}$$

The measured GRE was validated by measuring the delays composing it and comparing them against deduced ones, as shown in Fig. 10. The HRL, CTL, and G2G latencies were measured. The sum of these delays represents the theoretical GRE latency. Furthermore, the GRE latency was measured, which was validated by comparing it against the theoretical one.

The measured CTL and HRL results, shown in Fig. 8(d) and (e), are compared against the deduced results, as shown in Fig. 11. We can see that the human reaction delay, shown in Fig. 11(a), is independent of the network and is therefore almost steady for all network types. Moreover, the error between the deduced results and the measured ones is small as of 11.8 ms for WiFi experiments, 15.26 ms for 4G, and 11.59 ms for 5G. These delays are between 200 and 300 ms, which is similar to the results known in the state of the art [80].

Thereafter, we compared the measured and deduced delays for the CTL, as shown in Fig. 11 (b). As we can see, this delay is network-dependent since delays are higher when using 4G, then lower when using 5G, and even lower for WiFi. The error between the deduced and measured values was very small as 4.37 ms for WiFi, 10.39 ms for 5G, and 14.73 ms for 4G. These small errors between the theoretically deduced and measured delays prove the validity of our measurement method.

V. Conclusion

In this article, we proposed a real-life testbed, along with its management architecture, for VR-based remote control of UAVs, assisted by several IoT sensors. In the experiments, UAVs were reachable through a WiFi network, a 4G or a 5G cellular system. In the evaluation, several delays were defined and a methodology for their measurements was proposed. The obtained results were promising and proved the efficiency of the proposed VR-based UAV remote control architecture and the delay measurement method. The errors between the measured and deduced delays were very small, which validates the proposed measurement method. While the obtained results were encouraging, there is still room for improvement to minimize the 360° video streaming delays, such as the use of a camera rig instead of a 360° camera, and the application of machine learning techniques to predict the FoV of a user and to ultimately stream the watched FoV with high quality and stream the nonwatched FoV with lower quality to reduce the overall latency. Furthermore, to develop a full picture of VR-based UAV remote controlling, this study suggests customizing a design that considers the high requirements of immersive services and characteristics of UAVs. This design is needed to give flexibility to fully use the recently developed technologies. Recent development in communication protocols such as WebRTC coupled with transport protocols such as quick UDP Internet connections (QUICs) would optimize the services and reduce further the latency.

References

[1] S. Nalbandian, "A survey on Internet of Things: Applications and challenges," in *Proc. Int. Congr. Technol. Commun. Knowl. (ICTCK)*, 2015, pp. 165–169.

[2] S. Nižetić, P. Šolić, D. L.-D.-I. González-de-Artaza, and L. Patrono, "Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future," *J. Clean. Prod.*, vol. 274, Nov. 2020, Art. no. 122877.

[3] E. Park, Y. Cho, J. Han, and S. J. Kwon, "Comprehensive approaches to user acceptance of Internet of Things in a smart home environment," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2342–2350, Dec. 2017.

[4] W.-L. Chen et al., "AgriTalk: IoT for precision soil farming of turmeric cultivation," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5209–5223, Jun. 2019.

[5] L. Xu, R. Collier, and G. M. P. O'Hare, "A survey of clustering techniques in WSNs and consideration of the challenges of applying such to 5G IoT scenarios," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1229–1249, Oct. 2017.

[6] L. Chettri and R. Bera, "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, Jan. 2020.

[7] T. Taleb et al., "Towards supporting XR services: Architecture and enablers," *IEEE Internet Things J.*, early access, 2022, doi: 10.1109/JIOT.2022.3222103.

[8] T. Theodoropoulos et al., "Cloud-based XR services: A survey on relevant challenges and enabling technologies," *J. Netw. Netw. Appl.*, vol. 2, no. 1, pp. 1–22, Feb. 2022.

[9] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based Internet of Things services: Comprehensive survey and future perspectives," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 899–922, Dec. 2016.

[10] N. H. Motlagh, M. Bagaa, and T. Taleb, "Energy and delay aware task assignment mechanism for UAV-based IoT platform," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6523–6536, Aug. 2019.

[11] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.

[12] R. Li, "Towards a new Internet for the year 2030 and beyond," in *Proc. 3rd Annu. ITU IMT/5G Workshop Demo Day*, 2018, pp. 1–21.

[13] R. Li, K. Makhijani, and L. Dong, "New IP: A data packet framework to evolve the Internet," in *Proc. IEEE HPRS Workshop New IP*, May 2020, pp. 1–8.

[14] C. Qiu, X. Wang, H. Yao, J. Du, F. R. Yu, and S. Guo, "Networking integrated cloud–edge–end in IoT: A blockchain-assisted collective Q-learning approach," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12694–12704, Aug. 2021.

[15] J. Chakareski, S. Naqvi, N. Mastronarde, J. Xu, F. Afghah, and A. Razi, "An energy efficient framework for UAV-assisted millimeter wave 5G heterogeneous cellular networks," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 1, pp. 37–44, Mar. 2019.

[16] A. Samanta, Y. Li, and F. Esposito, "Battle of microservices: Towards latency-optimal heuristic scheduling for edge computing," in *Proc. IEEE Conf. Netw. Softw. (NetSoft)*, Jun. 2019, pp. 223–227.

[17] K. Samdanis and T. Taleb, "The road beyond 5G: A vision and insight of the key technologies," *IEEE Netw.*, vol. 34, no. 2, pp. 135–141, Mar./Apr. 2020.

[18] J. Chakareski, "VR/AR immersive communication: Caching, edge computing, and transmission trade-offs," in *Proc. Workshop Virtual Reality Augmented Reality Netw.*, 2017, pp. 36–41.

[19] X. Xia et al., "A 6-DOF telexistence drone controlled by a head mounted display," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces (VR)*, 2019, pp. 1241–1242.

[20] G. Lan et al., "Development of UAV based virtual reality systems," in *Proc. IEEE Int. Conf. Multisens. Fusion Integr. Intell. Syst. (MFI)*, 2016, pp. 481–486.

[21] J. van der Hooft, T. Wauters, F. De Turck, C. Timmerer, and H. Hellwagner, "Towards 6DoF HTTP adaptive streaming through point cloud compression," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 2405–2413.

[22] F. Qian, B. Han, J. Pair, and V. Gopalakrishnan, "Toward practical volumetric video streaming on commodity smartphones," in *Proc. 20th Int. Workshop Mobile Comput. Syst. Appl.*, 2019, 2019, pp. 135–140.

[23] Z. Liu et al., "Point cloud video streaming: Challenges and solutions," *IEEE Netw.*, vol. 35, no. 5, pp. 202–209, Sep./Oct. 2021, doi: 10.1109/MNET.101.2000364.

[24] J. Park, P. A. Chou, and J.-N. Hwang, "Rate-utility optimized streaming of volumetric media for augmented reality," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 149–162, Mar. 2019.

[25] M. Hosseini and V. Swaminathan, "Adaptive 360 VR video streaming: Divide and conquer," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, 2016, pp. 107–110.

[26] X. Corbillon, A. Devlic, G. Simon, and J. Chakareski, "Optimal set of 360-degree videos for viewport-adaptive streaming," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 943–951.

[27] J. Chakareski, R. Aksu, X. Corbillon, G. Simon, and V. Swaminathan, "Viewport-driven rate-distortion optimized 360° video streaming," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2018, pp. 1–7.

[28] X. Corbillon, G. Simon, A. Devlic, and J. Chakareski, "Viewport-adaptive navigable 360-degree video delivery," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2017, pp. 1–7.

[29] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the Internet," *IEEE Multimedia*, vol. 18, no. 4, pp. 62–67, Apr. 2011.

[30] X. Hu, W. Quan, T. Guo, Y. Liu, and L. Zhang, "Mobile edge assisted live streaming system for omnidirectional video," *Mobile Inf. Syst.*, vol. 2019, May 2019, Art. no. 8487372. [Online]. Available: https://www.hindawi.com/journals/misy/2019/8487372/

[31] T. T. Le, J. Jeong, and E.-S. Ryu, "Efficient transcoding and encryption for live 360 CCTV system," *Appl. Sci.*, vol. 9, no. 4, p. 760, 2019.

[32] R. Aksu, J. Chakareski, and V. Swaminathan, "Viewport-driven rate-distortion optimized scalable live 360° video network multicast," in *Proc. IEEE Int. Conf. Multimedia Expo Hot 3D Workshop*, 2018, pp. 1–6.

[33] J. Yi, M. R. Islam, S. Aggarwal, D. Koutsonikolas, Y. C. Hu, and Z. Yan, "An analysis of delay in live 360° video streaming systems," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 982–990.

[34] X. Liu, B. Han, F. Qian, and M. Varvello, "LIME: Understanding commercial 360° live video streaming services," in *Proc. 10th ACM Multimedia Syst. Conf.*, 2019, pp. 154–164.

[35] J. Yi, S. Luo, and Z. Yan, "A measurement study of YouTube 360° live video streaming," in *Proc. 29th ACM Workshop Netw. Oper. Syst. Support Digit. Audio Video*, 2019, pp. 49–54.

[36] Z. Li, C. Bampis, J. Novak, A. Aaron, K. Swanson, A. Moorthy, and J. D. Cock, *VMAF: The Journey Continues*, vol. 25, Netflix Technol. Blog, Los Gatos, CA, USA, 2018.

[37] ITUT FG-NET2030, "New services and capabilities for network 2030: Description, technical gap and performance target analysis," ITUT, Geneva, Switzerland, FG-NET2030 document NET2030-O-027, 2019.

[38] A. Clemm and T. Eckert, "High-precision latency forwarding over packet-programmable networks," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp. (NOMS)*, 2020, pp. 1–8.

[39] A. A. Malik, T. Masood, and A. Bilberg, "Virtual reality in manufacturing: Immersive and collaborative artificial-reality in design of human-robot workspace," *Int. J. Comput. Integr. Manuf.*, vol. 33, no. 1, pp. 22–37, 2020.

[40] R. Bevilacqua et al., "Non-immersive virtual reality for rehabilitation of the older people: A systematic review into efficacy and effectiveness," *J. Clin. Med.*, vol. 8, no. 11, p. 1882, 2019.

[41] A. Acemoglu et al., "Operating from a distance: Robotic vocal cord 5G telesurgery on a cadaver," *Ann. Internal Med.*, vol. 173, no. 11, pp. 940–941, 2020.

[42] E. Frontoni, J. Loncarski, R. Pierdicca, M. Bernardini, and M. Sasso, "Cyber physical systems for industry 4.0: Towards real time virtual reality in smart manufacturing," in *Proc. Int. Conf. Augmented Reality Virtual Reality Comput. Graph.*, 2018, pp. 422–434.

[43] C. Linn, S. Bender, J. Prosser, K. Schmitt, and D. Werth, "Virtual remote inspection—A new concept for virtual reality enhanced real-time maintenance," in *Proc. 23rd Int. Conf. Virtual Syst. Multimedia (VSMM)*, 2017, pp. 1–6.

[44] M. S. Feurstein, "Towards an integration of 360-degree video in higher education," in *Proc. 16th e-Learn. Conf. German Comput. Soc.*, 2018, pp. 1–12.

[45] O. Kim, Y. Pang, and J.-H. Kim, "The effectiveness of virtual reality for people with mild cognitive impairment or dementia: A meta-analysis," *BMC Psychiatry*, vol. 19, no. 1, p. 219, 2019.

[46] F. Hu et al., "Cyberphysical system with virtual reality for intelligent motion recognition and training," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 347–363, Feb. 2017.

[47] C.-C. Teng, N. Jensen, T. Smith, T. Forbush, K. Fletcher, and M. Hoover, "Interactive augmented live virtual reality streaming: A health care application," in *Proc. 2nd Int. Conf. Med. Health Inform.*, 2018, pp. 143–147.

[48] Q. Yang, H. Wang, M. Dohler, Y. Wen, and G. Xue, "Guest editorial multimedia communication in the Internet of Things," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 484–486, Apr. 2017.

[49] O. Erat, W. A. Isop, D. Kalkofen, and D. Schmalstieg, "Drone-augmented human vision: Exocentric control for drones exploring hidden areas," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 4, pp. 1437–1446, Apr. 2018.

[50] O. El-Marai, T. Taleb, and J. Song, "AR-based remote command and control service: Self-driving vehicles use case," *IEEE Netw.*, early access, Oct. 4, 2022, doi: 10.1109/MNET.119.2200058.

[51] O. El-Marai, T. Taleb, and J. Song, "Roads infrastructure digital twin: A step toward smarter cities realization," *IEEE Netw.*, vol. 35, no. 2, pp. 136–143, Mar./Apr. 2021.

[52] O. El-Marai and T. Taleb, "Smooth and low latency video streaming for autonomous cars during handover," *IEEE Netw. Mag.*, vol. 34, no. 6, pp. 302–309, 2020.

[53] T. Taleb, Z. Nadir, H. Flinck, and J. Song, "Extremely interactive and low-latency services in 5G and beyond mobile systems," *IEEE Commun. Stand. Mag.*, vol. 5, no. 2, pp. 114–119, Jun. 2021.

[54] Z. Nadir, T. Taleb, H. Flinck, O. Bouachir, and M. Bagaa, "Immersive services over 5G and beyond mobile systems," *IEEE Netw.*, vol. 35, no. 6, pp. 299–306, Nov./Dec. 2021.

[55] M. Aloqaily, O. Bouachir, A. Boukerche, and I. A. Ridhawi, "Design guidelines for blockchain-assisted 5G-UAV networks," *IEEE Netw.*, vol. 35, no. 1, pp. 64–71, Jan./Feb. 2021.

[56] Y. Bao, H. Wu, T. Zhang, A. A. Ramli, and X. Liu, "Shooting a moving target: Motion-prediction-based transmission for 360-degree videos," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2016, pp. 1161–1170.

[57] T. Braud, F. H. Bijarbooneh, D. Chatzopoulos, and P. Hui, "Future networking challenges: The case of mobile augmented reality," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2017, pp. 1796–1807.

[58] S. Oh and S. Hwang, "OMAF: Generalized signaling of region-wise packing for omnidirectional video," in *Proc. 118th MPEG Meeting ISO/IEC JTC1/SC29/WG11 MPEG2017/m40423*, 2017, pp. 1–4.

[59] K. K. Sreedhar, I. D. D. Curcio, A. Hourunranta, and M. Lepistö, "Immersive media experience with MPEG OMAF multi-viewpoints and overlays," in *Proc. 11th ACM Multimedia Syst. Conf.*, 2020, pp. 333–336.

[60] M. M. Hannuksela and Y.-K. Wang, "An overview of omnidirectional media format (OMAF)," *Proc. IEEE*, vol. 109, no. 9, pp. 1590–1606, Sep. 2021.

[61] R. Skupin, Y. Sanchez, D. Podborski, C. Hellge, and T. Schierl, "Viewport-dependent 360 degree video streaming based on the emerging omnidirectional media format (OMAF) standard," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2017, p. 4592.

[62] I. D. D. Curcio, K. Kammachi-Sreedhar, and S. Mate, "Multi-viewpoint and overlays in the MPEG OMAF standard," *ITU J. ICT Disc.*, vol. 3, no. 1, pp. 17–24, 2020.

[63] C. Hellge, "Live streaming of ultra-high resolution 360-degree video with MPEG-OMAF and HEVC tiles," Fraunhofer HHI, Berlin, Germany, White Paper, 2018.

[64] Y. S. de la Fuente, G. S. Bhullar, R. Skupin, C. Hellge, and T. Schierl, "Delay impact on MPEG OMAF's tile-based viewport-dependent 360° video streaming," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 18–28, Mar. 2019.

[65] B. Shneiderman, "Response time and display rate in human performance with computers," *ACM Comput. Surv.*, vol. 16, no. 3, pp. 265–285, 1984.

[66] L. Liu et al., "Cutting the cord: Designing a high-quality untethered VR system with low latency remote rendering," in *Proc. 16th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2018, pp. 68–80.

[67] J. Newmarch, "FFmpeg/Libav," in *Linux Sound Programming*. Berkeley, CA, USA: Apress, Jan. 2017, pp. 227–234, [Online]. Available: http://dx.doi.org/10.1007/978-1-4842-2496-0_12

[68] G. Zhu, F. Zhang, W. Zhu, and Y. Zheng, "HTML5 based media player for real-time video surveillance," in *Proc. 5th Int. Congr. Image Signal Process.*, 2012, pp. 245–248.

[69] A. Koubaa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, "Micro air vehicle link (MAVlink) in a nutshell: A survey," *IEEE Access*, vol. 7, pp. 87658–87680, 2019.

[70] R. Yadav and G. Kataria, "5G—Current developments and what to expect," *Int. J. Innov. Sci. Res. Technol.*, vol. 5, pp. 315–319, Jan. 2020.

[71] O. Boyaci, A. Forte, S. A. Baset, and H. Schulzrinne, "vDelay: A tool to measure capture-to-display latency and frame rate," in *Proc. 11th IEEE Int. Symp. Multimedia*, 2009, pp. 194–200.

[72] J. Jansen and D. C. A. Bulterman, "User-centric video delay measurements," in *Proc. 23rd ACM Workshop Netw. Oper. Syst. Support Digit. Audio Video*, 2013, pp. 37–42.

[73] K. Raaen and I. Kjellmo, "Measuring latency in virtual reality systems," in *Proc. Int. Conf. Entertainment Comput.*, 2015, pp. 457–462.

[74] J. MacCormick, "Video chat with multiple cameras," in *Proc. Conf. Comput. Supported Cooper. Work Companion*, 2013, pp. 195–198.

[75] R. Hill, C. Madden, A. van den Hengel, H. Detmold, and A. Dick, "Measuring latency for video surveillance systems," in *Proc. Digit. Image Comput. Techn. Appl.*, 2009, pp. 89–95.

[76] R. Gruen, E. Ofek, A. Steed, R. Gal, M. Sinclair, and M. Gonzalez-Franco, "Measuring system visual latency through cognitive latency on video see-through AR devices," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces (VR)*, 2020, pp. 791–799.

[77] Y. Xu, C. Yu, J. Li, and Y. Liu, "Video telephony for end-consumers: Measurement study of Google+, iChat, and Skype," in *Proc. Internet Meas. Conf.*, 2012, pp. 371–384.

[78] C. Bachhuber and E. Steinbach, "A system for high precision glass-to-glass delay measurements in video communication," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2016, pp. 2132–2136.

[79] H. Kalva, "The H.264 video coding standard," *IEEE MultiMedia*, vol. 13, no. 4, pp. 86–90, Oct.–Dec. 2006, doi: 10.1109/MMUL.2006.93.

[80] M. Gonzalez-Franco, A. I. Bellido, K. J. Blom, M. Slater, and A. Rodriguez-Fornells, "The neurological traces of look-alike avatars," *Front. Human Neurosci.*, vol. 10, p. 392, Aug. 2016.

[81] Y. Liu, C. Zhou, S. Wang, and M. Xiao, "FFmpeg360 for 360-degree videos: Edge-based transcoding, view rendering, and visual quality comparison: Poster," in *Proc. 4th ACM/IEEE Symp. Edge Comput.*, 2019, pp. 337–339.

[82] M. Orduna, C. Díaz, L. Muñoz, P. Pérez, I. Benito, and N. García, "Video multimethod assessment fusion (VMAF) on 360VR contents," *IEEE Trans. Consum. Electron.*, vol. 66, no. 1, pp. 22–31, Feb. 2020.

Tarik Taleb (Senior Member, IEEE) received the B.E. degree (with Distinction) in information engineering and the M.Sc. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 2001, 2003, and 2005, respectively.

He is currently a Professor with the Center of Wireless Communications, The University of Oulu, Oulu, Finland, where he is the Founder and the Director of the MOSA!C Lab. He was a Professor with the School of Electrical Engineering, Aalto University, Espoo, Finland, from October 2014 and December 2021. Prior to that, he was working as a Senior Researcher and a 3GPP Standards Expert with NEC Europe Ltd., Heidelberg, Germany. Before joining NEC and till March 2009, he worked as an Assistant Professor with the Graduate School of Information Sciences, Tohoku University in a lab fully funded by KDDI. He worked as a Research Fellow with the Intelligent Cosmos Research Institute, Sendai, from October 2005 to March 2006. His research interests lie in the field o telco cloud, network softwarization and network slicing, AI-based SW-defined security, immersive communications, mobile multimedia streaming, and next generation mobile networking.

Prof. Taleb is the recipient of the 2021 IEEE ComSoc Wireless Communications Technical Committee Recognition Award in December 2021, the 2017 IEEE ComSoc Communications Software Technical Achievement Award in December 2017 for his outstanding contributions to network softwarization. He is also the (co-) recipient of the 2017 IEEE Communications Society Fred W. Ellersick Prize in May 2017, and many other awards from Japan. Some of his research works have been also awarded best paper awards at prestigious IEEE-flagged conferences. He has been also directly engaged in the development and standardization of the Evolved Packet System as a member of 3GPP's System Architecture Working Group 2. He served for the IEEE Communications Society Standardization Program Development Board. He served as the General Chair for the 2019 edition of the IEEE Wireless Communications and Networking Conference held in Marrakech, Morocco. He was the Guest Editor-in-Chief of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS Series on Network Softwarization & Enablers. He served for the editorial board of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, *IEEE Wireless Communications Magazine*, IEEE JOURNAL ON INTERNET OF THINGS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE COMMUNICATIONS SURVEYS & TUTORIALS, and a number of Wiley journals. Till December 2016, he served as the Chair for the Wireless Communications Technical Committee, the largest in IEEE ComSoC.

Nassim Sehad received the Bachelor of Science degree in telecommunication and the master's Diploma degree in networks and telecommunication from the University of Sciences and Technology Houari Boumediene, Algiers, Algeria, in September 2018 and 2020, respectively. He is currently pursuing the Doctoral degree with the Department of Communications and Networking, Aalto University, Espoo, Finland.

His main research topics of interest are multisensory multimedia, IoT, cloud computing, and networks.

Zinelaabidine Nadir received the M.Sc. degree in computer science from Laghouat University, Laghouat, Algeria, in 2012. He is currently pursuing the Doctoral degree with Aalto University, Espoo, Finland.

His current research focuses on immersive services, multimedia streaming, and next generation mobile networking.

Jaeseung Song (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science from Sogang University, Seoul, South Korea, in 2000 and 2002, respectively, and the Ph.D. degree from the Department of Computing, Imperial College London, London, U.K., in 2013.

He is currently a Professor with Sejong University, Seoul. His research interests span the areas of beyond 5G and 6G, AI/ML-enabled network systems, software engineering, and networked systems and security, with focus on the design and engineering of reliable and intelligent IoT/M2M platforms.