IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 22, 2021, accepted August 3, 2021, date of publication September 3, 2021, date of current version September 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3110284

The Upstream Matters: Impact of Uplink Performance on YouTube 360° Live Video Streaming in LTE

LUIS ROBERTO JIMÉNEZ, (Graduate Student Member, IEEE), MARTA SOLERA, MATÍAS TORIL, SALVADOR LUNA-RAMÍREZ, AND JUAN L. BEJARANO-LUQUE

Instituto de Telecomunicación (TELMA), Universidad de Málaga, CEI Andalucía TECH, E.T.S. Ingeniería de Telecomunicación, 29010 Málaga, Spain

Corresponding author: Luis Roberto Jiménez (lrjp@ic.uma.es)

This work was supported in part by the Spanish Ministry of Science, Innovation and Universities under Grant RTI2018-099148-B-I00, and in part by the Presidency of the State Research Agency under Grant FPI/BES-2016-07631.

ABSTRACT Live video streaming services are gaining momentum as network and terminal capabilities improve. However, 360° live video streaming services pose new challenges due to its high bandwidth and computational requirements both on the user and service provider. In this paper, a study of the impact of the uplink of a cellular network on the performance of 360° live video streaming in YouTube is presented. Unlike previous works, the analysis focuses on the upstream between the video source and the server, not on the downstream between the server and viewers. To this end, a measurement campaign is conducted where a live video feed is transmitted and received through YouTube 360° platform in a pilot Long Term Evolution (LTE) system. During the tests, a large dataset of real traces is collected at different protocol layers, both in upstream and downstream, to check the correlation between TCP/IP metrics and key service performance indicators (e.g., video segment quality and end-to-end latency). Results show that uplink performance has a strong impact on the latency perceived by the user, which is critical for the considered live services.

INDEX TERMS YouTube, live video streaming, 360-degree video, latency, Quality of Experience, uplink.

I. INTRODUCTION

The latest advances in multimedia technology have caused the proliferation of advanced video applications [1]. By 2020, live video streaming outpaced the growth of other types of online video, thanks to the rise of entertainment applications, such as sports and cultural event broadcasting, video surveillance and teleconferencing for meetings and distance learning [2]. Besides, the lower cost of video cameras allows users to make live transmissions in increasingly complex and sophisticated immersive formats. A good example is 360° video, recorded with omnidirectional cameras and projected in 2D using mapping techniques, allowing users to freely change the direction of the content display using a mouse, a phone touch screen or virtual reality headsets.

Major service platforms, such as YouTube, Instagram and Facebook, already offer live 360° video streaming [3]. However, the distribution of these videos poses new challenges. Currently, the video client downloads the entire scene, even if the user terminal only displays a portion of the image (i.e., viewport). This results in inefficient use of network bandwidth by downloading data never used. A solution to improve network efficiency is adaptive 360° video streaming [4], [5]. On the client side, the future user display port is predicted and segments of that area of the image are requested with a higher quality than in the rest of the scene. On the server side, the original video stream received is stored and transcoded into segments of different qualities according to the set of possible viewports. Thus, the video download is optimized depending on network conditions and display prediction.

In parallel, technological advances have increased user expectations, forcing operators to change the way they manage their networks. To this end, network operators have moved from a network-centric approach, focused on network performance, to a user-centric approach, focused on user opinion [6]. In this context, understanding the factors that affect the delivery of services is key for network operators to provide an adequate Quality of Experience (QoE). This is

The associate editor coordinating the review of this manuscript and approving it for publication was Wenjie Feng.

especially true for cellular network operators about to deploy 5G systems, for which live immersive video broadcast is one of the main use cases [7].

To date, many works have published methods for estimating the QoE of adaptive video streaming services [8]–[20]. In practice, most cellular operators currently use passive parametric methods that rely on a few network-level metrics collected by probes in the core domain [21]–[27]. These methods have been adapted to also cover live video streaming services [28]. Yet, the above methods are only focused on downlink performance, which is the restricting link when mobile users download information from a server. In legacy cellular systems, most studies on video streaming focused on the downlink, because the bulk of traffic was generated by the download of large video files on demand. With recent advances in mobile handsets, users can now generate appealing live content with their terminals, which is uploaded in real time through the uplink of a cellular network to content servers. It is well-known that time-varying wireless channel conditions affect smooth data transmission [29]. These issues severely affect the uplink, since effective uplink data rates tend to be lower than downlink data rates in live systems due to diverse factors (e.g., simpler multi-antenna configurations, limited modulation and coding schemes, different parameterization of link adaptation algorithms, less agile scheduling, larger interference variability…) [30], [31]. Moreover, unlike the link between server and viewer, where dynamic adaptive streaming techniques are used, the link between the video source and server often uses non-adaptive online streaming protocols. Therefore, the study of factors of uplink performance in terms of bandwidth or latency is key to maximize end-user QoE of Live Video Streaming [31]. This is especially true for data intensive applications like 360° immersive video streaming. However, in a live mobile broadcast, the uplink used by the video source to upload content to the server in real time is a major contributor to service performance. Likewise, classical video streaming QoE models conceived for offline streaming do not take latency into account. However, in live streaming, users are also sensitive to the end-to-end video latency. To the authors' knowledge, no previous work has checked the impact of the uplink of a cellular radio access network on the latency performance in a live broadcast through a popular over-the-top (OTT) video streaming platform.

In this work, a study of the impact of the uplink of a cellular network on the performance of 360° live video streaming in YouTube is presented. It should be pointed out that 360° live video content currently available on YouTube is handled like a conventional video. Nonetheless, such an immersive video requires higher data rates, making the video distribution more challenging. Unlike previous works, the analysis is focused on the upstream between the video source and the server, and not on the downstream between the server and viewers. Generally, "downlink" refers to a radio link for transmitting signals from the base station to a mobile terminal, while the "uplink" refers to a radio link for the transmission of signals from the mobile terminal to the base station. To avoid confusion, we use the term "upstream" to refer to the bidirectional link (consisting of uplink and downlink) between video source and server, and the term "downstream" to refer to the bidirectional link (also comprising uplink and downlink) between video server and viewer. In the case considered here, when the user located on the upstream segment (from video source to server) generates the 360° live video streaming, the uplink is used for video ingest to the content server, using RTMP as streaming protocol. In contrast, in the downstream segment (from server to viewer), downlink is used for content delivery using a dynamic adaptive streaming protocol (e.g., DASH). These adaptation mechanisms make the downlink more robust against fluctuations of link conditions. In addition, uplink data rates tend to be lower than downlink data rates. All these issues make the upstream more prone to impairments due to poor uplink performance. In order to study the impact of the uplink (upstream segment), a measurement campaign is conducted where a live video feed is uploaded and received through YouTube 360° platform in a pilot Long Term Evolution (LTE) system. During the tests, a large dataset of real traces is collected at different protocol layers, both in the upstream and downstream. With this data, it is possible to check the correlation between network-level metrics in the uplink (e.g., average user throughput) and key service performance indicators (e.g., image quality, stalling ratio, and end-to-end latency) on a per-session basis. Such a piece of information can be used by network operators to derive simple parametric QoE models for this new service that also take the upstream into account. Note that any uplink or downlink throttling will affect end-to-end latency. Low downlink throughput increases latency downstream but does not affect video ingestion to YouTube (upstream). In contrast, low throughput levels on the uplink affect the whole system. Certainly, network-layer QoE models can be derived from both links (uplink/downlink), but only the uplink can model the behavior from the video source to the viewers (end-to-end). The rest of the paper is structured as follows. Section II reviews related work. Section III outlines the relevant LTE-specific aspects influencing end-to-end latency. Section IV explains the experimental methodology. Section V shows the analysis of measurements. Finally, section VI presents the conclusions of the work.

II. RELATED WORK

In the literature, several objective metrics have been proposed as key metrics to measure the QoE in video streaming services. The first client-side QoE models were designed for conventional video-on-demand streaming based on HTTP progressive download (HPD), where stalling is the most critical factor. In [32], a simple model for non-adaptive streaming estimates user experience directly from network-layer metrics, such as packet loss/jitter/delay and bandwidth, which is suitable for encrypted traffic. More advanced approaches estimate user experience from application-layer metrics. In [22], a QoE model for YouTube based on passive in-device measurements is proposed. In [8], [33], [34], initial reproduction delay, stalling frequency, and stalling duration are used to estimate the QoE of HPD. In [15], a QoE model for the conventional YouTube service based on estimating the client buffer level is presented. With the advent of adaptive video streaming services [14], modern approaches add new metrics, such as image quality and frequency of switches between quality levels for QoE estimation of encrypted video traffic [35], combined with stalling statistics [36], [17]. To circumvent the complexity of defining closed-form analytical models, the most sophisticated schemes use machine learning techniques to find complex dependencies between indicators in QoE estimation [37]–[42].

A few works have evaluated user experience in 360° video streaming services. Preliminary works collected subjective quality measurements on 360° video [43] to build visual quality assessment (VQA) models for 360° video [4], [44]–[46]. Later works highlight download bitrate as the leading cause affecting the QoE [47], [48]. More advanced studies add other factors (e.g., stallings, resolution, ...) [49]–[52] and derive their interactions with machine learning techniques for predicting QoE specifically for 360° videos [53]–[56]. In [57], a deep-learning scheme for maximizing the quality of video chunks delivered with latency constraints by small cell base stations working in the millimeter band is proposed for multi-user virtual reality 360° videos. The proposed scheme predicts user field of view, clusters users in base stations and schedules multicast transmission of future video chunks. The approach is validated through simulations. Other works are focused on the uplink bitrate management as the main element to improve the QoE, only taking into account resolution changes in the video segments sent to the content server due to the changing network conditions [58]–[62].

With the recent success of live video transmissions, there is a growing interest on latency issues [63]. Legacy television broadcasting systems achieve video latency values of a few seconds, mainly originated by video frame synchronizers in routing and switching devices [64]. In contrast, OTT video streaming platforms offer latency values ranging from 10 to 30 seconds [63]. Several strategies have been proposed to reduce latency. On the one hand, the average segment duration, configured by the service provider in the planning stage, has been shortened from the typical 10 seconds to counteract that most HTTP-based players store a fixed number of segments before starting playback for robustness [65]. In parallel, many current browsers include real-time streaming protocols, such as Real–Time Messaging Protocol (RTMP), low-latency HTTP live streaming (LL-HLS), Secure Reliable Transport (SRT) or Web Real-Time Communication (webRTC), to achieve video latencies below 1 second [66]. The combination of these protocols with content delivery networks can help to improve scalability while decreasing latency. At the same time, chunked-encoding can be used to divide a video segment into multiple shorter video chunks (typically, 1-15 video frames), which are then encoded, streamed and decoded in a pipelined fashion in chunked-transfer encoding [67]. For this purpose, standard media containers must be paired with encoder, content delivery network and client, so that the overall system provides low latency [68]. Alternative model predictive control schemes perform online bitrate adaptation to find the optimal trade-off between video quality, playback latency, video freeze, and skip [69]–[71].

Several methodologies have been proposed to measure video latency. The basic approach is to introduce timestamps into the video stream [72], [73]. At the receiver side, an application decodes the timestamp and estimates capture-to-display latency. For this purpose, the source and receiving end must be synchronized to generic Network Time Protocol (NTP) servers available on the Internet. In [74], glass-to-glass video delay is estimated from video round trip time. To this end, QR codes are displayed in front of the sender camera and the resulting image is transferred to the remote computer, where it is captured by a second camera in front of the display that detects the QR code and notifies it to the sender. Alternatively, the latest YouTube Live updates include latency between parameters that can be monitored by ''Stats for nerds'' feature [75].

To the authors' knowledge, no previous work has checked the influence of the uplink in a live broadcast with a popular over-the-top (OTT) video streaming platform when uploading the content through a cellular radio access network. Hence, the main contribution of this work is the analysis of the impact of LTE uplink on factors affecting the experience of end users in YouTube 360° live streaming service.

III. LATENCY OF LIVE VIDEO STREAMING SERVICE IN LTE

The new habits of users, who need to generate content at any time and place to publish it on social networks, make mobile networks the main technology for providing these services. In live streaming, users need to reduce the time gap between the moment the camera captures the event, and the user displays it on their screen (end-to-end latency). Many mechanisms are implemented today to reduce this time, but some residual value is still present.

Fig. 1 summarizes the different components contributing to end-to-end latency in a live videostreaming service when both source and target user are connected to a LTE system. Transport network refers to the transport system connecting the cellular system to the public Internet. Delay values associated to video processing at the end devices (camera and display, in red) correspond to a video profile with 2560 x 1440 pixel resolution at 30 fps and H.264 AVC codec. These terms are excluded from the analysis in the following sections, since the focus is on network-related aspects.

Fig. 2 shows the reference LTE architecture, consisting of two main subsystems [76]. The first one, Evolved Universal Mobile Telecommunications System Terrestrial Radio Access Network (E-UTRAN), is responsible for providing the connection between user equipment (UE) and core equipment through base stations known as evolved-Node B (eNB). eNB is the entity that manages radio resources

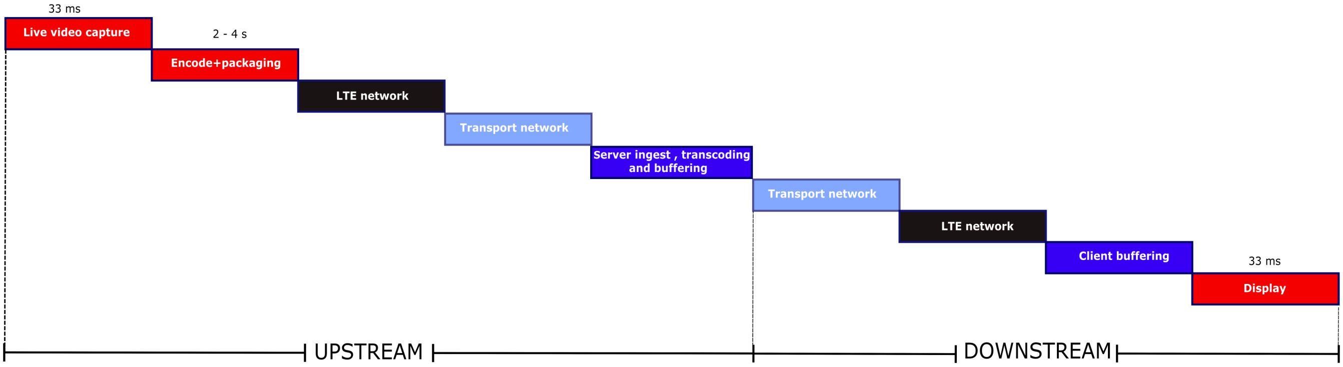


FIGURE 1. Processes contributing to end-to-end latency for a live video streaming service in LTE.

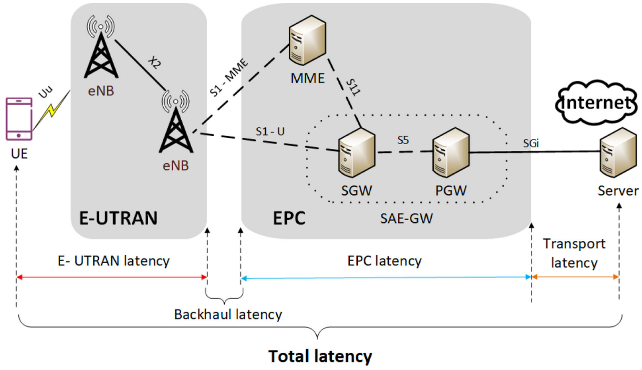


FIGURE 2. LTE reference architecture for video streaming services.

and is, therefore, in charge of admission control, mobility control, dynamic resource allocation, and interference control. The second subsystem is the core network, known as Evolved Packet Core (EPC), responsible for mobility management, access control, user connection management and interconnection with other networks. The EPC mainly consists of a control plane node, known as mobility management entity (MME), and two user plane nodes, called the Service Gateway (S-GW) and Packet Data Network Gateway (P-GW). MME is in charge of processing signaling information between users and the backbone. S-GW is the link between E-UTRAN and EPC at user level, while P-GW provides connectivity between the LTE system and external networks. All the above subsystems contribute to the overall latency of the LTE network.

In cellular networks, latency is divided into control plane and user plane latency. Control plane latency is defined as the time for the UE to transit from idle state to active state by establishing a Radio Resource Control (RRC) connection. In this work, it is assumed that the user is already active, so that the control plane latency is neglected. Then, user plane latency is defined as the one-way transmit time from a packet being available in the IP layer at the UE to the availability of this packet in the IP layer at the server. Table 1 shows typical latency values reported for the different segments in the cellular system [77], [78]. For illustrative purposes, Table 2 breaks down the different processes introducing latency in the air interface of E-UTRAN latency for a fixed Hybrid automatic repeat request (HARQ) retransmission probabilities of 10% [78], [79]. In practice, latency values might be larger due to radio link quality issues and capacity bottlenecks at the eNB.

TABLE 1. Typical latency values in LTE architecture [77], [78].

Step	Component	Typical value
1	E- UTRAN latency	5 ms
2	Backhaul latency	10 ms
3	EPC latency	15 - 85 ms
	Total one-way delay	**30 – 100 ms**

TABLE 2. Analysis of user-plane latency in the radio interface [78], [79].

Step	Component	Value (10% HARQ retx.probability)
1	UE processing delay	1 ms
2	Frame alignment	0.5 ms
3	Time Transmission Interval for UL data packet	1 ms
4	HARQ retransmission	0.1*5 ms
5	eNB processing delay (Uu – S1-U)	1 ms
	Total one-way delay	**4 ms**

IV. EXPERIMENTAL METHODOLOGY

Firstly, the platform where measurements are collected is outlined. Then, the analysis methodology is detailed.

A. MEASUREMENT PLATFORM

Fig. 3 shows a diagram of the platform used to automate measurement collection. It is made up of two modules: a broadcast module (upper left part of the diagram) responsible for the live broadcast of a 360° video on the YouTube Live platform, and a measurement module (bottom part of the diagram) responsible for deciphering, collecting and analyzing measurements. Both modules use a private (i.e., non-commercial) pilot LTE network as their access network.

The broadcast module consists of a Dell OptiPlex Personal Computer (PC, 790 MT model, Quad Core i7 3.4 GHz processor, 8 GB RAM, integrated Intel HD Graphics 2000 Dynamic video card, 500 GB hard drive) and a live broadcast camera (Samsung Gear 360° model) connected to the PC via a USB 3.0 interface. The Cyberlink Samsung Gear 360° Action software allows the creation and broadcast of the 360° live video on the YouTube platform from a computer [80] through an event previously configured in the broadcast manager (YouTube Studio) [81]. To connect to the LTE network,

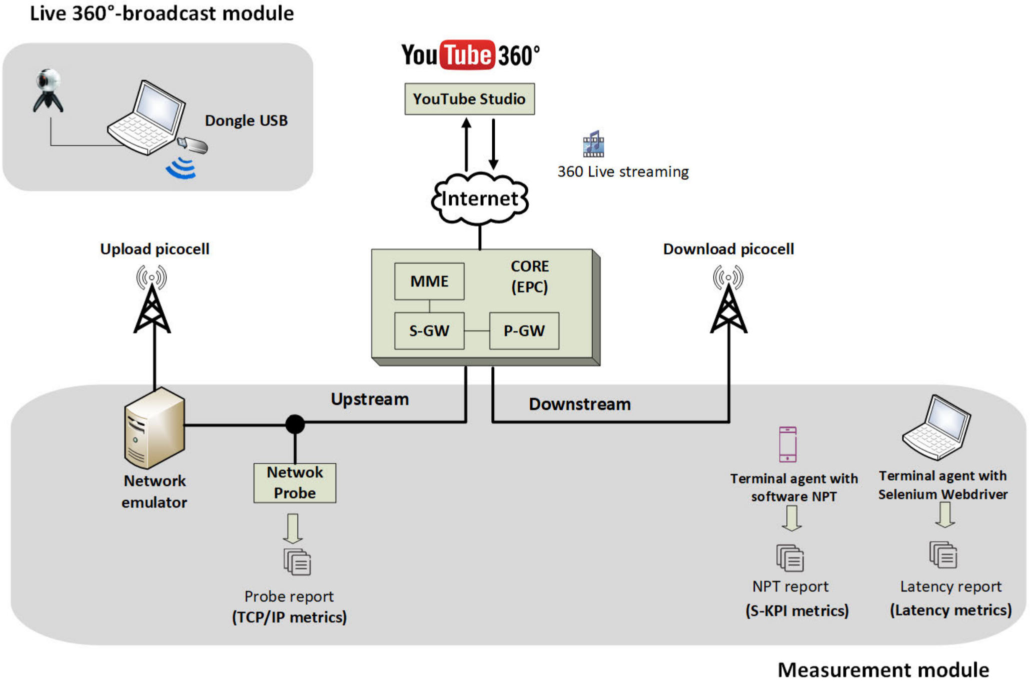

Live 360°-broadcast module
Dongle USB
YouTube 360°
YouTube Studio
360 Live streaming
Internet
Upload picocell
CORE (EPC)
MME
S-GW
P-GW
Download picocell
Upstream
Downstream
Network emulator
Netwok Probe
Probe report (TCP/IP metrics)
Terminal agent with software NPT
Terminal agent with Selenium Webdriver
NPT report (S-KPI metrics)
Latency report (Latency metrics)
Measurement module

FIGURE 3. Experimental test platform.

a Huawei 4G LTE USB dongle is used with a SIM card registered in the pilot network management module.

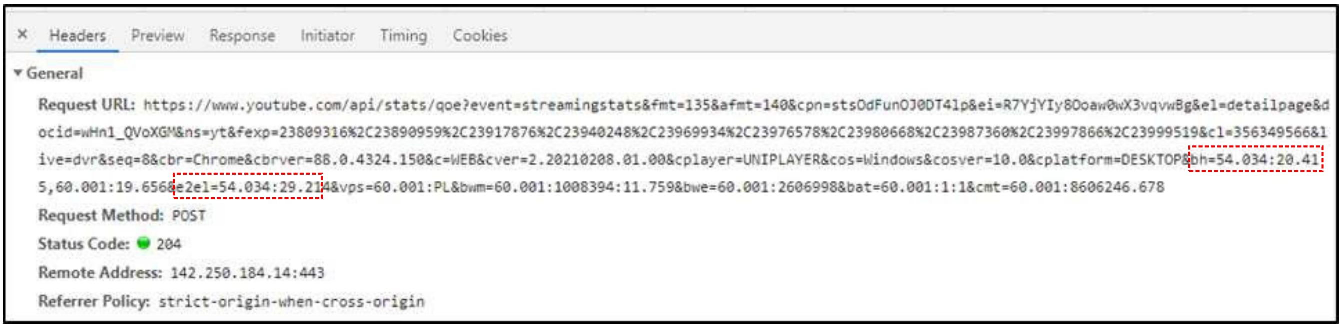
YouTube Studio allows controlling end-to-end latency by selecting between three different modes: ultra-low, low and normal [82]. Normal mode is the default option for no interaction with the audience. Low mode is designed for content that tolerate a latency between 2 and 15 seconds (comparable to broadcast latency), as in live sports. Ultra-low mode is used for highly interactive live streams, such as user-generated content in live video games.

The pilot LTE network comprises a radio access network consisting of 12 indoor picocells (Huawei BTS3911B) in the 2.6 GHz band (EARFC downlink 2850, EARFC uplink 18000) with 20 MHz system bandwidth. Only 2 picocells are used in the experiments, one for the user uploading the video content (referred to as upload picocell) and one for the users downloading the content (referred to as download picocell). Cells are logically connected to a compact virtualized core (Huawei eCNS600) and a computer with the network management tool (Huawei RH5885). These elements are physically interconnected through a transmission network consisting of a switch (Huawei S5700) and a Gigabit Ethernet local area network. The core network is connected to the Internet by a Switch S5700.

The measurement module consists of two terminal agents, a network emulator and a network probe. The two agents intend to reflect two users that visualize the live video feed. The first terminal agent runs Ericsson Network Performance Test (NPT) software [83] on a Samsung Galaxy S9 mobile terminal. Such an application mimics user interactions in a live video streaming session, while registering measurements of the main service key performance indicators (S-KPI) in a log file (NPT report). The second terminal agent is used to collect latency measurements not provided by NPT. It runs on a PC (Lenovo Thinkpad T560, I5 8th Gen processor, 8 GB RAM, Intel HD Graphics 520 video card and 500 GB hard disk) with a LTE dongle and replicates user behavior in an emulated Samsung Galaxy S9 handset. It is developed using a Python script from Selenium Webdriver and Google Chrome development tools. With the latter, a HTTP Archive (HAR) file is generated with all HTTP messages exchanged between server and client, which is later parsed to obtain end-to-end latency values. These periodic measurements are saved per session in the output report (latency report).

The network emulator is a PC running Ubuntu 19.10 (Eoan Ermine, 3 GHz i5-750 processor, 8 GB of RAM, with two network cards linked via a routing table) with the NetEm module [84] to modify the uplink conditions in a controlled way (e.g., uplink throughput). It is located between the upload picocell and the core of the pilot LTE network, allowing throttling of the uplink of the emission module (THRU_UP). Packet-level network measurements are collected on the Internet-side interface of the network emulator using a standard capturing tool (Tcpdump) [85]. Then, this data is processed with a traffic monitoring and analysis application (Network Probe) to extract basic quality-of-service (QoS) metrics per session, which are later used to relate network performance metrics to S-KPIs.

Three main measurement points are enabled in the platform, one on the live video output and two on the receiving



▾ General

```
Request URL: https://www.youtube.com/api/stats/qoe?event=streamingstats&fmt=135&afmt=140&cpn=stsOdFunOJ0DT41p&ei=R7YjYIy8Ooaw0wX3vqvwBg&el=detailpage&d
ocid=wHn1_QVoXGM&ns=yt&fexp=23809316%2C23890959%2C23917876%2C23940248%2C23969934%2C23976578%2C23980668%2C23987360%2C23997866%2C23999519&cl=356349566&l
ive=dvr&seq=8&cbr=Chrome&cbrver=88.0.4324.150&c=WEB&cver=2.20210208.01.00&cplayer=UNIPLAYER&cos=Windows&cosver=10.0&cplatform=DESKTOP&bh=54.034:20.41
5,60.001:19.656&e2el=54.034:29.214&vps=60.001:PL&bwm=60.001:1008394:11.759&bwe=60.001:2606998&bat=60.001:1:1&cmt=60.001:8606246.678
Request Method: POST
Status Code: ● 204
Remote Address: 142.250.184.14:443
Referrer Policy: strict-origin-when-cross-origin
```

FIGURE 4. QoE HTTP message header.

side. The former is dedicated to the collection of performance statistics of packet transmission at network level (TCP/IP metrics). The other two extract the S-KPIs needed to estimate the QoE of the live 360° video session in a mobile device (S-KPI metrics) and capture HTTP messages received by a personal computer for the extraction of latency measurements (latency metrics).

B. ANALYSIS METHODOLOGY

In this section, the key performance indicators for the analyzed service are defined. Then, the experimental methodology used in the tests is detailed.

1) S-KPI DEFINITION

Four fundamental S-KPIs determine user experience in live video streaming: the initial delay of video playback (initial buffering time), the reproduced image quality (given by the time period each resolution is displayed), the frequency of interruptions (rebuffering/stalling frequency) and the duration of each stalling event [86], [87]. Table 3 presents the image formats (itags) used in live 360° video sessions in YouTube. The above indicators are obtained per session from the NPT tool. Based on these indicators, a rough QoE measure in mean opinion score (MOS) is provided by NPT, following Mode 0 in ITU-T P.1203.3 standard [88]. Due to the difficulty of emulating changes in the viewport introduced by the user, NPT assumes that the 360° video user visualizes the whole frame, as in a conventional video streaming session. Hence, user QoE in a 360° video session is overestimated by NPT, since displayed image resolution would be less in practice, as the actual viewport size is only a fraction of the entire frame. This issue has to be corrected by scaling image resolution data.

A fifth S-KPI representing the average end-to-end latency per session is obtained by deciphering HTTP messages exchanged between the server and terminals visualizing video content. End-to-end latency is defined as the time difference between the moment the camera captures the event and the user displays it on his/her screen. It comprises: a) processing delays at the source, destination and YouTube server for coding, transcoding, packaging and decoding audio and video content, b) propagation and transmission delays on the uplink and downlink, and c) buffering times on the server and client receiving sides to cope with link speed fluctuations. Latency information is extracted from QoE HTTP messages sent periodically by the server. Fig. 4 shows an example of QoE message header. It contains the URL with a large list of parameters, from which two indicators are extracted, end-to-end latency (e2el) and buffer health (bh), the latter showing the video sequence time stored in the client buffer. In the example, the QoE message shows that, at 54.034 s, latency is 29.214 s and buffer health is 20.415 s. In contrast, at 60.001 s, latency is not reported, but buffer health is 19.656 s. As stated above, latency measurements are collected only for the terminal agent in the PC for simplicity.

TABLE 3. YouTube itag mapping for 360° live streaming [89].

itag	Resolution	Codec	Bitrate [Mbits/s]
133	426 x 240	H.264	0.3 - 0.7
134	640 x 360	H.264	0.4 - 1
135	854 x 480	H.264	0.5 - 2
136	1280 x 720	H.264	1.5 - 4
137	1920 x 1080	H.264	3 - 6
264	2560 x 1440	H.264	9 - 18
266	3840 x 2160	H.264	13 - 34

By analyzing HAR files, it has been observed that the latency parameter in QoE messages is sometimes missing. An inspection of traces has shown that, if the latency parameter is present, video playback is in live mode; otherwise, the downloaded video comes from a file that YouTube server has created from the live broadcast. Thus, it can be detected when YouTube switches between live and traditional video-on-demand (off-line) streaming depending on upstream network conditions. From this information, a sixth S-KPI is derived showing the ratio of time the server is delivering live content.

2) EXPERIMENTS

To check the impact of the above S-KPIs on a YouTube Live 360° video streaming session, a test battery was carried out consisting of the transmission of a 360° video from a live broadcast camera from April 1st to 30th 2020.

The broadcasted scene is a laboratory environment. A video profile of 2560 × 1280 pixels at 30 fps and H.264 AVC encoder is selected in the camera's management software. Fig. 5 shows the traffic profile at the output of the camera with unlimited transmission bandwidth. It is observed

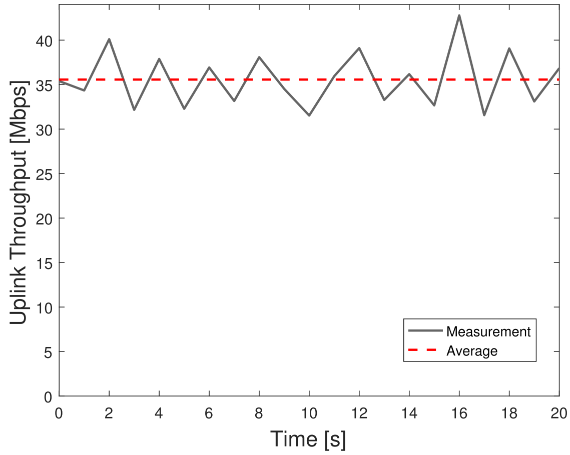


FIGURE 5. Evolution of source video bitrate.

that the source traffic is bursty, indicating that the codec of the camera is set to provide a variable output bitrate, with an average value reaching 36 Mbps.

For the distribution of live video, the YouTube platform is used as a gateway. Unless stated otherwise, the normal latency mode is selected, since it is default mode. The adopted streaming protocols are RTMP for ingest and Dynamic Adaptive Streaming for HTTP (DASH) for delivery. Note that, even if YouTube already supports DASH in the ingest link, this protocol is not suitable for ultra-low latency mode [90]. This is the main reason for selecting a non-HTTP-based protocol, like RTMP, as ingest protocol. For the delivery link, the use of a HTTP-based protocol, like DASH, ensures scalability by leveraging content delivery networks and native support on mobile devices.

Once live streaming is launched, the network emulator and the terminal agents run synchronized. Each client receives a 360° live video streaming session of 300 s, which is repeated until the end of the test battery. NetEm modifies the uplink data rate in the upstream every 60 minutes, limiting the link bandwidth to: 4, 5, 10, 20 or 50 Mbps. Below 4 Mbps, it is checked that streaming frequently stops, because YouTube server does not receive enough video to maintain smooth streaming, causing that most viewers experience rebuffering events [91]. Likewise, 50 Mbps is high enough to emulate ideal link conditions. The total time of a test cycle is 5 hours. During one month, a total of 10 test cycles are executed, resulting in 120 measurements per throttling state. It should be pointed out that only the radio access domain is controlled during the experiments. For this reason, a large set of measurements is required to filter out undesired fluctuations in link capacity due to domains outside the pilot LTE network.

The aim is to check the influence of TCP/IP metrics on the uplink of the ingest link on the 5 selected S-KPIs affecting user experience. As in [28], for simplicity, the analysis is restricted to the impact of uplink session throughput (THRU_UP), which ultimately reflects any problem in the ingest link (e.g., throttling, packet delay or packet losses). To isolate the effect of the uplink, all terminals emulating end users are located near the base station to ensure good radio conditions on the downlink. It will be shown later that the video resolution downloaded by user agents is low enough to ensure that the ingest link is the main bottleneck of the system.

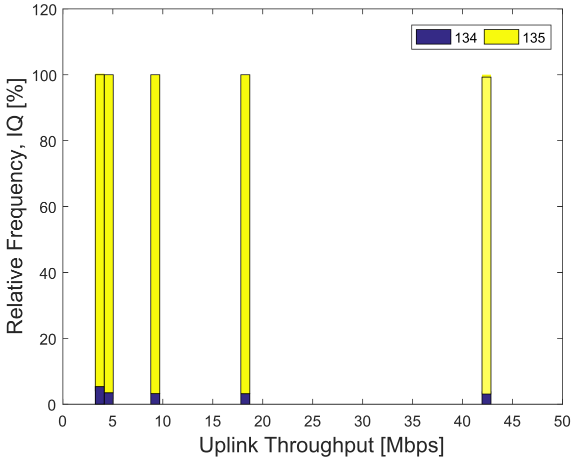


FIGURE 6. Distribution of itag values versus uplink throughput per session.

V. RESULTS

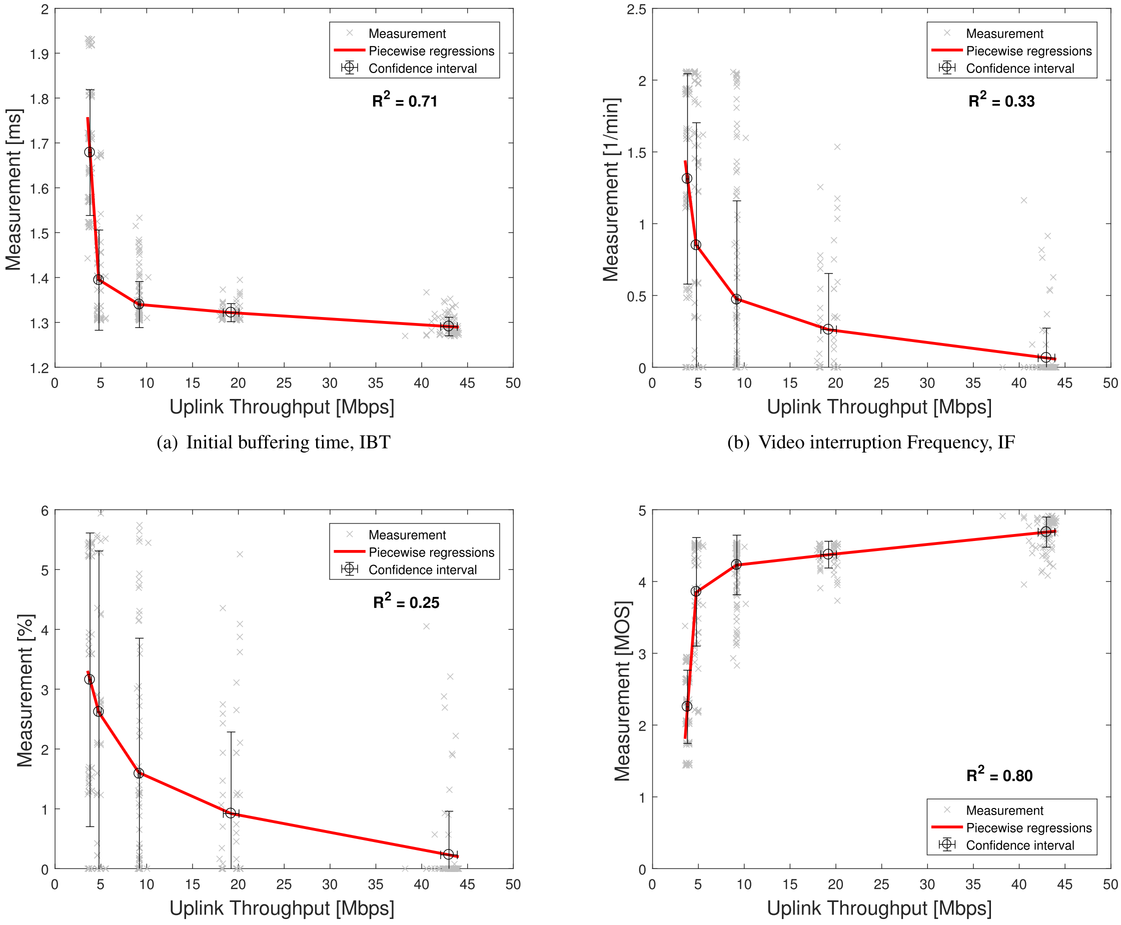
For clarity, the analysis starts with image quality statistics, which help to identify in which upstream conditions the link does not work properly. Then, the analysis is focused on the QoE statistics provided by NPT, computed as in ITU-T P.1203.3, which does not take viewport size and latency into account. Finally, the analysis is focused on latency statistics, which is the main contribution of this work.

A. IMAGE QUALITY DISTRIBUTION

Fig. 6 represents a stacked bar graph with the itag distribution received by the mobile terminal for different values of uplink session throughput. Each column represents a different throttling value. Column width reflects the range of THRU_UP for that setting. The y-axis shows the percentage of itags of each class within each column. It is observed that only two itag values appear (134, 135), with 135 the most frequent value, corresponding to a 854 x 480 video resolution and a video rate between 0.5 and 2 Mbps. The behavior is similar for all throttling values except for 4 Mbps. A more detailed analysis of the 4 Mbps HAR traces (not shown here) reflects stalling problems in 85% of video playback sessions. Consequently, more sessions that use lower resolutions (itag 134) are observed. Above 4 Mbps, uplink throttling in the ingest link does not directly affect the client's video image quality in terms of resolution. This is just a consequence of the lack of adaptation in video transmission in the ingest link with RTMP.

B. QoE

Fig. 7 shows a scatter plot of the 4 S-KPIs for the 360° live video streaming sessions as a function of uplink throughput in the ingest link (THRU_UP). Specifically, Fig. 7.(a)-(d) show the initial buffering time (IBT), the video interruption frequency (IF), the video interruption duration ratio (IDR) and



(a) Initial buffering time, IBT

(b) Video interruption Frequency, IF

(c) Video interruption duration ratio, IDR

(d) QoE estimated

FIGURE 7. S-KPIs versus uplink throughput per session.

the QoE estimated with ITU-T P.1203.3 model, respectively. Each point represents a 300 s session. The segmented regression curve is obtained by computing the horizontal and vertical averages for each throttling value (4/5/10/20/50 Mbps). The error bars show the 10% and 90% confidence intervals. The determination coefficient is also superimposed. For brevity, the analysis is focused on the factors degrading the QoE.

In Fig. 7d, it is observed that QoE decreases as THRU_UP decreases. A closer analysis reveals that QoE for high throughputs is limited by the initial buffering time. Specifically, a THRU_UP of 50 Mbps results in an average MOS of 4.68, below the maximum of 4.91. An inspection of Fig. 7 shows that THRU_UP values above 50 Mbps lead to IBT ≈ 1.29 ms, IF ≈ 0.07, IDR ≈ 0.23%. Thus, it is concluded that the 1.29 ms initial delay limits the maximum QoE to 4.68 MOS. In contrast, for lower throughputs, QoE is limited by stalling events. Specifically, for a THRU_UP of 4 Mbps, MOS decreases to 2.25 due to an IF increase from 0.07 to 1.41.

C. LATENCY

Fig. 8 shows the percentage of QoE messages received by the PC containing the latency parameter (CR). Again, the five center points (circles) correspond to the vertical (CR) and horizontal (THRU_UP) averages of measurements for each throttling value (4/5/10/20/50 Mbps). The error bars show the 10% and 90% confidence intervals. As latency measurements are periodic, the live connection ratio reflects the percentage of time that a session has been transmitting in live mode. It can be observed that, as uplink data rate improves, video sessions offer a live video service close to 100%. However,

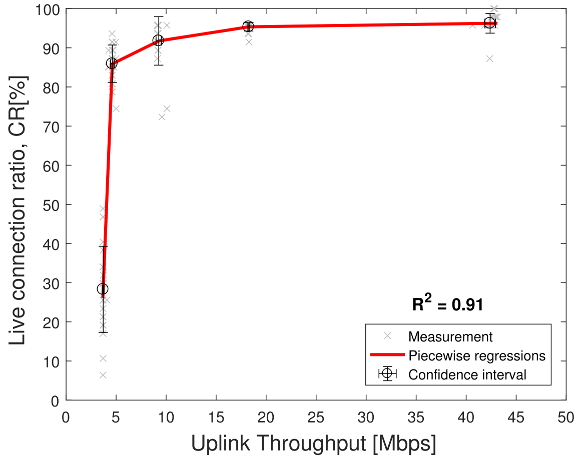


FIGURE 8. Live connection ratio versus uplink throughput per session.

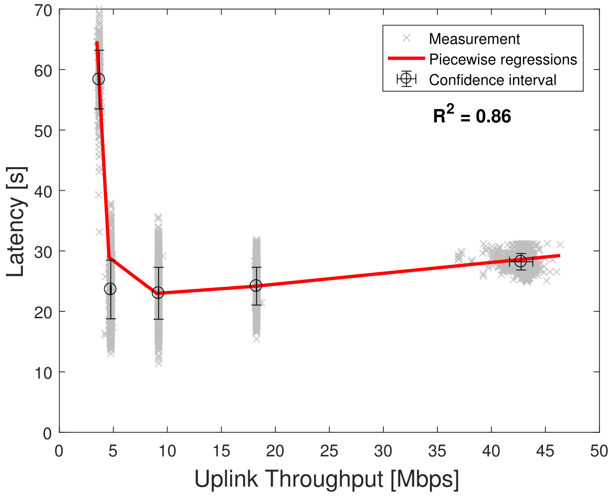


FIGURE 9. End-to-end latency versus uplink throughput per session.

for values of 5 Mbps, this ratio decreases to 85% approximately. Below 5 Mbps, YouTube does not offer continuous live video service.

Fig. 9 shows the end-to-end latency as a function of uplink throughput. The five circles correspond to the average latency and THRU_UP measurements for each throttling value (4/5/10/20/50 Mbps). The error bars for 10% and 90% confidence intervals are superimposed on the average value. The curve is the best piecewise regression, resulting in a determination coefficient of $R^2 = 0.86$. Measurements show that, for uplink data rates between 5 Mbps and 50 Mbps, the latency is kept below 30 s. The lowest value is reached at 10 Mbps, increasing slightly as uplink data rate increases. In contrast, for 4 Mbps, the latency shows extremely high values exceeding 60 s.

To check how much latency is due to the end-user buffer, Fig. 10 presents the playback buffer size as a function of the uplink data rate in the upstream. Again, each circle represents the average buffer size and THRU_UP for each throttling value, while the error bars show the 10% and 90% confidence intervals. The segmented regression curve that best fits the data is superimposed, resulting in a determination coefficient of $R^2 = 0.5$. In the figure, as in the previous case, two different behaviors are observed. From 5 to 50 Mbps, the buffer size is kept below 20 s and shows a slightly upward trend. In contrast, for 4 Mbps, it increases dramatically to 27 s on average. This result points out that playback buffer size not only depends on downstream conditions, but also on those of the upstream.

Playback buffer size increases as network conditions degrade to allow continuous playback on the client. Fig. 11 shows the temporal evolution of latency and buffer size for many 360° live video sessions for four throttling values in the uplink of the ingest link. For consistency, the figures only show samples obtained from QoE messages, where the values of both parameters, latency and buffer health, appear. Thus, buffer health data is subsampled, as it is also reported in other messages.

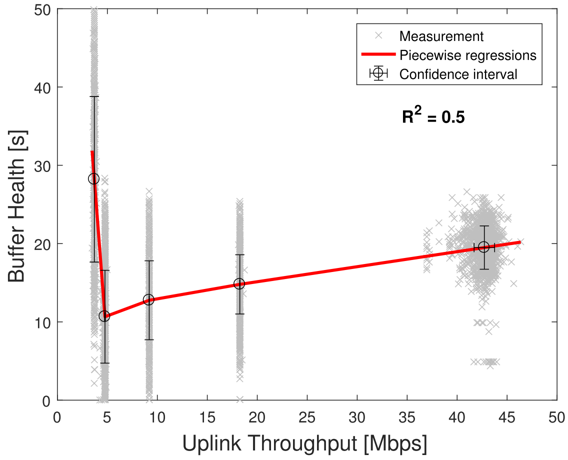


FIGURE 10. Playback buffer size versus uplink throughput per session.

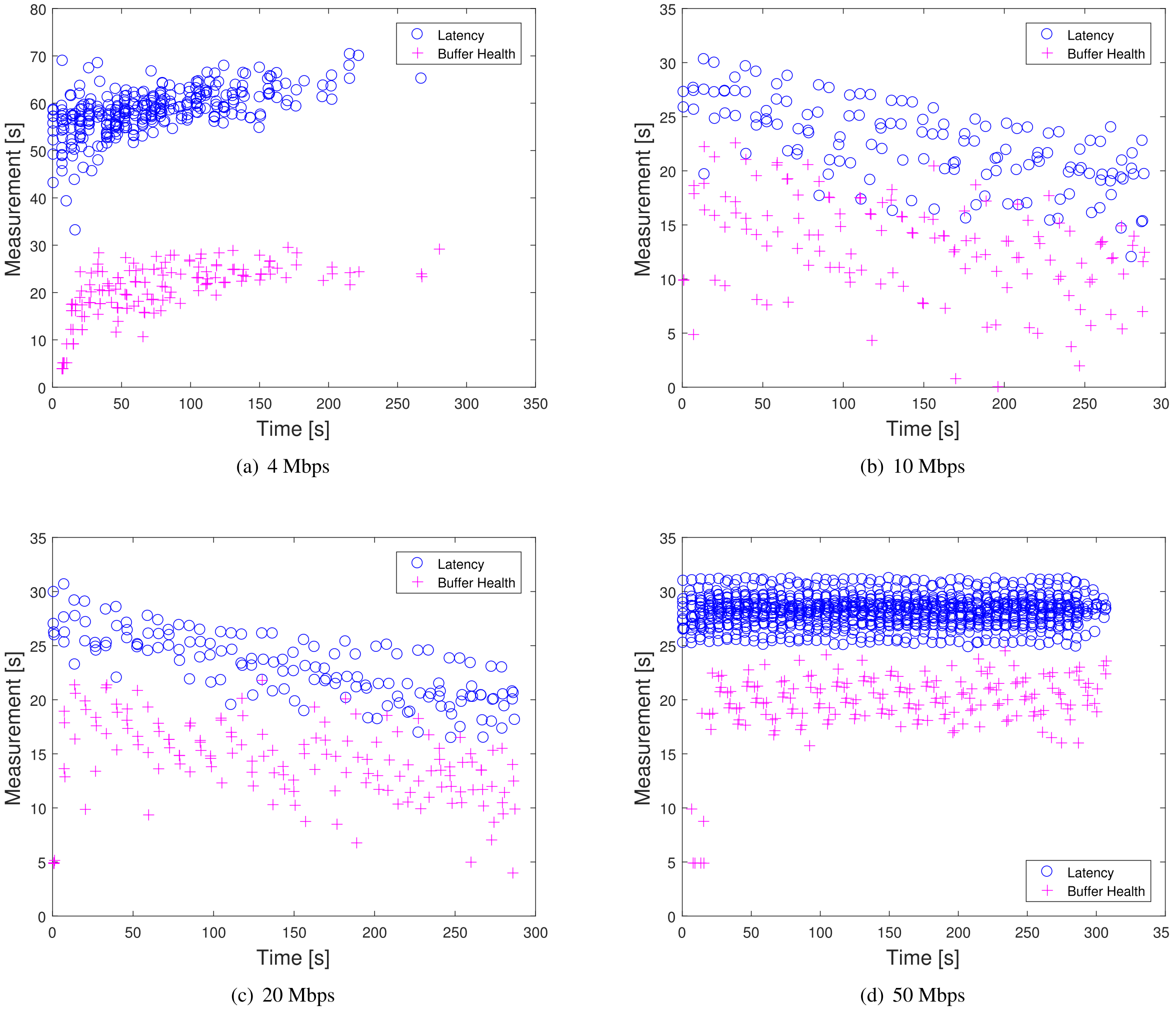
In the figure, three behaviors are distinguished:

a) The most stable pattern is observed at 50 Mbps, shown in Fig. 11d. In all sessions, an initial transient period of 30 seconds is observed, when buffer size rapidly increases, followed by a steady state, when buffer size fluctuates around 20 seconds. A closer look reveals a sawtooth pattern in buffer health due to the periodic reception of video segments of different sizes by the client. The duration of video segments can be estimated in steady state from the time difference between consecutive HTTP video playback requests [92]. By analyzing many HTTP traces, it has been confirmed that, in normal latency mode, when the buffer size is below a certain threshold, the next HTTP video playback request is sent after 1 s. As the buffer fills, the message is delayed 4 s, until the buffer is above a certain threshold, when the message is delayed 10 s. This behavior is in agreement with Fig. 11d, where it is observed that buffer health follows a sawtooth pattern.

b) For 10 and 20 Mbps, shown in Fig. 11b and 11c, latency and buffer health follow the same trend, showing a strong relationship. Initially, the client buffer fills up to the maximum target value, around 20 seconds, and then decreases throughout the session. Latency replicates the same behavior.

c) For 4 Mbps, shown in Fig. 11a, fewer points appear, clustered in the first half of the session. It is deduced that the number of QoE messages is less than in the previous cases, indicating that the session switches intermittently to the offline mode. Latency and buffer health have the same trend. After the initial buffer filling period, both delays reach their maximum values, of 30 and 70 seconds, respectively, which are kept throughout the session for the few moments of live transmission. Note the large value of end-to-end latency (70 seconds) for these extreme throttling conditions.

To further inspect the relationship between latency and buffer size, Fig. 12 shows a scatter plot of these two variables across multiple sessions for the different throttling settings.



(a) 4 Mbps

(b) 10 Mbps

(c) 20 Mbps

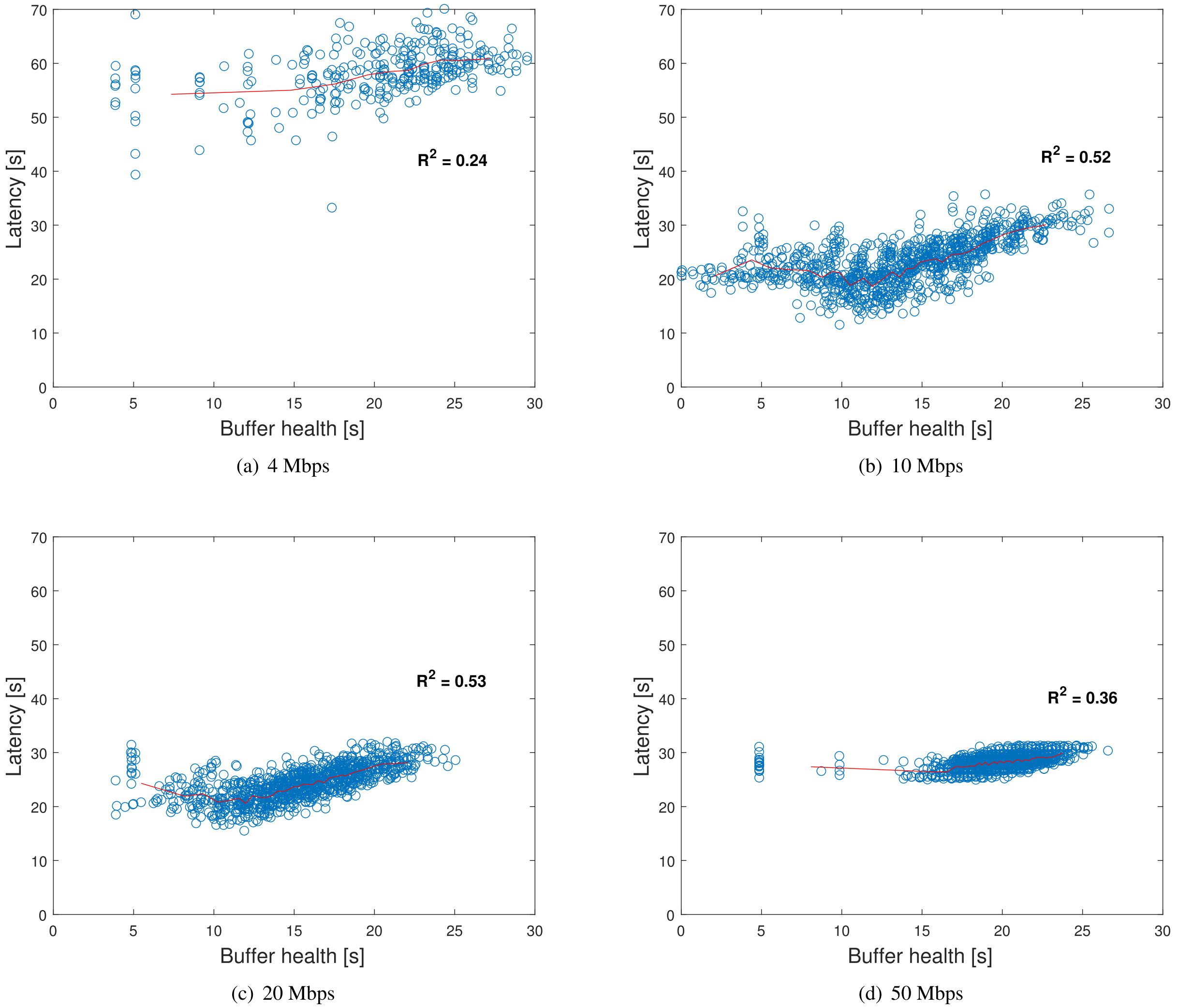
(d) 50 Mbps

FIGURE 11. Examples of latency evolution per session for different uplink throttling settings.

A piecewise regression curve is also included, together with the value of the determination coefficient. By observing Fig. 11(a)-(d), it can be deduced that most buffer health values below 10 s in Fig. 12(a)-(d) tend to occur at the beginning of sessions, when the buffer is still filling up. In this period, the trend line is almost flat, suggesting that latency reported in the initial transient period does not depend on buffer state. Thereafter, an increasing trend is observed, where latency rises as buffer size increases. Based on the value of the determination coefficient, correlation is weaker for 4 Mbps and stronger for cases above 10 Mbps. A closer analysis shows that the slope of the regression curve in the latter interval is close to 1 for 10 and 20 Mbps. This fact suggests that, in those uplink conditions, end-to-end latency increases by the same amount as playback buffer size. However, for 50 Mbps, this rising trend is much more attenuated, showing a slope much lower than 1. All these results show a clear dependence between latency and uplink capacity, which might be due to the variable output bitrate of the video source.

With an average target video bitrate of 36 Mbps at the source, if the uplink is throttled very aggressively (4 Mbps), a 3 fold increase of end-to-end latency is observed at the end of the session (from 20 to 60 seconds), causing at the same time switches to offline mode. In this case, the latency increase might be due to the delay imposed by the YouTube server transcoder, possibly due to unavailable image frames or sudden changes in service mode. As the ingest link deteriorates, YouTube server might increase its ingest buffer, which would lead to an increase in the overall system latency. At the same time, on the client size, the playback buffer size is also increased, approaching 30 s.

In the opposite conditions, when 50 Mbps throttling is applied on the uplink, the source bitrate is unrestricted and transmitted as if there were no link constraints. In this case, the latency is slightly higher than for 10 and 20 Mbps, but remains within very narrow limits during all experiments, as seen in the vertical error bar in Fig. 9. This constant value of latency suggests that the higher end-to-end delay could be due to an increased buffer length in the ingest server to cope with the higher video bitrate and increased computational load for generating multiple copies of the video at different resolutions. Moreover, there is an increase in the client buffer size compared to the lower throttling values.



(a) 4 Mbps

(b) 10 Mbps

(c) 20 Mbps

(d) 50 Mbps

FIGURE 12. Correlation between buffer health and latency measurements.

For intermediate conditions (5, 10, and 20 Mbps), the source video bitrate is throttled, but a live video service is maintained. In these cases, the video bitrate is lower, so transcoding in the ingest server is easier. Thus, latency values strongly depend on the playback buffer time. At the beginning of the connection, the buffer is filled to the maximum value to avoid interruptions in video playback, which implies maximum latency values. As session progresses, there is a reduction in buffer size and, therefore, also in latency. Such a reduction might be originated by YouTube delivery server, which, in an attempt to reduce latency in live transmissions, might progressively reduce the size of the playback buffer, and, thus, the end-to-end latency.

To check the impact of latency settings, Fig. 13 shows latency values obtained in a session without throttling for the three latency modes offered by YouTube (normal, low and ultra-low). Each point represents a latency measurement, reported every 6 seconds. As expected, a significant reduction in latency is achieved by using the low and ultra-low latency modes (3-fold reduction for low mode, 5-fold for ultra-low mode). More interestingly, it is observed that ultra-low mode presents a lower number of reports. Such an intermittent reporting indicates the switching to the offline service during those periods without reports.

The above-described latency reduction is partly achieved by dividing the live video stream in smaller segments. Fig. 14 shows the distribution of the size of video segments for the three latency modes. For a fair comparison, measurements are segregated per itag. It is observed that video segment size is smaller in low and ultra-low modes. Specifically, the median values are 69, 106, 179 and 360 for ultra-low (itag 136), ultra-low (itag 137), low (itag 137) and normal (itag 137) modes, respectively.

In the absence of stalling events, video duration per segment can be deduced from the time difference between consecutive video segment requests. Fig. 15 shows the distribution of video request interarrival times for the different latency modes and itags. In the figure, it is clear that the normal mode behaves differently from the low/ultra-low modes. In normal mode, the time gap between video requests fluctuates between 1, 4 and 10 seconds, depending on playback buffer state. In contrast, in low and ultra-low

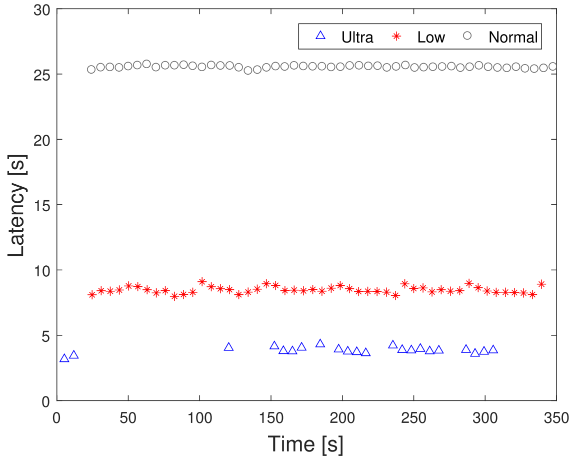


FIGURE 13. Latency values in a YouTube session for normal, low and ultra-low mode.

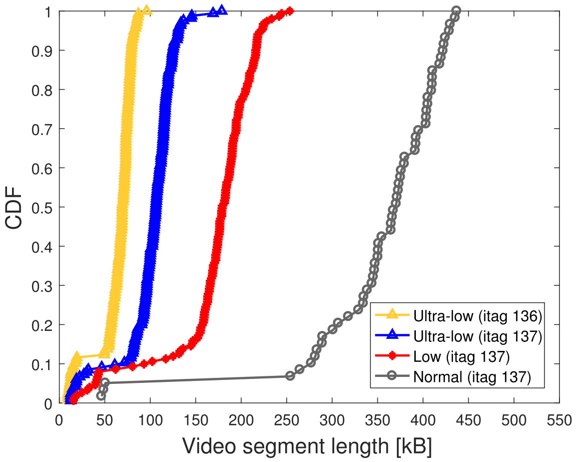


FIGURE 14. Distribution of the size of the video segments for normal, low and ultra-low mode.

modes, such a time gap is nearly fixed to 2 and 1 seconds, respectively. These values are in line with the observed decrease in segment size.

Ultimately, it is the combination of segment size and frequency that controls playback buffer state. Fig. 16 depicts the distribution of instantaneous video bitrate, estimated by dividing video segment request size and interarrival time on a per-segment basis. As expected, it is observed that frequent changes in interarrival times in the normal mode cause larges changes of bitrate. In contrast, low/ultra-low modes keep bitrate almost constant. Likewise, the lowest median bitrate is obtained for the lowest itag (136). More subtle is the fact that median bitrate values slightly differ between modes for the highest itag (137). The same holds for the overall video bitrate (i.e., total bits/session duration), which is 557, 834, 716 and 706 kbps for ultra-low (itag 136), ultra-low (itag 137), low (itag 137) and normal (itag137) modes, respectively. This effect is possibly due to a larger protocol overhead for smaller segment sizes.

D. DISCUSSION

The above results can be used to derive parametric network-layer QoE models that estimate end-to-end video latency from uplink throughput measurements on a per-session basis. These methods would be suitable for encrypted traffic, provided that the different traffic flows can be isolated. The main difficulty lies in the characterization of user sensitivity to latency, as it is application dependent. In sport broadcast, the target latency is similar to that of the traditional broadcast chain, which varies between 2 and 15 seconds [63]. In this context, a minimum latency is needed to ensure that content can be censored. In contrast, interactive applications where the streamer interacts with the audience, such as online gambling and e-sports, require lower latency values to ensure a fluid conversation. Likewise, in teleoperated systems with haptic feedback, target latency is in the order of milliseconds [93]. Such application-dependent performance thresholds can only be derived from subjective tests or massive analysis of session traces collected in live networks [94].

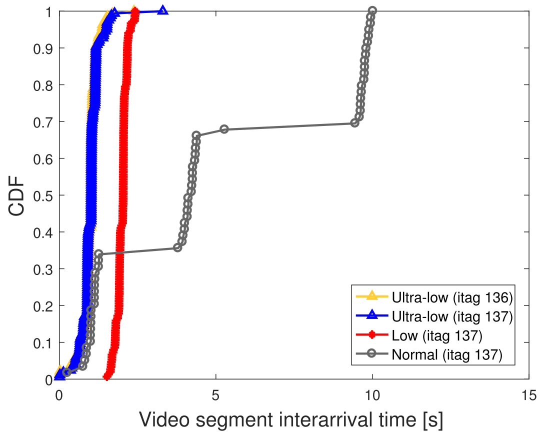


FIGURE 15. Distribution of video segment interarrival times for normal, low and ultra-low mode.

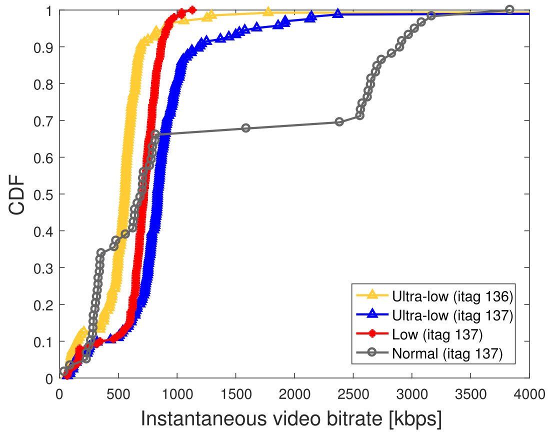


FIGURE 16. Distribution of instantaneous video bitrate for normal, low and ultra-low mode.

VI. CONCLUSION

Low-latency video streaming has recently gained momentum in the industry and research community. This article has presented a study of the impact of the uplink of a Long Term Evolution network in the ingest path of a live 360° video broadcast using YouTube streaming platform. The analysis has been performed by first introducing throttling in the uplink of the upload cell and then collecting network-level traces for TCP/IP metrics, HTTP traces for latency measurements and application-level traces for MOS calculation.

Results have shown that, even if uplink performance does not influence the image quality of the downloaded video segments, it has a strong impact on end-to-end latency. It has also been shown that the relationship between uplink throughput and end-to-end latency is not straight-forward, as it depends on multiple mechanisms located in the content delivery network and user terminal. These results justify the need for new ways to estimate end-to-end latency from TCP/IP metrics.

Future work will check the impact of activating DASH protocol in the ingest link, which might be suitable for premium content requiring higher image quality at the expense of a slightly higher latency. Likewise, the experiments will be repeated in a 5G system, where configuring an adequate numerology and a network slice for ultra-reliable low-latency services will reduce latency in the access domain. These enhanced capabilities will pave the way to tactile Internet applications based on immersive video.

ACKNOWLEDGMENT

The authors thank Ericsson Spain for the use of the NPT user agent.

REFERENCES

[1] Sandvine. (Mar. 2021). *The Mobile Internet Phenomena Report*. Accessed: Mar. 2021. [Online]. Available: https://www.sandvine.com/download-report-mobile-internet-phenomena-report-2020-sandvine

[2] Ericsson. (Mar. 2021). *Ericsson Mobility Report*. Accessed: Mar. 2021. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/june-2020

[3] S. Hollister. (2015). *YouTube's Ready to Blow Your Mind With 360-Degree Videos*. Accessed: Mar. 2021. [Online]. Available: https://gizmodo.com/youtubes-ready-to-blow-your-mind-with-360-degree-videos-1690989402

[4] C.-L. Fan, W.-C. Lo, Y.-T. Pai, and C.-H. Hsu, "A survey on 360° video streaming: Acquisition, transmission, and display," *ACM Comput. Surveys*, vol. 52, no. 4, pp. 1–36, Sep. 2019.

[5] A. Yaqoob, T. Bi, and G.-M. Muntean, "A survey on adaptive 360° video streaming: Solutions, challenges and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2801–2838, Jul. 2020.

[6] A. Banerjee, "Revolutionizing CEM with subscriber-centric network operations and QoE strategy," Accanto Syst., Heavy Reading, Paijanne Tavastia, Finland, White Paper 1, 2014.

[7] R. E. Hattachi and J. Erfanian, "Next generation mobile networks alliance (NGMN): 5G, white paper," NGMN Ltd., Frankfurt, Germany, 2015, pp. 1–125, vol. 1, no. 2.

[8] R. K. Mok, E. W. Chan, and R. K. Chang, "Measuring the quality of experience of HTTP video streaming," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage.*, May 2011, pp. 485–492.

[9] A. Rao, A. Legout, Y.-S. Lim, D. Towsley, C. Barakat, and W. Dabbous, "Network characteristics of video streaming traffic," in *Proc. 7th Conf. Emerg. Netw. EXperiments Technol. (CoNEXT)*, 2011, p. 25.

[10] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao, "YouTube everywhere: Impact of device and infrastructure synergies on user experience," in *Proc. ACM SIGCOMM Conf. Internet Meas. Conf. (IMC)*, 2011, pp. 345–360.

[11] P. Ameigeiras, J. Ramos-Munoz, J. Navarro-Ortiz, and J. M. Lopez-Soler, "Analysis and modelling of YouTube traffic," *Trans. Emerg. Telecommun. Technol.*, vol. 23, no. 4, pp. 360–377, 2012.

[12] K. D. Singh, Y. Hadjadj-Aoul, and G. Rubino, "Quality of experience estimation for adaptive HTTP/TCP video streaming using H.264/AVC," in *Proc. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2012, pp. 127–131.

[13] T. Hoßfeld, R. Schatz, E. Biersack, and L. Plissonneau, "Internet video delivery in YouTube: From traffic measurements to quality of experience," in *Data Traffic Monitoring and Analysis*. Vienna, Austria: Springer, 2013, pp. 264–301.

[14] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of HTTP adaptive streaming," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 469–492, Sep. 2014.

[15] F. Wamser, P. Casas, M. Seufert, C. Moldovan, P. Tran-Gia, and T. Hossfeld, "Modeling the YouTube stack: From packets to quality of experience," *Comput. Netw.*, vol. 109, pp. 211–224, Nov. 2016.

[16] T. Mangla, E. Halepovic, M. Ammar, and E. Zegura, "eMIMIC: Estimating HTTP-based video QoE metrics from encrypted network traffic," in *Proc. Netw. Traffic Meas. Anal. Conf. (TMA)*, Jun. 2018, pp. 1–8.

[17] W. Huang, Y. Zhou, X. Xie, D. Wu, M. Chen, and E. Ngai, "Buffer state is enough: Simplifying the design of QoE-aware HTTP adaptive video streaming," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 590–601, Jun. 2018.

[18] N. Barman and M. G. Martini, "QoE modeling for HTTP adaptive video streaming—A survey and open challenges," *IEEE Access*, vol. 7, pp. 30831–30859, 2019.

[19] T. Mangla, E. Halepovic, M. Ammar, and E. Zegura, "Using session modeling to estimate HTTP-based video QoE metrics from encrypted network traffic," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 3, pp. 1086–1099, Sep. 2019.

[20] Tisa-Selma, A. Bentaleb, and S. Harous, "Inferring quality of experience for adaptive video streaming over HTTPS and QUIC," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2020, pp. 81–87.

[21] R. Schatz, T. Hoßfeld, and P. Casas, "Passive Youtube QoE monitoring for ISPs," in *Proc. 6th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput.*, Jul. 2012, pp. 358–364.

[22] P. Casas, M. Seufert, and R. Schatz, "YOUQMON: A system for on-line monitoring of YouTube QoE in operational 3G networks," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 2, pp. 44–46, Dec. 2013.

[23] A. Baer, P. Casas, A. D'Alconzo, P. Fiadino, L. Golab, M. Mellia, and E. Schikuta, "DBStream: A holistic approach to large-scale network traffic monitoring and analysis," *Comput. Netw.*, vol. 107, pp. 5–19, Oct. 2016.

[24] F. Qian, L. Ji, B. Han, and V. Gopalakrishnan, "Optimizing 360 video delivery over cellular networks," in *Proc. 5th Workshop All Things Cellular, Oper., Appl. Challenges*, Oct. 2016, pp. 1–6.

[25] L. Skorin-Kapov, M. Varela, T. Hoßfeld, and K.-T. Chen, "A survey of emerging concepts and challenges for QoE management of multimedia services," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 14, no. 2s, p. 29, 2018.

[26] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360ProbDASH: Improving QoE of 360 video streaming using tile-based HTTP adaptive streaming," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 315–323.

[27] T. Mangla, E. Zegura, M. Ammar, E. Halepovic, K.-W. Hwang, R. Jana, and M. Platania, "VideoNOC: Assessing video QoE for network operators using passive measurements," in *Proc. 9th ACM Multimedia Syst. Conf.*, Jun. 2018, pp. 101–112.

[28] L. R. Jiménez, M. Solera, and M. Toril, "A network-layer QoE model for YouTube live in wireless networks," *IEEE Access*, vol. 7, pp. 70237–70252, 2019.

[29] O. Oyman, J. Foerster, Y.-J. Tcha, and S.-C. Lee, "Toward enhanced mobile video services over WiMAX and LTE [WiMAX/LTE update]," *IEEE Commun. Mag.*, vol. 48, no. 8, pp. 68–76, Aug. 2010.

[30] S. Sesia, I. Toufik, and M. Baker, *LTE—The UMTS Long Term Evolution: From Theory to Practice*. Hoboken, NJ, USA: Wiley, 2011.

[31] A. E. Essaili, Z. Wang, E. Steinbach, and L. Zhou, "QoE-based cross-layer optimization for uplink video transmission," *ACM Trans. Multimedia Comput. Commun.*, vol. 12, no. 1, pp. 1–22, Aug. 2015.

[32] H. J. Kim, D. G. Yun, H.-S. Kim, K. S. Cho, and S. G. Choi, "QoE assessment model for video streaming service using QoS parameters in wired-wireless network," in *Proc. 14th Int. Conf. Adv. Commun. Technol. (ICACT)*, 2012, pp. 459–464.

[33] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of quality of experience of video-on-demand services: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 401–418, 1st Quart., 2015.

[34] M. Gadaleta, F. Chiariotti, M. Rossi, and A. Zanella, "D-DASH: A deep Q-learning framework for DASH video streaming," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 703–718, Dec. 2017.

[35] J. D. Vriendt, D. D. Vleeschauwer, and D. Robinson, "Model for estimating QoE of video delivered using HTTP adaptive streaming," in *Proc. Int. Symp. Integr. Netw. Manage. (IM)*, 2013, pp. 1288–1293.

[36] M. Claeys, S. Latre, J. Famaey, and F. D. Turck, "Design and evaluation of a self-learning HTTP adaptive video streaming client," *IEEE Commun. Lett.*, vol. 18, no. 4, pp. 716–719, Apr. 2014.

[37] W. Robitza, M.-N. Garcia, and A. Raake, "A modular HTTP adaptive streaming QoE model—Candidate for ITU-T P.1203 ('P.NATS')," in *Proc. 9th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, 2017, pp. 1–6.

[38] P. Casas, A. D'Alconzo, F. Wamser, M. Seufert, B. Gardlo, A. Schwind, P. Tran-Gia, and R. Schatz, "Predicting QoE in cellular networks using machine learning and in-smartphone measurements," in *Proc. 9th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, May 2017, pp. 1–6.

[39] P. Casas and S. Wassermann, "Improving QoE prediction in mobile video through machine learning," in *Proc. 8th Int. Conf. Netw. Future (NOF)*, Nov. 2017, pp. 1–7.

[40] I. Orsolic, D. Pevec, M. Suznjevic, and L. Skorin-Kapov, "A machine learning approach to classifying YouTube QoE based on encrypted network traffic," *Multimedia Tools Appl.*, vol. 76, no. 21, pp. 22267–22301, 2017.

[41] E. Demirbilek and J.-C. Grégoire, "Machine learning–based parametric audiovisual quality prediction models for real-time communications," *ACM Trans. Multimedia Comput.*, vol. 13, no. 2, pp. 16:1–16:25, Mar. 2017.

[42] W. Pan and G. Cheng, "QoE assessment of encrypted YouTube adaptive streaming for energy saving in smart cities," *IEEE Access*, vol. 6, pp. 25142–25156, 2018.

[43] E. Upenik, M. Řeřábek, and T. Ebrahimi, "Testbed for subjective evaluation of omnidirectional visual content," in *Proc. Picture Coding Symp. (PCS)*, 2016, pp. 1–5.

[44] X. Liu, Q. Xiao, V. Gopalakrishnan, B. Han, F. Qian, and M. Varvello, "360° innovations for panoramic video streaming," in *Proc. 16th ACM Workshop Hot Topics Netw.*, Nov. 2017, pp. 50–56.

[45] B. Zhang, J. Zhao, S. Yang, Y. Zhang, J. Wang, and Z. Fei, "Subjective and objective quality assessment of panoramic videos in virtual reality environments," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2017, pp. 163–168.

[46] M. Xu, C. Li, S. Zhang, and P. L. Callet, "State-of-the-art in 360° video/image processing: Perception, assessment and compression," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 1, pp. 5–26, Jan. 2020.

[47] H. Yuan, S. Zhao, J. Hou, X. Wei, and S. Kwong, "Spatial and temporal consistency-aware dynamic adaptive streaming for 360-degree videos," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 1, pp. 177–193, Jan. 2019.

[48] R. Schatz, A. Zabrovskiy, and C. Timmerer, "Tile-based streaming of 8K omnidirectional video: Subjective and objective QoE evaluation," in *Proc. 11th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, Jun. 2019, pp. 1–6.

[49] R. Schatz, A. Sackl, C. Timmerer, and B. Gardlo, "Towards subjective quality of experience assessment for omnidirectional video streaming," in *Proc. 9th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, May 2017, pp. 1–6.

[50] W. Zou and F. Yang, "Measuring quality of experience of novel 360-degree streaming video during stalling," in *Proc. Int. Conf. Communicatins Netw. China*. Cham, Switzerland: Springer, 2017, pp. 417–424.

[51] W. Zhang, W. Zou, and F. Yang, "The impact of stalling on the perceptual quality of HTTP-based omnidirectional video streaming," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 4060–4064.

[52] A. Muhammad, W. Jing, A. Ullah, K. Wahab, A. Sadique, and F. Zesong, "Measuring quality of experience for 360-degree videos in virtual reality," *Sci. China Inf. Sci.*, vol. 63, pp. 1–5, Oct. 2020.

[53] M. S. Anwar, J. Wang, A. Ullah, W. Khan, S. Ahmad, and Z. Li, "Impact of stalling on QoE for 360-degree virtual reality videos," in *Proc. Int. Conf. Signal, Inf. Data Process. (ICSIDP)*, 2019, pp. 1–6.

[54] K. Kanai, B. Wei, Z. Cheng, M. Takeuchi, and J. Katto, "Methods for adaptive video streaming and picture quality assessment to improve QoS/QoE performances," *IEICE Trans. Commun.*, vol. E102.B, no. 7, pp. 1240–1247, 2019.

[55] M. S. Anwar, J. Wang, W. Khan, A. Ullah, S. Ahmad, and Z. Fei, "Subjective QoE of 360-degree virtual reality videos and machine learning predictions," *IEEE Access*, vol. 8, pp. 148084–148099, 2020.

[56] M. S. Anwar, J. Wang, S. Ahmad, A. Ullah, W. Khan, and Z. Fei, "Evaluating the factors affecting QoE of 360-degree videos and cybersickness levels predictions in virtual reality," *Electronics*, vol. 9, no. 9, p. 1530, Sep. 2020.

[57] C. Perfecto, M. S. Elbamby, J. D. Ser, and M. Bennis, "Taming the latency in multi-user VR 360°: A QoE-aware deep learning-aided multicast framework," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2491–2508, Apr. 2020.

[58] A. E. Essaili, L. Zhou, D. Schroeder, E. Steinbach, and W. Kellerer, "QoE-driven live and on-demand LTE uplink video transmission," in *Proc. IEEE 13th Int. Workshop Multimedia Signal Process.*, Oct. 2011, pp. 1–6.

[59] F. Pervez and M. S. Raheel, "QoE-based network-centric resource allocation for on-demand uplink adaptive HTTP streaming over LTE network," in *Proc. IEEE 8th Int. Conf. Appl. Inf. Commun. Technol. (AICT)*, Oct. 2014, pp. 1–5.

[60] K. Nihei, H. Yoshida, N. Kai, D. Kanetomo, and K. Satoda, "QoE maximizing bitrate control for live video streaming on a mobile uplink," in *Proc. 14th Int. Conf. Telecommun. (ConTEL)*, Jun. 2017, pp. 91–98.

[61] J. Yang, J. Luo, and F. Yang, "Quality of experience-driven resource allocation optimized for 360-degree video transmission over LTE uplink," *DEStech Trans. Comput. Sci. Eng.*, p. 1, May 2018.

[62] J. Li, R. Feng, W. Sun, Z. Liu, and Q. Li, "QoE-driven coupled uplink and downlink rate adaptation for 360-degree video live streaming," *IEEE Commun. Lett.*, vol. 24, no. 4, pp. 863–867, Apr. 2020.

[63] P. Cluff. *The Low Latency Live Streaming Landscape in 2019*. Accessed: Mar. 2021. [Online]. Available: https://mux.com/blog/the-low-latency-live-streaming-landscape-in-2019

[64] R. Huysegems, B. De Vleeschauwer, T. Wu, and W. Van Leekwijck, "SVC-based HTTP adaptive streaming," *Bell Labs Tech. J.*, vol. 16, no. 4, pp. 25–41, 2012.

[65] S. Wei and V. Swaminathan, "Low latency live video streaming over HTTP 2.0," in *Proc. Netw. Operating Syst. Support Digit. Audio Video Workshop*, Mar. 2014, pp. 37–42.

[66] S. Zhao, Z. Li, and D. Medhi, "Low delay streaming of DASH content with WebRTC data channel," in *Proc. IEEE/ACM 24th Int. Symp. Qual. Service (IWQoS)*, Jun. 2016, pp. 1–2.

[67] N. Bouzakaria, C. Concolato, and J. L. Feuvre, "Overhead and performance of low latency live streaming using MPEG-DASH," in *Proc. 5th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Jul. 2014, pp. 92–97.

[68] W. Law. *Best Practice for Ultra-Low-Latency Streaming Using Chunked-Encoded and Chunked-Transferred*. Accessed: Mar. 2021. [Online]. Available: https://tinyurl.com/47f43e76

[69] L. Sun, T. Zong, Y. Liu, Y. Wang, and H. Zhu, "Optimal strategies for live video streaming in the low-latency regime," in *Proc. IEEE 27th Int. Conf. Netw. Protocols (ICNP)*, Oct. 2019, pp. 1–4.

[70] K. Miller, A.-K. Al-Tamimi, and A. Wolisz, "QoE-based low-delay live streaming using throughput predictions," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 13, no. 1, pp. 1–24, Jan. 2017.

[71] G. Zhang and J. Y. B. Lee, "LAPAS: Latency-aware playback-adaptive streaming," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6.

[72] 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*, Dec. 2009, pp. 194–200.

[73] A. Kryczka, A. Arefin, and K. Nahrstedt, "AvCloak: A tool for black box latency measurements in video conferencing applications," in *Proc. IEEE Int. Symp. Multimedia*, Dec. 2013, pp. 271–278.

[74] A. Kaknjo, M. Rao, E. Omerdic, L. Robinson, D. Toal, and T. Newe, "Real-time video latency measurement between a robot and its remote control station: Causes and mitigation," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–19, Dec. 2018.

[75] YouTube. *Send YouTube Debug Information*. Accessed: Mar. 2021. [Online]. Available: https://support.google.com/youtube/answer/7519898

[76] H. Holma and A. Toskala, *LTE for UMTS: Evolution to LTE-Advanced*. Hoboken, NJ, USA: Wiley, 2011.

[77] 3GPP. (Mar. 2021). *TR 25.912 Universal Mobile Telecommunications System (UMTS), Antipolis Technology Park in France, European Telecommunications Standards Institute*. Accessed: Mar. 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_TR/125900_125999/125912/15.00.00_60/tr_125912v150000p.pdf

[78] C. A. Garcia-Perez and P. Merino, "Enabling low latency services on LTE networks," in *Proc. IEEE 1st Int. Workshops Found. Appl. Self Syst. (FAS W)*, Sep. 2016, pp. 248–255.

[79] I. Grigorik, *High Performance Browser Networking: What Every Web Developer Should Know About Networking and Web Performance*. Newton, MA, USA: O'Reilly Media, 2013.

[80] CyberLink. *Introducing Gear 360 ActionDirector*. Accessed: Mar. 2021. [Online]. Available: https://www.cyberlink.com/learning/gear-360-actiondirector/606/introducing-gear-360-actiondirector

[81] YouTube. *Create a Live Stream With an Encoder*. Accessed: Mar. 2021. [Online]. Available: https://support.google.com/youtube/answer/2907883?hl=en

[82] YouTube. *Live Streaming Latency*. Accessed: Mar. 2021. [Online]. Available: https://support.google.com/youtube/answer/7444635?hl=en

[83] *S-KPI Benchmarking 20.3 NPT User Guide*, Ericsson, Kista, Stockholm, Sweden, 2020.

[84] Linux Foundation. *NetEm*. Accessed: Mar. 2021. [Online]. Available: https://wiki.linuxfoundation.org/netem

[85] Tcpdump-Workers. *Tcpdump*. Accessed: Mar. 2021. [Online]. Available: http://www.tcpdump.org

[86] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, "Quantification of YouTube QoE via crowdsourcing," in *Proc. IEEE Int. Symp. Multimedia*, Dec. 2011, pp. 494–499.

[87] P. Casas, A. Sackl, S. Egger, and R. Schatz, "YouTube & Facebook quality of experience in mobile broadband networks," in *Proc. IEEE Globecom Workshops*, Dec. 2012, pp. 1269–1274.

[88] *Parametric Bitstream-Based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services Over Reliable Transport-Quality Integration Module*, document ITU-T Rec. P.1203, 2019.

[89] YouTube. *Live Encoder Settings, Bitrates, and Resolutions*. Accessed: Mar. 2021. [Online]. Available: https://support.google.com/youtube/answer/2853702

[90] YouTube. *YouTube Live Streaming Ingestion Protocol Comparison*. Accessed: Mar. 2021. [Online]. Available: https://developers.google.com/youtube/v3/live/guides/ingestion-protocol-comparison

[91] YouTube. *Live Streaming Error Messages*. Accessed: Mar. 2021. [Online]. Available: https://support.google.com/youtube/answer/3006768?

[92] A. Mondal, S. Sengupta, B. R. Reddy, M. J. V. Koundinya, C. Govindarajan, P. De, N. Ganguly, and S. Chakraborty, "Candid with YouTube: Adaptive streaming behavior and implications on data consumption," in *Proc. 27th Workshop Netw. Operating Syst. Support Digit. Audio Video*, Jun. 2017, pp. 19–24.

[93] X. Xu, Q. Liu, and E. Steinbach, "Toward QoE-driven dynamic control scheme switching for time-delayed teleoperation systems: A dedicated case study," in *Proc. IEEE Int. Symp. Haptic, Audio Vis. Environ. Games (HAVE)*, Oct. 2017, pp. 1–6.

[94] A. J. García, C. Gijón, M. Toril, and S. Luna-Ramírez, "Data-driven construction of user utility functions from radio connection traces in LTE," *Electronics*, vol. 10, no. 7, p. 829, Mar. 2021.

LUIS ROBERTO JIMÉNEZ (Graduate Student Member, IEEE) received the M.S. degree in electronics and communications engineering from Santo Domingo Institute of Technology (INTEC), Dominican Republic, in 2013, and the M.S.E. degree in telematics and telecommunication networks from the University of Málaga, Spain, in 2015, where he is currently pursuing the Ph.D. degree. His current research interests include self-optimization networks, performance evaluation of multimedia services over mobile networks, and data analytics. He was a recipient of Junta de Andalucía Scholarship (2017–2021) over methods planning and optimizing QoE in B4G networks.

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. Since 1996, she has been lecturing in several universities, such as UPC, the Universidad Nacional Autonoma de Mexico (UNAM), and Universidad de Málaga (UMA). She is currently an Associate Professor with the Department of Communication Engineering, 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.

MATÍAS TORIL received the M.S. and Ph.D. degrees in telecommunication engineering from the University of Málaga, Spain, in 1995 and 2007, respectively. Since 1997, he has been a Lecturer with the Communications Engineering Department, University of Málaga, where he is currently a Full Professor. He has coauthored more than 150 publications in leading conferences and journals and eight patents owned by Nokia and Ericsson. His current research interests include self-organizing networks, radio resource management, and data analytics.

SALVADOR LUNA-RAMÍREZ received the M.S. degree in telecommunication engineering and the Ph.D. degree from the University of Málaga, Spain, in 2000 and 2010, respectively. He has been a Lecturer with the Communications Engineering Department, University of Málaga, since 2000, where he is currently a Full Professor. His research interests include self-optimization of mobile radio access networks and management of radio resources, in addition to research and collaboration with companies in the field of acoustic engineering.

JUAN L. BEJARANO-LUQUE received the B.S. degree in telecommunications engineering and the M.S. degree in acoustic engineering from the University of Málaga, Málaga, Spain, in 2015 and 2016, respectively, where he is currently pursuing the Ph.D. degree in telecommunications engineering. His research interests include optimization of radio resource management for mobile networks, location-based services and management, and data analytics.