Video Streaming and Cloud Gaming services over 4G and 5G: a complete network and service metrics dataset

Carlos Baena [1], Oswaldo Peñaherrera [2], Lourdes Camacho [2], Raquel Barco [2], and Sergio Fortes [2]

[1]University of Málaga
[2]Affiliation not available

October 30, 2023

Abstract

The enhancement reached in the telecommunication field, especially in cellular communications, has led to the emergence of multiple multimedia services. The easy supported by the mobile networks together with their novelty has triggered their popularization among society, being considered already essential in the daily life. Likewise, the increasingly connectivity dependence of these services leads to have a huge impact of the network on the user's perception, being their adequate provision one of the biggest concerns among telcos. Moreover, data encryption used for the security and integration of network information exchanged, together with the limited access to the application data, difficulties the assessment of the service's end-to-end (E2E) performance. In this scope, this work presents a dataset based on multiple metrics namely KQIs, which provide the E2E conditions of different services. Particularly, the dataset considers video streaming and cloud gaming (CG) services. Furthermore, a detailed description of the testbed where the dataset has been generated is given, discussing the reasoning behind the choices made for its development.

Video Streaming and Cloud Gaming services over 4G and 5G: a complete network and service metrics dataset

Carlos Baena, O. S. Peñaherrera-Pulla, *Member IEEE*, Lourdes Camacho, Raquel Barco, Sergio Fortes, *Member IEEE*

***Abstract*—The enhancement reached in the telecommunication field, especially in cellular communications, has led to the emergence of multiple multimedia services. The easy access supported by the mobile networks together with their novelty has triggered their popularization among society, being considered already essential in the daily life. Likewise, the increasingly connectivity dependence of these services leads to have a huge impact of the network on the user's perception, being their adequate provision one of the biggest concerns among telcos. Moreover, data encryption used for the security and integration of network information exchanged, together with the limited access to the application data, difficulties the assessment of the service's end-to-end (E2E) performance. In this scope, this work presents a dataset based on multiple metrics namely KQIs, which provide the E2E conditions of different services. Particularly, the dataset considers video streaming and cloud gaming (CG) services. Furthermore, a detailed description of the testbed where the dataset has been generated is given, discussing the reasoning behind the choices made for its development.**

***Index Terms*—Cellular communications, experimental dataset, multimedia services, E2E, video streaming, cloud gaming, testbed.**

I. Introduction

Cellular communications through their different generations have often been seen as the key to the emergence of multiple services. The increasingly advanced capabilities offered by each generation have allowed this kind of network to become the main alternative for the provision of many services. This has been reflected in the growth of traffic through these networks: according to 2022 Ericsson Mobility Report [1], global mobile data traffic volume gets around 67 EB (Exabytes) per month by the end of 2021. From this value, it is estimated that 69 percent corresponds to video traffic. Additionally, the adoption of cloud-based services, such as cloud gaming (CG) together with XT-type services that include AR (Augmented Reality), VR (Virtual Reality), and MR (Mixed Reality), will boost the volume of data exchanged through these networks, projecting a value of around 282 EB per month in 2027.

This fact makes network management to be more complex since each of the services, from a wide range of available ones, has heterogeneous requirements. Besides, the performance of the services also depends on the network condition, therefore, their quality of experience (QoE).

Thus, the research community has put a lot of effort into the design and implementation of multiple algorithms that helps management tasks, including traffic steering, load balancing, and end-to-end (E2E) Quality of Service (QoS) provision. Likewise, the introduction of the Machine Learning (ML) concept to this kind of algorithm has led to the paradigm of SON (*Self-Organizing Networks*), which follows to make the network smarter.

In this scope, the vast majority of the algorithms proposed in the literature are tested in simulators, where network scenarios are based on statistical models that follow to fit the radio environment (e.g., propagation loss, fading) presented in a real network. Here, NS-3 and Simulink toolbox (Matlab) are two of the most extended simulators used in the literature. Likewise, the emergence of SDN (*Software Defined Networks*) and NFV (*Network Function Virtualization*) has supposed the appearance of other alternatives [2]–[5], which usually provide some interesting features from 5G, such as network slicing.

Conversely, other testbeds presented in the literature such as [6]–[10] are more focused on the evaluation of multimedia services whose performance highly lies on the network. Most of them use an enhanced router to emulate different network conditions and typologies (e.g., WiFi, cellular communications) by introducing models based on standards.

Here, this work describes the testbed in [11], [12] for the evaluation of E2E performance of some popular services such as video on demand (VOD), live streaming (LS), and CG in multiple network environments, including real cellular network deployments. Moreover, some details of a framework used to complement the testbed will be given [13]. This framework allows both the interaction and gathering of information from network deployments based on Amarisoft's solutions, easing the assessment of ML-based mechanisms for SON algorithms. Likewise, we provide categorized measurement data obtained from this testbed with the aim of facilitating future research.

The rest of the paper is organized as follows. Section II introduces the general considerations of the multimedia services taken into account in this work, paying special attention to the metrics used to measure the E2E performance of each service. Based on these aspects, Section III discusses the design of the testbed. Then, Section IV describes the practical implementation of the testbed in [11]–[13]. In Section V, a detailed description of the dataset is provided. Subsequently, Section VI offers different ML applications for the presented dataset. Finally, Section VII summarizes the conclusions and future works.

The authors are with Telecommunication Research Institute (TELMA), Universidad de Málaga, E.T.S. Ingeniería de Telecomunicación, Bulevar Louis Pasteur 35, 29010, Málaga (Spain) (e-mail: {jcbg, sppulla, lcm, rbm, sfr}@ic.uma.es) *(Corresponding author: Sergio Fortes.)*

II. General considerations of multimedia services

The streaming concept refers to the distribution of multimedia content (e.g., music, video...) through a network, allowing the user to consume the content while it is being downloaded. This differs from the traditional way of delivering content that must be previously downloaded before its use. To do so, the download flow is stored in a buffer located in the user's client, from which the data is taken for their playback. Hence, this new approach to accessing multimedia content is having a huge impact not only on its consumption but also on the way content is created and distributed, leading to two main types of streaming: on-demand and live.

In streaming on demand, the content is formerly generated and saved in a server, from which the clients can play the content at any time as many times as they want. When it comes to LS, the user plays content that is generated at the moment, bearing a strong resemblance to conventional radio or TV services. Besides, LS platforms usually promote interaction between the content creator and the viewers via implemented text chats, boosting their popularity among society.

Nonetheless, streaming services require a certain data rate for their delivery, which means that the perception that users have of the services (i.e., QoE) highly depends on the network performance. In this scope, services are often assessed regarding some E2E metrics, namely KQIs (*Key Quality Indicators*), which are different concerning the type of service.

The essence of streaming services (i.e., playing the content meanwhile it is being downloaded) always puts the focus on the freezing or stalls, which is related to the temporally stop of the service (e.g., image in a video is frozen). These events are mainly triggered by the lack of data in the user's buffer, which means that new content cannot be displayed. Hence, to qualify the service's smoothness, multiple KQIs related to stalls are considered such as the number, frequency, or duration of them.

Along these lines, some protocols such as HLS (*HTTP Live Streaming)* or DASH (*Dynamic Adaptive Streaming over HTTP*) furnish the services with ABR (*Adaptive Bitrate streaming*) algorithms. As the name indicates, these algorithms follow to optimize the quality of the streaming by adapting the bitrate required to fill the buffer. Hence, the multimedia content is codified in segments in the server with different levels of quality (i.e., resolution), allowing the user at any time to request the segment that best fits with the condition of the network connectivity, and therefore, avoids freezing. In this scope, the resolution with which the content is displayed, together with the number of resolution switches during the playback are also paramount parameters to consider when assessing the service's perception.

Together with these metrics, the initial time of the playback and the video latency are other KQIs contemplated for video streaming. The former contemplates the time taken to fill the buffer with the needed data to start the video. The latter, which is more related to live streaming, refers to the time delay between the content generator and the audience, which is primordial for the interaction between both of them.

Similarly, the game industry is also focusing on another alternative in the provision of their services. Frequently, this industry finds the main bottleneck for the supply of their services in user devices: the high computational requirements for the execution of some games hinder their use among users without powerful equipment, reducing the market extension they may have, hence, losing money. Thus, in its goal of overcoming this issue, they have found a new lease of life in the CG concept.

CG paradigm takes advantage of the cloud computing approach in order to offer a gaming experience in devices with constrained computational capabilities. This means that the vast majority of tasks related to the execution of the game are now located in a remote server, therefore, unleashing computing on the user device. In this way, the CG concept can be seen as a streaming service: the server, in which the game is hosted and launched, streams the rendered content to the user device, which is only in charge of gathering their actions and displaying the multimedia content. This induces the use of some video streaming KQIs to also assess the E2E performance of CG, such as resolution and freezing. Higher resolutions provide finer image granularity, which brings players closer to increasingly realistic virtual scenarios, therefore, increasing the QoE of the service. On its behalf, frozen scenes will difficult the interaction with the game, leading to a heavy degradation of the quality of the service. Related to this interplay, new KQIs are thoughtful for the service's assessment. On the one hand, the number of images used to represent game scenes (i.e., frame rate) shows a high influence on the perception of the service, where smoother movements come with higher frame rates. On the other hand, the time taken between the user's input and its representation on the screen, widely known as input lag or E2E latency, is the primordial factor of CG services where high input lag values usually demote the E2E performance of the service.

However, the impact that CG has on the network is different: the volume of data exchanged in this kind of service is quite superior compared to video streaming. The main reason behind this lies in the interactivity offered by the service, whose latency sensitivity difficulties video compression. For example, the extended H.264 codec is used in both video streaming and CG services, albeit its bi-directional coding modes are avoided in the latter to reduce the delay. This also causes remarkable increments in the network requirements for the provision of CG services, whose bitrate of the final bitstream is far beyond the rate of on-demand video of the same quality. This means that freezing events and high input lag values will be likely to appear in networks that do not meet the ideal requirements, where both the resolution and frame rate of the streamed content also play an important role.

III. Testbed design

The main goal of the testbed is the collection of metrics from different parts of the architecture, as well as easing the execution of the services for their deployment on network benchmarks. In order to do that, the testbed is split into three main blocks: multimedia services, network adapters, and network infrastructure. Figure 1 shows the distribution of these three modules, as well as the elements, protocols, or connectivity of which they are made up.

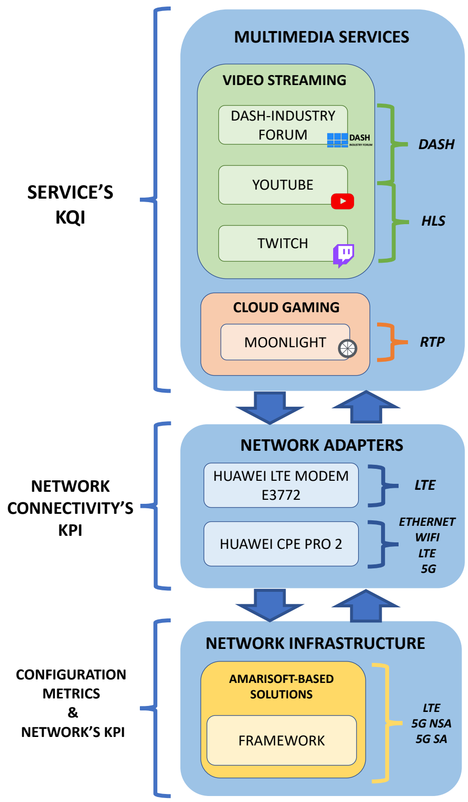

Fig. 1: Modular distribution of the testbed

A. *Multimedia Services*

In this first block, four different elements can be seen distributed between video streaming and CG services. For the case of video streaming, three platforms are considered: Dash-Industry Forum, Youtube, and Twitch. The former is an open-source streaming client, which allows customizing some parameters of the player (e.g., ABR algorithms) as well as playing any kind of DASH content located in private and public servers. Conversely, the latter two correspond to two of the most extended platforms that offer free VOD and LS content. For the case of Youtube, the platform uses DASH and HLS protocols for the provision of VOD and live streaming respectively, enabling 4K resolution playbacks. On its behalf, Twitch only bases its streamings on HLS, supporting the playback of video up to 1080p resolution. All these considered streaming alternatives settle its accessibility in web clients, easing their execution and opening different ways of data extraction.

When it comes to CG services, only Moonlight platform is considered in the testbed. This platform is an open-source implementation of the Nvidia low-latency streaming protocol, which has been specially designed over RTP (*Real-time Transport Protocol*) for the streaming of games. Thus, Moonlight enables the deployment of a private CG architecture, where only an Nvidia GPU (*Graphical Processing Unit*) is required in the server for the streaming of the content to any kind of device (e.g., laptop, smartphone, tablet). These devices only need to have installed the Moonlight client, which is available on all platforms (e.g., Windows, Linux, Android, iOS). Moreover, the client enables the configuration of the CG session, which means the set up of streaming parameters such as resolution or frame rate. All these features place Moonlight as a good alternative for the development of KQI extraction tools, thanks to the high control of the platform that it offers.

B. *Network adapters*

This module is responsible for providing different types of connection transparently to the upper module. Here, thanks to their specifications, two devices are mainly considered in the testbed: Huawei LTE Modem E3772 and Huawei CPE PRO 2. The first one is a dongle device that enables LTE connectivity through a USB port. The latter is an enhanced router with Ethernet ports and WiFi 6 connectivity. However, its most interesting feature resides in the backhaul connectivity, which ranges from wired to wireless networks, including cellular communications (i.e., LTE and 5G). Additionally, they are equipped with an API (Application Programming Interface) that eases the extraction of data from the devices, boosting the complementation of the service's KQI metrics with information of the network performance from the point of view of the UE. Nonetheless, the use of the upper layer is not restricted to these other alternatives for the provision of connectivity can be used, such as any kind of router or mobile tethering.

C. *Network infrastructure*

Although this last module follows to offer some flexibility to the testbed in terms of network connectivity, the testbed is complemented by a framework [13] for further integration of the above modules with the network. This framework is designed over the top of Amarisoft's solutions for the virtual deployment of cellular networks, offering a cheaper alternative to the common network infrastructure by taking advantage of SDR (*Software Defined Radio*). Therefore, the goal of this block is to ease the implementation of network actions, such as the setup of different network configurations or the collection of information, achieving in this way a full control of the service's deployment.

IV. Testbed Implementation

This section describes several considerations to deploy the testbed described in the previous section. Thus, the methodology and techniques for the implementation of the different modules are exposed. Complementary, an overview of the most interesting metrics available gathered by the testbed will be provided.

A. *Service's KQI extraction*

For the collection of the KQI metrics for each service, two primordial points are considered. Firstly, the extraction of these metrics must be over the top of the service to get accurate values. Consequently, the aim is to obtain these metrics as close to the user as possible as depicted in Figure 2. Secondly, the metrics must be obtained in an automatic way to ease the execution of multiple experiments, hence, launch campaigns. This also means automating service execution (e.g., the repeated playback of a specific video or the configuration and playability with a game). To achieve that, two different approaches have been followed regarding the type of service.

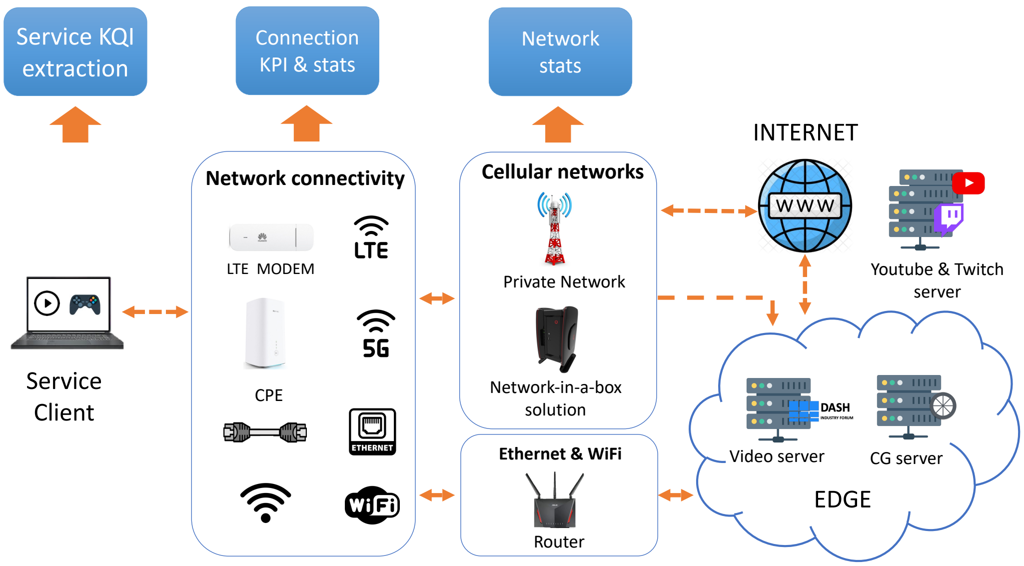

Fig. 2: Physical architecture of the testbed

1) Video streaming: Most video streaming platforms provide web clients to consume their content. Besides, some metrics related to the performance of the service are accessible through the advanced configuration of these web clients. Along these lines, the implementation of the video streaming block lies on the *web scraping* methodology. This technique allows the inspection and interaction with multiple web elements, easing the extraction of information from websites. Thus, the block consists in several Python scripts that base their actions in the *Selenium WebDriver*, which acts as a link with the web browser. That is to say that the *WebDriver* provides the capability of carrying out different actions with the browser, such as the initialization of the service or the selection of a certain video resolution (i.e., disabling ABR algorithm). In addition, it allows accessing and processing parameters from the player such as the resolution displayed or buffer health, as well as the calculation of other parameters, such as the initial time. Table I shows the primordial metrics obtained from the services, while more player and platform-specific metrics are obtained.

2) Cloud Gaming: Although the service can be considered partly video streaming, the approach for the metrics collection must be different. Firstly, the considered CG platform (i.e., Moonlight) requires the installation of lightweight software, precluding the adoption of *web scrapping* technique. Furthermore, the experience of this service highly lies on the interaction of the users, introducing complexity in the recollection process. In order to solve that, the Python scripts that compose the module for this service follow three main steps: emulation of user actions, perception tracking and metrics calculation.

The former pursuits the interaction with the client for the game configuration, as well as the replication of some prerecorded user actions during the gameplay. This enables the possibility of carrying out multiple experiments with the same input actions, hence, giving objectivity to the results obtained in each test. Additionally, it is also in charge of gathering the timestamp of each action, considered a key factor in calculating the interactivity of the service. Regarding perception tracking, the scripts are responsible for detecting the result of the user's action and finding the exact time when it has been perceived by the user. Likewise, they record the session with a high frame rate (i.e., 144 FPS) for the further calculation of the service's KQIs.

Once all these steps have been fulfilled, the KQIs of the service are calculated from the correlation of the data obtained in the previous steps. For example, timestamps of automated actions are mapped with the time in which the scene has changed, thus, obtaining the latency or input lag of the system in detail (i.e., latency for each action in the client and server). Furthermore, other required actions such as frame decimation are carried out in this step for the accurate calculation of some metrics like the freeze events or the effective frame rate (EFPS) perceived by the user. Particularly for these metrics, frame decimation is required to fit the frame rate of the recording with the configured one in the CG server for streaming. In this way, frame decimation eases the abolition of false positives freeze events or the miscalculation of the EFPS value.

Last but not least, this information is complemented with some performance data from the platform, which is obtained by the inspection and parsing of the logs provided by the platform itself. Table I reveals some of the parameters of interest for the assessment of the services obtained through the testbed.

B. Network connectivity adapters

Even though any network adapter can be used for enabling the execution of the services (see Figure 2), the available API in Huawei devices makes their consideration in the testbed. In this way, the Python functions that fit in this block are in charge of wrapping the different HTTP requests needed for interacting with the API, easing the gathering of information related to radio connectivity, such as RSRP (*Reference Signal Received Power*), RSRQ (*Reference Signal Received Quality*) or SINR (*Signal to Interference & Noise Ratio*), and values

TABLE I: Testbed parameters overview

Source	Type	Metrics	Description
Service	VOD	Initial time	Time required to start the video.
	VOD & LS	Resolution displayed	Resolution with which the video is being delivered to the user.
		Buffer health	Amount of video stored in the client's buffer.
		Frame per Seconds (FPS)	Frame rate with which the video is being displayed.
		Frames dropped	Number/ratio of frames which are dropped due to network issues.
		Freeze events	Number/duration/ratio of frozen events during the video playback.
	LS	Video latency	Latency between the content generator and the video playback.
	CG	Incoming frame rate	Avg. estimated number of frames received in the thin client network interface.
		Decoding frame rate	Avg. number of decoded rames in the client.
		Rendering frame rate	Avg. number of rendered frames in the client.
		Frames dropped by network	Avg. percentage of lost frames in the transport process due to network errors.
		Frames dropped due to jitter	Avg. percentage of lost frames due to jitter.
		Average receive time	Avg. time to encode a frame since the server sends the first packet.
		Average decoding time	Avg. time that a reassembled frame needs to be decoded in the client.
		Average rendering time	Avg. time to render and represent a decoded frame.
		Average frame queue delay	Avg. time that a decoded frame awaits in the queue before being rendered.
		Client latency	E2E latency or input lag of the Cloud Gaming system.
		Host latency	Input lag of the server which hosts the game.
		Number/duration of stalls	Number and duration of frozen events in the session.
		EFPS	Actual frame rate displayed on the client's screen.
Network Adapter	Radio metrics	RSRP	Power lineal average of all element that carry reference signals.
		RSRQ	Information about the interference and strength of the desired signal.
		RSSI	Total power received by the UE in all the frequency band.
		SINR	Relationship between signal and noise level.
		PPUSCH	Power measured in Physical Uplink Shared Channel (PUSCH).
		PPUCCH	Power measured in Physical Uplink Control Channel (PUCCH).
		PSRS	Power measured in Sounding Reference Signal (SRS).
		PRACH	Power measured in Physical Random Access Channel (RACH).
		CQI	Indicator sent by UE carrying the information on the channel quality.
		MCS	Code that defines the modulation scheme used to transmit data.
	Exchanged data	DL/UL Bitrate	Data rate with which the device has sent/received data.
		Data volume	Total amount of data exchanged (UL/DL) with the device.
Network Framework	Radio statistics	Users scheduled stats	Min/Avg/Max number of users scheduled in DL/UL in the cell.
		RRC counters	Classic counters related to Radio Resource Control layer (e.g. number of requests).
		DL/UL Bitrate	Uplink and Downlink bitrate of the cell measured at MAC layer.
		DL/UL Retransmission	Number of retransmitted frames in the cell by each UE.
	Configuration	Configuration parameters	Parameters used in Amarisoft's software for the RAN/CORE deployment.

from the data exchanged through the network adapter, such as data rates or data volumes.

Thus, the repeated and multiple calls in the background of these functions enable the monitoring of the network adapter device whilst the service is running, positioning as another source of information for the service assessment. Table I summarizes, among others, all available metrics collected from the network adapter.

C. Amarisoft-based solutions framework

This framework, presented in [13], is set up over Amarisoft's software solutions for the virtual deployment of 4G and 5G networks. Its main goal is to endow the testbed with some kind of interaction with the network infrastructure that provides Internet connectivity. In this sense, the implementation of the framework is divided according to the goal of the interaction: configuration, monitoring, and action with the network entities.

When it comes to configuration, a set of functions is in charge of reading and parsing the configuration files of the multiple elements that conform the network (e.g., eNB/gNB, MME, AMF...), enabling the gathering and manipulation of their values through JSON (*JavaScript Object Notation*) syntax. With regards to monitoring case, the module makes multiple calls to an API provided by the device itself, also wrapping both the request and response in JSON to manage them as a human-readable format.

Table I shows the most relevant metrics obtained from this framework that can be useful for the service assessment. Furthermore, the framework offers a set of functions to perform actions in the different processes that compose the cellular network such as the deployment of predefined virtual entities.

Finally, all these blocks are orchestrated by a management element, which equips the framework with a REST (*Representational State Transfer*) interface, therefore, easing the interaction through HTTP (*HyperText Transfer Protocol*) requests. Like the previous block, the use of this framework is not mandatory for the execution of the service's KQI extraction block, albeit their wide configuration availability as well as the accessibility to different network metrics make it useful.

V. Dataset

This section describes the format, organization an overview of the provided dataset. Furthermore, some important aspects of these measurements are described in order to ease its understanding and processing.

A. Dataset overview

The dataset is made up of 8461 samples corresponding to 2784 samples of VOD service, 1933 of live streaming and 3694 CG KQI samples. From these samples, 5413 have been gathered by using LTE connectivity (with different configurations and radio conditions) and 1344 of them have been carried out under 5G networks. The rest of them have been collected using Ethernet and WiFi connections. Table II provides detailed information about the distribution of the data across services and technologies.

Conversely, in order to make the dataset consistent in terms of network impact, all the samples have been collected under the same service context. This means that the service has been launched with the same configuration and actions in every execution regarding its type:

1) VoD: The same video is played multiple times during one or two minutes, freeing the cache memory between each playback for the correct gathering of some KQIs like the initial time.

2) LS: Since in this service the content is generated on live, the measurement campaign has been carried out by displaying multiple times the same content broadcast during 1 minute. Nonetheless, the dataset presents different campaigns which consider different broadcasting sources to also behold the impact of the service by considering the broadcaster.

3) CG: For this case, multiple *League of Legends* sessions are launched, in which five equal mouse actions are performed throughout the experiments in order to keep consistency in the measurements. Besides, there is always a 5-seconds time gap between each action to simplify the gathering process.

Last but not least, all the data is distributed in different folders, whose names correspond with the service and technology used for the data generation. This information is also part of the name of the files together with some information about the measurement campaign (e.g., number of PRBs, noise level, or antenna gain values configured for the tests).

B. Dataset format and organization

All the information that composes the dataset is stored in multiple files corresponding to the measurement campaign. These files follow a JSON sintax, which enables the saving of nested data variables as well as offers a human-readable overview of the raw files that contain the data. Moreover, it is a compatible format with multiple data science libraries, easing the analysis of the data.

Regarding the organization of these files, they are composed by four main fields: *pingTest*, *service*, *networkAdapter* and *network*. Here, each label corresponds to the source or kind of data that can be found in each field.

TABLE II: Distribution of dataset samples across services and technologies

Service	**Platform**	**Technology**			
		Ethernet	**WiFi**	**LTE**	**5G**
VOD	Dash Forum	0	0	1594	0
	Youtube	40	0	600	600
LS	Twitch	258	52	1359	264
CG	Moonlight	420	934	1860	480

1) pingTest: This field contains the results of multiple ping tests performed during the campaign, considering different objects based on the destination of the ping. Each of these objects is a list with the results of the ping for each global experiment. Besides, a special notation has been followed in order to remark on the condition of the test, being added a "_" suffix in the name of each object (e.g., *ping_dns_*) if the results were obtained without running other tests. Conversely, the absence of this suffix (e.g., *ping_dns*) indicates that the tests were carried out while the service was running.

2) service: This field is mainly composed by one object labeled as *metrics*, which contains all the information gathered from the service in a list. Since this information will depend on the service tested, these metrics are supplemented by the type of service and platform used. It is worth mentioning that this field stores more data about the player and platform than the primordial information which is exposed in Table I.

3) networkAdapter: For this case, the information is distributed as shown in Table I, which means two objects regarding the radio metrics (labelled as *radioKPI*) and exchanged data information (labelled as *stats*). It must be taken into account that all the objects of the field can be empty values if another network adapter is used.

4) network: Like in the case of *networkAdapter*, the information in this field is categorized following the distribution seen in Table I. Nevertheless, the former (i.e., radio statistics) is divided into cell and UE stats, corresponding to the value's level of specification. On its behalf, the configuration parameters provide the values of all the parameters with which the network has been deployed through the Amarisoft's software. This information can be found as an object or as a list of objects, regarding if several network configuration parameters have been used during the campaign. Note that *network* field can be empty, which means that another cellular network solution/infrastructure has been used for connectivity.

VI. ML applications

The provided dataset can be used in multiple ML applications. Our aim here is to support the research community in cellular management in different areas saving the difficult and time-consuming task of data acquisition.

Among the main ML fields of application of the dataset, the training of classification and clustering models is envisaged to support users/session categorization based on available network metrics or the network anomaly detection based on E2E metrics. Conversely, E2E metrics from services and network can be mapped by using regression techniques, enabling the estimation of these metrics from network parameters as proposed in [12] and [14]. This approach can also boost the development of ML techniques for network management optimization purposes such as policy definition and network configuration setting to offer services with specific requirements under constrained resources, such as the ones preliminarily envisaged in [11] and [13].

VII. Conclusion

This paper has presented a dataset of E2E metrics from some popular and extended services such as video streaming and CG over different network technologies such as Ethernet, WiFi, 4G and 5G. Thus, the paramount features of these services have been highlighted, offering a baseline that allows boosting the comprehension and analysis of the provided data. Furthermore, the tested used for the creation of the dataset is described, putting special attention to its design and implementation. In this sense, this work tries to encourage the research community in two aspects. First, the provided data can boost the design and implementation of new algorithms, such as SON algorithms, which can support the service provision over commercial networks. Finally, the technical considerations described in the paper can inspire the development of new testbeds that consider some improvements or new implementations for data mining of cutting-edge service such as VR.

Acknowledgments

This work has been partially funded by Ministerio de Asuntos Económicos y Transformación Digital and European Union - NextGenerationEU within the framework "Recuperación, Transformación y Resiliencia y el Mecanismo de Recuperación y Resiliencia" under the project MAORI; by Ministerio de Ciencia e Innovación (grant FPU19/04468); and by Junta de Andalucía with predoctoral grant (Ref. PRE-DOC_01712).

References

[1] Ericsson, "Ericsson Mobility Report (June 2022)," https://www.ericsson.com/en/reports-and-papers/mobility-report, (accessed September 2022).

[2] A. Esmaeily, K. Kralevska, and D. Gligoroski, "A cloud-based SDN/NFV testbed for end-to-end network slicing in 4G/5G," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2020.

[3] N. Nikaein, C.-Y. Chang, and K. Alexandris, "Mosaic5G: Agile and flexible service platforms for 5G research," *ACM SIGCOMM Computer Communication Review*, vol. 48, no. 3, pp. 29–34, 2018.

[4] G. Garcia-Aviles, M. Gramaglia, P. Serrano, and A. Banchs, "POSENS: A practical open source solution for end-to-end network slicing," *IEEE Wireless Communications*, vol. 25, no. 5, pp. 30–37, 2018.

[5] T. Dreibholz, "Flexible 4G/5G testbed setup for mobile edge computing using openairinterface and open source mano," in *Workshops of the international conference on advanced information networking and applications*. Springer, 2020, pp. 1143–1153.

[6] G. Gómez and et al., "A Real-Time End-to-End Testbed for Evaluating the Performance of Multimedia Services," in *Future Multimedia Networking*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 212–217.

[7] D. Renzel, R. Klamma, and M. Spaniol, "MobSOS - A Testbed for Mobile Multimedia Community Services," in *2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, 2008, pp. 139–142.

[8] B. Lewcio and S. Möller, "A testbed for QoE-based multimedia streaming optimization in heterogeneous wireless networks," in *2011 5th International Conference on Signal Processing and Communication Systems (ICSPCS)*, 2011, pp. 1–9.

[9] C.-Y. Huang, K.-T. Chen, D.-Y. Chen, H.-J. Hsu, and C.-H. Hsu, "GamingAnywhere: The first open source cloud gaming system," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 10, no. 1s, pp. 1–25, 2014.

[10] H. Ahmadi, M. R. Hashemi, and S. Shirmohammadi, "An Open Source Cloud Gaming Testbed Using DirectShow," in *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2015, pp. 606–610.

[11] C. Baena, S. Fortes, E. Baena, and R. Barco, "Estimation of Video Streaming KQIs for Radio Access Negotiation in Network Slicing Scenarios," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1304–1307, 2020.

[12] O. S. Peñaherrera-Pulla, C. Baena, S. Fortes, E. Baena, and R. Barco, "Measuring Key Quality Indicators in Cloud Gaming: Framework and Assessment Over Wireless Networks," *Sensors*, vol. 21, no. 4, 2021.

[13] C. Baena, S. Fortes, O. Penaherrera, and R. Barco, "A Framework to boost the potential of network-in-a-box solutions," *Proceedings of the 2021 12th International Conference on Network of the Future*, 2021.

[14] A. Herrera-Garcia, S. Fortes, E. Baena, J. Mendoza, C. Baena, and R. Barco, "Modeling of Key Quality Indicators for End-to-End Network Management: Preparing for 5G," *IEEE Vehicular Technology Magazine*, vol. 14, no. 4, pp. 76–84, 2019.

Author Information

Carlos Baena (M.Sc.) received his M.Sc. degree in telematic engineering from the University of Málaga, Spain, in 2018, where he is currently pursuing a Ph.D. degree. Since 2018, he has worked as a research assistant with the Department of Communications Engineering, University of Málaga. His research interest include Quality of Experience of multimedia services, machine learning and radio resource management for mobile networks.

O.S. Peñaherrera-Pulla received his Engineering degree from the Universidad Politécnica Salesiana of Ecuador in 2018 and his M.Sc. degree in Telematics from the University of Málaga, Spain, in 2020. Currently he is pursuing a Ph.D degree and works as a research assistant at the Department of Communications Engineering of the University of Málaga. His research interests include monitoring and optimization of Quality of Experience for new-generation multimedia services and machine learning mechanisms for Radio Resource management.

Lourdes Camacho (M.Sc.) received his B.Sc and M.Sc degree in telematic engineering from de University of Málaga, Spain in 2019 and 2022 respectively.

Raquel Barco is Full Professor in Telecommunication Engineering at the University of Málaga. Before joining the university, she worked at Telefonica and at the European Space Agency (ESA). As researcher she is specialized in mobile communication networks and smart-cities, having led projects funded by several million euros, published more than 100 papers in high impact journals and conferences, authored 5 patents and received several research awards.

Sergio Fortes (Ph.D.) holds a M.Sc. and a Ph.D. in Telecommunication Engineering from the University of Málaga. He began his career being part of main european space agencies (DLR, CNES, ESA) and Avanti Communications plc, where he participated in various research and consultant activities on broadband and aeronautical satellite communications. In 2012, he joined the University of Málaga, where his research is focused on Self-Organizing Networks for cellular communications.