

## Research Article

# QoE Evaluation: The TRIANGLE Testbed Approach

Almudena Díaz Zayas <sup>1</sup>, Laura Panizo <sup>1</sup>, Janie Baños <sup>2</sup>,  
Carlos Cárdenas <sup>2</sup> and Michael Dieudonne <sup>3</sup>

<sup>1</sup>University of Malaga, Spain

<sup>2</sup>DEKRA, Spain

<sup>3</sup>Keysight Technologies, Belgium

Correspondence should be addressed to Almudena Díaz Zayas; [adz@uma.es](mailto:adz@uma.es)

Received 11 October 2018; Accepted 25 November 2018; Published 18 December 2018

Academic Editor: Giovanni Stea

Copyright © 2018 Almudena Díaz Zayas et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents the TRIANGLE testbed approach to score the Quality of Experience (QoE) of mobile applications, based on measurements extracted from tests performed on an end-to-end network testbed. The TRIANGLE project approach is a methodology flexible enough to generalize the computation of the QoE for any mobile application. The process produces a final TRIANGLE mark, a quality score, which could eventually be used to certify applications.

## 1. Introduction

The success of 5G (the fifth generation of mobile communications), and to some extent that of 4G, depends on its ability to seamlessly deliver applications and services with good Quality of Experience (QoE). Along with the user, QoE is important to network operators, product manufacturers (both hardware and software), and service providers. However, there is still no consensus on the definition of QoE, and a number of acronyms and related concepts (e.g., see [1]) add confusion to the subject: QoE (Quality of Experience), QoS (Quality of Service), QoS<sub>D</sub> (Quality of Service Delivered/achieved by service provider), QoS<sub>E</sub> (Quality of Service Experienced/Perceived by customer/user), and so forth. This is a field in continuous evolution, where methodologies and algorithms are the subject of study of many organisations and standardization bodies such as the ITU-T.

TRIANGLE project has adopted the definition of QoE provided by the ITU-T in Recommendation P.10/G.100 (2006) Amendment 1 “Definition of Quality of Experience (QoE)” [2].

*“The overall acceptability of an application or service, as perceived subjectively by the end-user”*

In [2], the ITU-T emphasizes that the Quality of Experience includes the complete end-to-end system effects: client

(app), device, network, services infrastructure, and so on. Therefore, TRIANGLE brings in a complete end-to-end network testbed and a methodology for the evaluation of the QoE.

Consistent with the definition, the majority of the work in this area has been concerned with subjective measurements of experience. Typically, users rate the perceived quality on a scale, resulting in the typical MOS (Mean Opinion Score). Even in this field, the methodology for subjective assessment is the subject of many studies [3].

However, there is a clear need to relate QoE scores to technical parameters that can be monitored and whose improvement or worsening can be altered through changes in the configurations of the different elements of the end-to-end communication channel. The E-model [4], which is based on modelling the results from a large number of subjective tests done in the past on a wide range of transmission parameters, is the best-known example of parametric technique for the computation of QoE. Also, one of the conclusions of the Project P-SERQU, conducted by the NGMN (Next Generation Mobile Networks) [5] and focused on the QoE analysis of HTTP Adaptive Streaming (HAS), is that it is less complex and more accurate to measure and predict QoE based on traffic properties than making a one-to-one mapping between generic radio and core network QoS to QoE. The TRIANGLE

project follows also a parametric approach to compute the QoE.

Conclusions in [5] point out that a large number of parameters in the model could be cumbersome due to the difficulty of obtaining the required measurements and because it would require significantly more data points and radio scenarios to tune the model. The TRIANGLE approach has overcome this limitation through the large variety of measurements collected, the variety of end-to-end network scenarios designed, and mostly the degree of automation reached, which enables the execution of intensive test campaigns covering all scenarios.

Although there are many proposals to calculate the quality of experience, in general, they are very much oriented to specific services, for example, voice [6] or video streaming [7, 8]. This paper introduces a methodology to compute the QoE of any application, even if the application supports more than one service.

The QoE, as perceived by the user, depends on many factors: the network conditions, both at the core (CN) and at the radio access (RAN), the terminal, the service servers, and human factors difficult to control. Due to the complexity and the time needed to run experiments or make measurements, most of the studies limit the evaluation of the QoE to a limited set of, or even noncontrolled, network conditions, especially those that affect the radio interface (fading, interference, etc.). TRIANGLE presents a methodology and a framework to compute the QoE, out of technical parameters, weighting the impact of the network conditions based on the actual uses cases for the specific application. As in ITU recommendation G1030 [9] and G1031 [10], the user's influence factors are outside of the scope of the methodology developed in TRIANGLE.

TRIANGLE has developed an end-to-end cellular network testbed and a set of test cases to automatically test applications under multiple changing network conditions and/or terminals and provide a single quality score. The score is computed weighting the results obtained testing the different uses cases applicable to the application, for the different aspects relevant to the user (the domains in TRIANGLE), and under the network scenarios relevant for the application. The framework allows specific QoS-to-QoE translations to be incorporated into the framework based on the outcome of subjective experiments on new services.

Note that although the TRIANGLE project also provides means to test devices and services, only the process to test applications is presented here.

The rest of the paper is organized as follows. Section 2 provides an overview of related work. Section 3 presents an overview of the TRIANGLE testbed. Section 4 introduces the TRIANGLE approach. Section 5 describes in detail how the quality score is obtained in the TRIANGLE framework. Section 6 provides an example and the outcome of this approach applied to the evaluation of a simple App, the Exoplayer. Finally, Section 7 summarizes the conclusions.

## 2. State of the Art

Modelling and evaluating QoE in current and next generation of mobile networks is an important and active research

area [8]. Different types of testbeds can be found in the literature, ranging from simulated to emulated mobile/wireless testbeds, which are used to obtain subjective or objective QoE metrics, to extract a QoE model, or to assess the correctness of a previously generated QoE model. Many of the testbeds reviewed have been developed for a specific research, instead of for a more general purpose, such as the TRIANGLE testbed, which can serve a wide range of users (researchers, app developers, service providers, etc.). In this section, some QoE-related works that rely on testbeds are reviewed.

The QoE Doctor tool [12] is closely related to the TRIANGLE testbed, since its main purpose is the evaluation of mobile apps QoE in an accurate, systematic, and repeatable way. However, QoE Doctor is just an Android tool that can take measurements at different layers, from the app user interface (UI) to the network, and quantify the factors that impact the app QoE. It can be used to identify the causes of a degraded QoE, but it is not able to control or monitor the mobile network. QoE Doctor uses an UI automation tool to reproduce user behaviour in the terminal (app user flows in TRIANGLE nomenclature) and to measure the user-perceived latency by detecting changes on the screen. Other QoE metrics computed by QoE Doctor are the mobile data consumption and the network energy consumption of the app by means of an offline analysis of the TCP flows. The authors have used QoE Doctor to evaluate the QoE of popular apps such as YouTube, Facebook, or mobile web browsers. One of the drawbacks of this approach is that most metrics are based on detecting specific changes on the UI. Thus, the module in charge of detecting UI changes has to be adapted for each specific app under test.

QoE-Lab [13] is a multipurpose testbed that allows the evaluation of QoE in mobile networks. One of its purposes is to evaluate the effect of new network scenarios on services such as VoIP, video streaming, or web applications. To this end, QoE-Lab extends BERLIN [14] testbed framework with support for next generation mobile networks and some new services, such as VoIP and video streaming. The testbed allows the study of the effect of network handovers between wireless technologies, dynamic migrations, and virtualized resources. Similar to TRIANGLE, the experiments are executed in a repeatable and controlled environment. However, in the experiments presented in [13], the user equipment were laptops, which usually have better performance and more resources than smartphones (battery, memory, and CPU). The experiments also evaluated the impact of different scenarios on the multimedia streaming services included in the testbed. The main limitations are that it is not possible to evaluate different mobile apps running in different smartphones or relate the QoE with the CPU, battery usage, and so forth.

De Moor et al. [15] proposed a user-centric methodology for the multidimensional evaluation of QoE in a mobile real-life environment. The methodology relies on a distributed testbed that monitors the network QoS and context information and integrates the subjective user experience based on real-life settings. The main component of the proposed architecture is the *Mobile Agent*, a component to be installed in the user device that monitors contextual data (location,

velocity, on-body sensors, etc.) and QoS parameters (CPU, memory, signal strength, throughput, etc.) and provides an interface to collect user experience feedback. A processing entity receives the (device and network) monitored data and analyzes the incoming data. The objective of this testbed infrastructure is to study the effects of different network parameters in the QoE in order to define new estimation models for QoE.

In [16], the authors evaluated routing protocols BATMAN and OLSR to support VoIP and video traffic from a QoS and QoE perspective. The evaluation took place by running experiments in two different testbeds. First, experiments were run in the Omnet++ simulator using the InetManet framework. Second, the same network topology and network scenarios were deployed in the Emulab test bench, a real (emulated) testbed, and the same experiments were carried out. Finally, the results of both testbeds (simulated and real-emulated) were statistically compared in order to find inconsistencies. The experiments in the simulated and emulated environments showed that BATMAN achieves better than OLSR and determined the relation between different protocol parameters and their performance. These results can be applied to implement network nodes that control in-stack protocol parameters as a function of the observed traffic.

In [17], a testbed to automatically extract a QoE model of encrypted video streaming services was presented. The testbed includes a software agent to be installed in the user device, which is able to reproduce the user interaction and collect the end-user application-level measurements; the network emulator NetEm, which changes the link conditions emulating the radio or core network, and a Probe software, which processes all the traffic at different levels, computes the TCP/IP metrics, and compares the end-user and network level measurements. This testbed has been used to automatically construct the model (and validate the model) of the video performance of encrypted YouTube traffic over a Wi-Fi connection.

More recently, in [18], Solera et al. presented a testbed for evaluating video streaming services in LTE networks. In particular, the QoE of 3D video streaming services over LTE was evaluated. The testbed consists of a streaming server, the NetEm network emulator, and a streaming client. One of the main contributions of the work is the extension of NetEm to better model the characteristics of the packet delay in bursty services, such as video streaming. Previously to running the experiments in the emulation-based testbed, the authors carried out a simulation campaign with an LTE simulator to obtain the configuration parameters of NetEm for four different network scenarios. These scenarios combine different positions of the user in the cell and different network loads. From the review of these works, it becomes clear that the setup of a simulation or emulation framework for wireless or mobile environments requires, in many cases, a deep understanding of the network scenarios. TRIANGLE aims to reduce this effort by providing a set of preconfigured real network scenarios and the computation of the MOS in order to allow both researchers and app developers to focus on the evaluation of new apps, services, and devices.

### 3. TRIANGLE

The testbed, the test methodology, and the set of test cases have been developed within the European funded TRIANGLE project. Figure 1 shows the main functional blocks that make up the TRIANGLE testbed architecture.

To facilitate the use of the TRIANGLE testbed for different objectives (testing, benchmarking, and certifying), to remotely access the testbed, and to gather and present results, a web portal, which offers an intuitive interface, has been implemented. It provides access to the testbed hiding unnecessary complexity to App developers. For advanced users interested in deeper access to configuration parameters of the testbed elements or the test cases, the testbed offers a direct access to the Keysight TAP (Testing Automation Platform), which is a programmable sequencer of actions with plugins that expose the configuration and control of the instruments and tools integrated into the testbed.

In addition to the testbed itself, TRIANGLE has developed a test methodology and has implemented a set of test cases, which are made available through the portal. To achieve full test case automation, all the testbed components are under the control of the testbed management framework, which coordinates their configuration and execution, processes the measurements made in each test case, and computes QoE scores for the application tested.

In addition, as part of the testbed management framework, each testbed component is controlled through a TAP driver, which serves as bridge between the TAP engine and the actual component interface. The configuration of the different elements of the testbed is determined by the test case to run within the set of test cases provided as part of TRIANGLE or the customized test cases built by users. The testbed translates the test cases specific configurations, settings, and actions into TAP commands that take care of commanding each testbed component.

TRIANGLE test cases specify the measurements that should be collected to compute the KPI (Key Performance Indicators) of the feature under test. Some measurements are obtained directly from measurement instruments but others require specific probes (either software or hardware) to help extract the specific measurements. Software probes, running on the same device (UE, LTE User Equipment) that the application under test, include DEKRA Agents and the TestelDroid [19] tool from UMA. TRIANGLE also provides an instrumentation library so that app developers can deliver measurement outputs, which cannot otherwise be extracted and must be provided by the application itself. Hardware probes include a power analyzer connected to the UE to measure power consumption and the radio access emulator that, among others, provides internal logs about the protocol exchange and radio interface low layers metrics.

The radio access (LTE RAN) emulator plays a key role in the TRIANGLE testbed. The testbed RAN is provided by an off-the-shelf E7515A UXM Wireless Test Set from Keysight, an emulator that provides state-of-the-art test features. Most important, the UXM also provides radio channel emulation for the downlink radio channel.

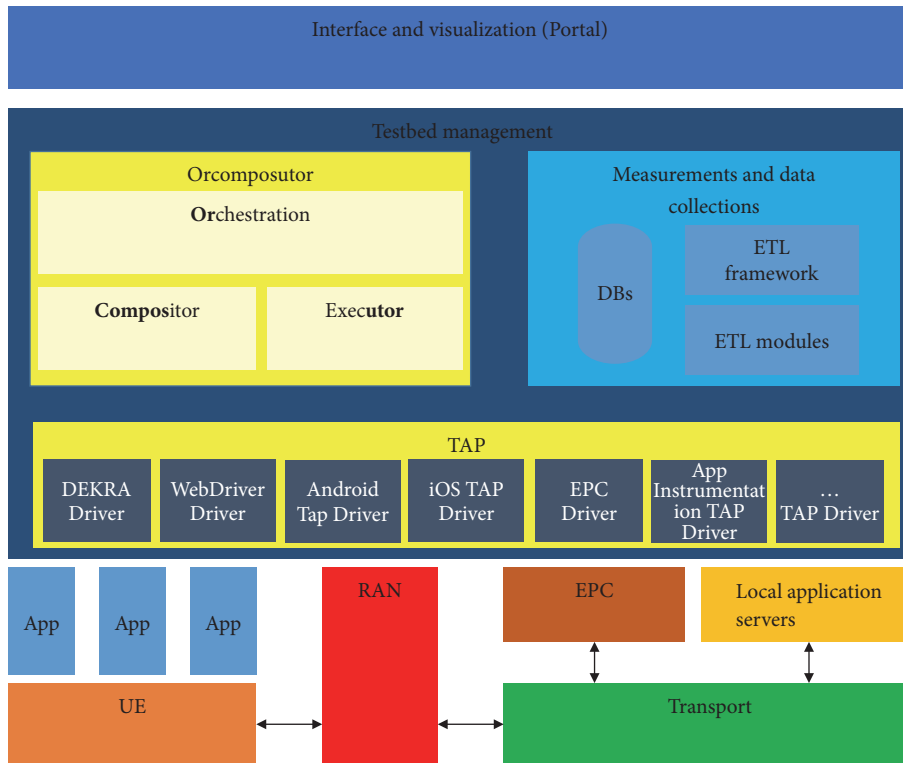


FIGURE 1: TRIANGLE testbed architecture.

In order to provide an end-to-end system, the testbed integrates a commercial EPC (LTE Evolved Packet Core) from Polaris Networks, which includes the main elements of a standard 3GPP compliant LTE core network, that is, MME (Mobility Management Entity), SGW (Serving Gateway), PGW (Packet Gateway), HSS (Home Subscriber Server), and PCRF (Policy and Charging Rules Function). In addition, this EPC includes the EPDNG (Evolved Packet Data Gateway) and ANDSF (Access Network Discovery and Session Function) components for dual connectivity scenarios. The RAN emulator is connected to the EPC through the standard S1 interface. The testbed also offers the possibility of integrating artificial impairments in the interfaces between the core network and the application servers.

The Quamotion WebDriver, another TRIANGLE element, is able to automate user actions on both iOS and Android applications whether they are native, hybrid, or fully web-based. This tool is also used to prerecord the app's user flows, which are needed to automate the otherwise manual user actions in the test cases. This completes the full automation operation.

Finally, the testbed also incorporates commercial mobile devices (UEs). The devices are physically connected to the testbed. In order to preserve the radio conditions configured at the radio access emulator, the RAN emulator is cable connected to the mobile device antenna connector. To accurately measure the power consumption, the N6705B power analyzer directly powers the device. Other measurement instruments may be added in the future.

#### 4. TRIANGLE Approach

The TRIANGLE testbed is an end-to-end framework devoted to testing and benchmarking mobile applications, services, and devices. The idea behind the testing approach adopted in the TRIANGLE testbed is to generalize QoE computation and provide a programmatic way of computing it. With this approach, the TRIANGLE testbed can accommodate the computation of the QoE for any application.

The basic concept in TRIANGLE's approach to QoE evaluation is that the quality perceived by the user depends on many aspects (herein called domains) and that this perception depends on its targeted use case. For example, battery life is critical for patient monitoring applications but less important in live streaming ones.

To define the different 5G uses cases, TRIANGLE based its work in the Next Generation Mobile Network (NGMN) Alliance foundational White Paper, which specifies the expected services and network performance in future 5G networks [20]. More precisely, the TRIANGLE project has adopted a modular approach, subdividing the so-called "NGMN Use-Cases" into blocks. The name *Use Case* was kept in the TRIANGLE approach for describing the application, service, or vertical using the network services. The diversification of services expected in 5G requires a concrete categorization to have a sharp picture of what the user will be expected to interact with. This is essential for understanding which aspect of the QoE evaluation needs to be addressed. The final use cases categorization was defined in [11] and encompasses both the services normally accessible via mobile

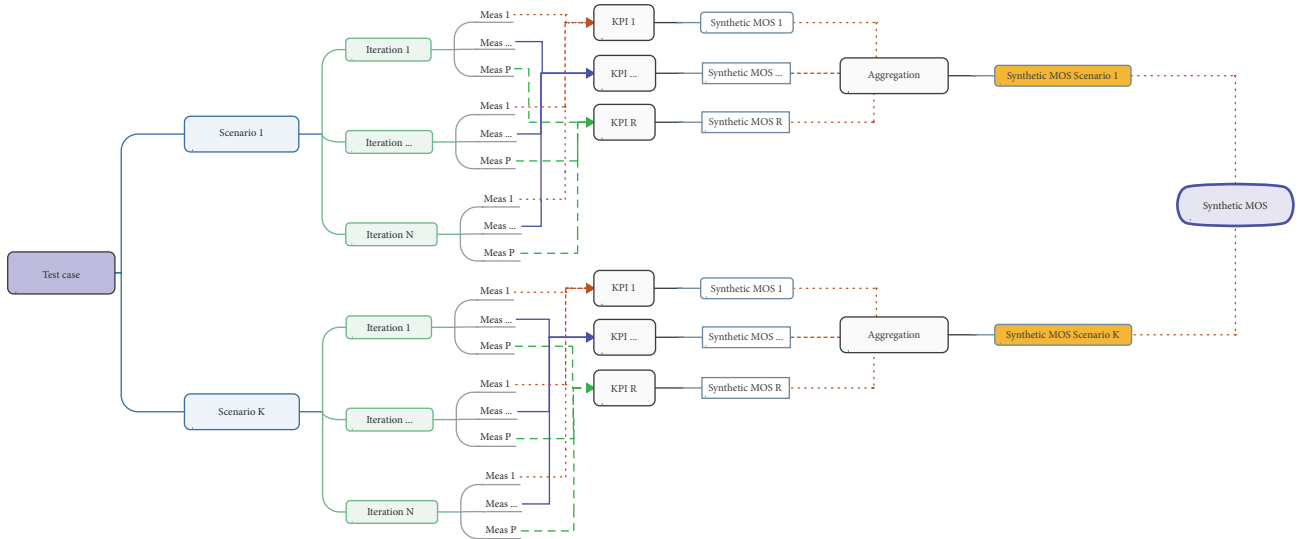


FIGURE 2: The process to obtain the *synthetic-MOS* score in a TRIANGLE test case.

TABLE 1: Uses cases defined in the TRIANGLE project.

Identifier	Use Case
VR	Virtual Reality
GA	Gaming
AR	Augmented Reality
CS	Content Distribution Streaming Services
LS	Live Streaming Services
SN	Social Networking
HS	High Speed Internet
PM	Patient Monitoring
ES	Emergency Services
SM	Smart Metering
SG	Smart Grids
CV	Connected Vehicles

phones (UEs) and the ones that can be integrated in, for example, gaming consoles, advanced VR gear, car units, or IoT systems.

The TRIANGLE domains group different aspects that can affect the final QoE perceived by the users. The current testbed implementation supports three of the several domains that have been identified: Apps User Experience (AUE), Apps Energy consumption (AEC), and Applications Device Resources Usage (RES).

Table 1 provides the use cases and Table 2 lists the domains initially considered in TRIANGLE.

To produce data to evaluate the QoE, a series of test cases have been designed, developed, and implemented to be run on the TRIANGLE testbed. Obviously, not all test cases are applicable to all applications under test, because not all applications need, or are designed, to support all the functionalities that can be tested in the testbed. In order to automatically determine the test cases that are applicable to an application under test, a questionnaire (identified as

features questionnaire in the portal), equivalent to the classical conformance testing ICS (Implementation Conformance Statement), has been developed and is accessible through the portal. After filling the questionnaire, the applicable test plan, that is, the test campaign with the list of applicable test cases, is automatically generated.

The sequence of user actions (type, swipe, tap, etc.) a user needs to perform in the terminal (UE) to complete a task (e.g., play a video) is called the “*app user flow*.” In order to be able to automatically run a test case, the actual application user flow, with the user actions a user would need to perform on the phone to complete certain tasks defined in the test case, also has to be provided.

Each test case univocally defines the conditions of execution, the sequence of actions the user would perform (i.e., the *app user flow*), the sequence of actions that the elements of the testbed must perform, the traffic injected, the collection of measurements to take, and so forth. In order to obtain statistical significance, each test case includes a number of executions (iterations) under certain network conditions (herein called *scenarios*). Out of the various measurements made in the different iterations under any specific network conditions (*scenario*), a number of KPIs (Key Performance Indicators) are computed. The KPIs are normalized into a standard 1-to-5 scale, as typically used in MOS scores, and referred to as *synthetic-MOS*, a terminology that has been adopted from previous works [7, 21]. The *synthetic-MOS* values are aggregated across network scenarios to produce a number of intermediate *synthetic-MOS* scores, which finally are aggregated to obtain a *synthetic-MOS* score in each test case (see Figure 2).

The process to obtain the final TRIANGLE mark is sequential. First, for each domain, a weighted average of the *synthetic-MOS* scores obtained in each test case in the domain is calculated. Next, a weighted average of the *synthetic-MOS* values in all the domains of a use case is calculated to provide a single *synthetic-MOS* value per use

TABLE 2: TRIANGLE domains.

Category	Identifier	Domain
<i>Applications</i>	AUE	Apps User experience
	AEC	Apps Energy consumption
	RES	Device Resources Usage
	REL	Reliability
	NWR	Network Resources
<i>Devices</i>	DEC	Energy Consumption
	DDP	Data Performance
	DRF	Radio Performance
	DRA	User experience with reference apps
	IDR	Reliability
	IDP	Data Performance
	IEC	Energy consumption

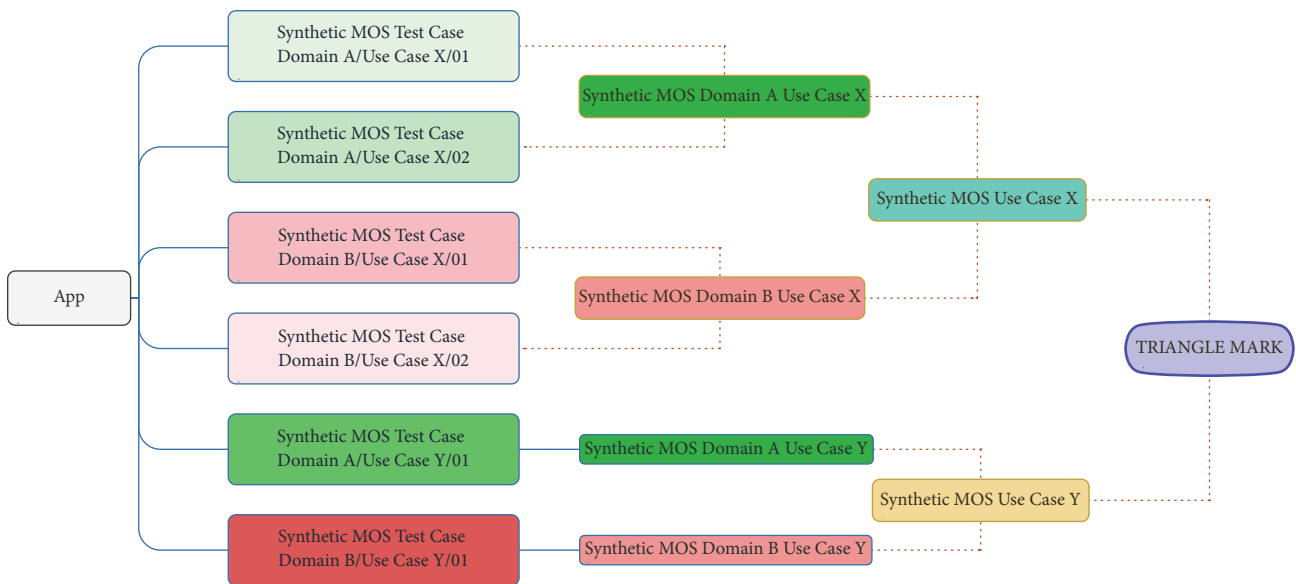


FIGURE 3: The process to obtain the TRIANGLE mark.

case. An application will usually be developed for one specific use case, as those defined in Table 1, but may be designed for more than one use case. In the latter case, a further weighted average is made with the synthetic-MOS scores obtained in each use case supported by the application. These sequential steps produce a single TRIANGLE mark, an overall quality score, as shown in Figure 3.

This approach provides a common framework for testing applications, for benchmarking applications, or even for certifying disparate applications. The overall process for an app that implements features of different use cases is depicted in Figure 3.

## 5. Details of the TRIANGLE QoE Computation

For each use case identified (see Table 1) and domain (see Table 2), a number of test cases have been developed within the TRIANGLE project. Each test case intends to test an

individual feature, aspect, or behaviour of the application under test, as shown in Figure 4.

Each test case defines a number of measurements, and because the results of the measurements depend on many factors, they are not, in general, deterministic, and, thus, each test case has been designed not to perform just one single measurement but to run a number of iterations ( $N$ ) of the same measurement. Out of those measurements, KPIs are computed. For example, if the time to load the first media frame is the measurement taken in one specific test case, the average user waiting time KPI can be calculated by computing the mean of the values across all iterations. In general, different use case-domain pairs have a different set of KPIs. The reader is encouraged to read [11] for further details about the terminology used in TRIANGLE.

Recommendation P.10/G.100 Amendment 1 Definition of Quality of Experience [2] notes that the overall acceptability may be influenced by user expectations and context. For the definition of the context, technical specifications ITU-T

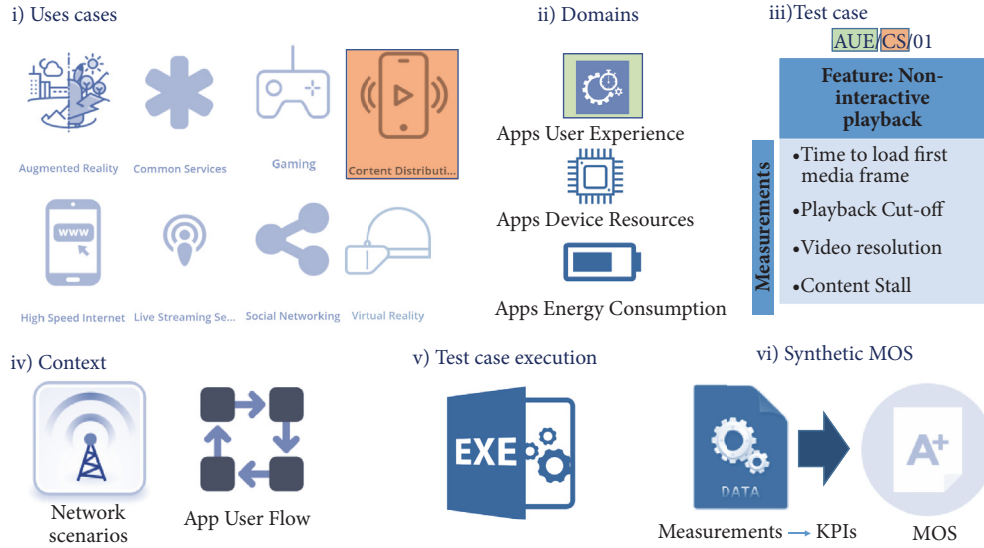


FIGURE 4: QoE computation steps.

G1030 “Estimating end-to-end performance in IP networks for data applications” [9] and ITU-T G1031 “QoE factors in web-browsing” [10] have been considered in TRIANGLE. In particular, ITU-T G1031 [10] identifies the following context influence factors: location (cafeteria, office, and home), interactivity (high-level interactivity versus low-level interactivity), task type (business, entertainment, etc.), and task urgency (urgent versus casual). User’s influence factors are, however, outside of the scope of the ITU recommendation.

In the TRIANGLE project, the context information has been captured in the networks *scenarios* defined (Urban - Internet Café Off Peak; Suburban - Shopping Mall Busy Hours; Urban - Pedestrian; Urban - Office; High speed train - Relay; etc.) and in the test cases specified in [11].

The test cases specify the conditions of the test but also a sequence of actions that have to be executed by the application (app user flows) to test its features. For example, the test case that tests the “Play and Pause” functionality defines the app user flow shown in Figure 5.

The transformation of KPIs into QoE scores is the most challenging step in the TRIANGLE framework. The execution of the test cases will generate a significant amount of raw measurements about several aspects of the system. Specific KPIs can then be extracted through statistical analysis: mean, deviation, cumulative distribution function (CDF), or ratio.

The KPIs will be individually interpolated in order to provide a common homogeneous comparison and aggregation space. The interpolation is based on the application of two functions, named Type I and Type II. By using the proposed two types of interpolations, the vast majority of KPIs can be translated into normalized MOS-type of metric (*synthetic-MOS*), easy to be averaged in order to provide a simple, unified evaluation.

*Type I.* This function performs a linear interpolation on the original data. The variables  $min_{KPI}$  and  $max_{KPI}$  are the worst and best known values of a KPI from a reference case. The

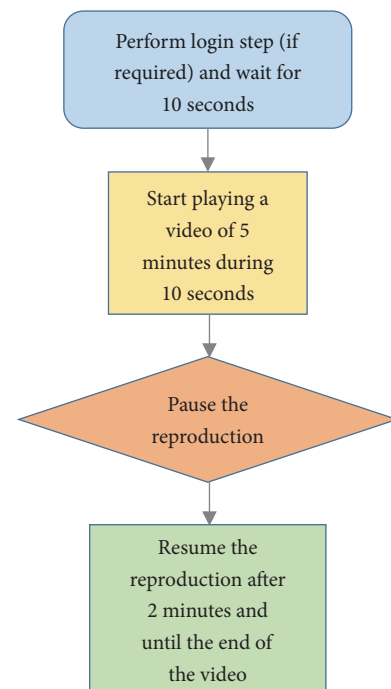


FIGURE 5: App user flow used in the “AUE/CS/02 Play and Pause” test case.

function maps a value,  $v$ , of a KPI, to  $v'$  (*synthetic-MOS*) in the range [1-to-5] by computing the following formula:

$$v' = \frac{v - min_{KPI}}{max_{KPI} - min_{KPI}} (5.0 - 1.0) + 1.0 \quad (1)$$

This function transforms a KPI to a synthetic-MOS value by applying a simple linear interpolation between the worst and best expected values from a reference case. If a future input case falls outside the data range of the KPI, the new value will

TABLE 3: AUE/CS/002 test case description.

<b>Identifier</b>	<i>AUE/CS/002 (App User Experience/Content Streaming/002)</i>
<b>Title</b>	Play and pause
<b>Objective</b>	Measure the ability of the AUT to pause and the resume a media file.
<b>Applicability</b>	(ICSG_ProductType = Application) AND (ICSG_UseCases includes CS) AND ICESA_CSPause
<b>Initial Conditions</b>	AUT in in [AUT_STARTED] mode. (Note: Defined in D2.2 [11] Appendix 4)
<b>Steps</b>	(1) The Test System commands the AUT to replay the Application User Flow (Application User Flow that presses first the Play button and later the Pause button). (2) The Test System measures whether pause operation was successful or not.
<b>Postamble</b>	(i) Execute the Postamble sequence (see section 2.6 in D2.2 [11] Appendix 4) (i) Playback Cut-off: Probability that successfully started stream reproduction is ended by a cause other than the intentional termination by the user.
<b>Measurements (Raw)</b>	(ii) Pause Operation: Whether pause operation is successful or not. (iii) Time to load first media frame (s) after resuming: The time elapsed since the user clicks resume button until the media reproduction starts. (Note: For Exoplayer the RESUME button is the PLAY button)

be set to the extreme value  $\min_{KPI}$  (if it is worse) or  $\max_{KPI}$  (if it is better).

*Type II.* This function performs a logarithmic interpolation and is inspired on the opinion model recommended by the ITU-T in [9] for a simple web search task. This function maps a value,  $v$ , of a KPI, to  $v'$  (*synthetic-MOS*) in the range [1-to-5] by computing the following formula:

$$v' = \frac{5.0 - 1.0}{\ln((a * \text{worst}_{KPI} + b) / \text{worst}_{KPI})} \cdot (\ln(v) - \ln(a * \text{worst}_{KPI} + a)) + 5 \quad (2)$$

The default values of  $a$  and  $b$  correspond to the simple web search task case ( $a = 0,003$  and  $b = 0,12$ ) [9, 22] and the worst value has been extracted from the ITU-T G1030. If during experimentation a future input case falls outside the data range of the KPI, the parameters  $a$  and  $b$  will be updated accordingly. Likewise, if through subjective experimentation other values are considered better adjustments for specific services, the function can be easily updated.

Once all KPIs are translated into synthetic-MOS values, they can be averaged with suitable weights. In the averaging process, the first step is to average over the network *scenarios* considered relevant for the use case, as shown in Figure 2. This provides the synthetic-MOS output value for the test case. If there is more than one test case per domain, which is generally the case, a weighted average is calculated in order to provide one synthetic-MOS value per domain, as depicted in Figure 3. The final step is to average the synthetic-MOS scores over all use cases supported by the application (see Figure 3). This provides the final score, that is, the TRIANGLE mark.

## 6. A Practical Case: Exoplayer under Test

For better understanding, the complete process of obtaining the TRIANGLE mark for a specific application, the Exoplayer,

TABLE 4: Measurement points associated with test case AUE/CS/002.

<i>Measurements</i>	<i>Measurement points</i>
<i>Time to load first media frame</i>	Media File Playback - Start
	Media File Playback - First Picture
<i>Playback cut-off</i>	Media File Playback - Start
	Media File Playback - End
<i>Pause</i>	Media File Playback - Pause

is described in this section. This application only has one use case: content distribution streaming services (CS).

Exoplayer is an application level media player for Android promoted by Google. It provides an alternative to Android's MediaPlayer API for playing audio and video both locally and over the Internet. Exoplayer supports features not currently supported by Android's MediaPlayer API, including DASH and SmoothStreaming adaptive playbacks.

The TRIANGLE project has concentrated in testing just two of the Exoplayer features: "Noninteractive Playback" and "Play and Pause." These features result in 6 test cases applicable, out of the test cases defined in TRIANGLE. These are test cases AUE/CS/001 and AUE/CS/002, in the App User Experience domain, test cases AEC/CS/001 and AEC/CS/002, in the App Energy Consumption domain, and test cases RES/CS/001 and RES/CS/002, in the Device Resources Usage domain.

The AUE/CS/002 "Play and Pause" test case description, belonging to the AUE domain, is shown in Table 3. The test case description specifies the test conditions, the generic app user flow, and the raw measurements, which shall be collected during the execution of the test.

The TRIANGLE project also offers a library that includes the measurement points that should be inserted in the source code of the app for enabling the collection of the measurements specified. Table 4 shows the measurement points required to compute the measurements specified in test case AUE/CS/002.

TABLE 5: Reference values for interpolation.

Feature	Domain	KPI	Synthetic MOS Calculation	KPI_min	KPI_max
Non-Interactive Playback	AEC	Average power consumption	Type I	10 W	0.8 W
Non-Interactive Playback	AUE	Time to load first media frame	Type II	KPI_worst=20 ms	
Non-Interactive Playback	AUE	Playback cut-off ratio	Type I	50%	0
Non-Interactive Playback	AUE	Video resolution	Type I	240p	720p
Non-Interactive Playback	RES	Average CPU usage	Type I	100%	16%
Non-Interactive Playback	RES	Average memory usage	Type I	100%	40%
Play and Pause	AEC	Average power consumption	Type I	10 W	0.8 W
Play and Pause	AUE	Pause operation success rate	Type I	50%	100%
Play and Pause	RES	Average CPU usage	Type I	100%	16%
Play and Pause	RES	Average memory usage	Type I	100%	40%

The time to load first media picture measurement is obtained subtracting the timestamp of the measurement point “Media File Playback – Start” from the measurement point “Media File Playback – First Picture.”

As specified in [11], all scenarios defined are applicable to the content streaming use case. Therefore, test cases in the three domains currently supported by the testbed are executed in all the *scenarios*.

Once the test campaign has finished, the raw measurement results are processed to obtain the KPIs associated with each test case: average current consumption, average time to load first media frame, average CPU usage, and so forth. The processes applied are detailed in Table 5. Based on previous experiments performed by the authors, the behaviour of the time to load the first media frame KPI resembles the web response time KPI (i.e., the amount of time the user has to wait for the service) and thus, as recommended in the opinion model for web search introduced in [9], a logarithmic interpolation (type II) has been used for this metric.

The results of the initial process, that is, the KPIs computation, are translated into *synthetics-MOS* values. To compute these values, reference benchmarking values for each of the KPIs need to be used according to the normalization and interpolation process described in Section 5. Table 5 shows what has been currently used by TRIANGLE for the App User Experience domain, which is also used by NGMN as reference in their precommercial Trials document [23].

For example, for the “time to load first media frame” KPI shown in Table 5, the type of aggregation applied is averaging and the interpolation formula used is Type II.

To achieve stable results, each test case is executed 10 times (10 iterations) in each network scenario. The synthetic-MOS value in each domain is calculated by averaging the measured synthetic-MOS values in the domain. For example, synthetic-MOS value in the RES domain is obtained by averaging the synthetic-MOS value of “average CPU usage” and “average memory usage” from the two test cases.

Although Exoplayer supports several video streaming protocols, in this work only DASH [24] (Dynamic Adaptive Streaming over HTTP) has been tested. DASH clients should seamlessly adapt to changing network conditions by making decisions on which video segment to download (videos are encoded at multiple bitrates). The Exoplayer’s default

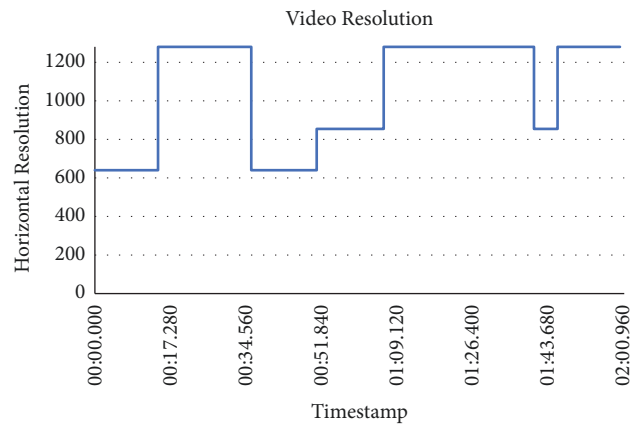


FIGURE 6: Video Resolution evolution in the Driving Urban Normal scenario.

adaptation algorithm is basically throughput-based and some parameters control how often and when switching can occur.

During the testing, the testbed was configured with the different network scenarios defined in [11]. In these scenarios, the network configuration changes dynamically following a random pattern, resulting in different maximum throughput rates. The expected behaviour of the application under test is that the video streaming client adapts to the available throughput by decreasing or increasing the resolution of the received video. Figure 6 depicts how the client effectively adapts to the channel conditions.

However, the objective of the testing carried out in the TRIANGLE testbed is not just to verify that the video streaming client actually adapts to the available maximum throughput but also to check whether this adaptation improves the users’ experience quality.

Table 6 shows a summary of the synthetic-MOS values obtained per scenario in one test case of each domain. The scores obtained in the RES and AEC domains are always high. In the AUE domain, the synthetic MOS associated with the Video Resolution shows low scores in some of the scenarios because the resolution decreases, reasonable good scores in the time to load first media, and high scores in the time to playback cut-off ratio. Overall, it can be concluded that the

TABLE 6: Synthetic MOS values per test case and scenario for the feature “Noninteractive Playback”.

Scenario	AUE domain			AEC domain	RES domain	
	Test Case AUE/CS/001			Test Case AEC/CS/001	Test Case RES/CS/001	
	Time to load first media frame	Playback Cut-off ratio	Video Resolution mode	Average Power Consumption	Average CPU Usage	Average RAM Usage
<i>HighSpeed Direct Passenger</i>	2.1	3.1	2.3	4.7	4.3	4.2
<i>Suburban Festival</i>	3.8	4.7	3.1	4.8	4.3	4.1
<i>Suburban shopping mall busy hours</i>	3.7	3.7	1.3	4.8	4.4	4.1
<i>Suburban shopping mall off-peak</i>	3.6	3.1	2.3	4.8	4.3	4.1
<i>Suburban stadium</i>	3.8	2.9	2.1	4.7	4.4	4.1
<i>Urban Driving Normal</i>	2.6	3.9	2.8	4.7	4.4	4
<i>Urban Driving Traffic Jam</i>	3.4	3.7	1.6	4.8	4.4	4
<i>Urban Internet Café Busy Hours</i>	3.8	3.7	1.9	4.8	4.4	4
<i>Urban Internet Café Off Peak</i>	3.8	3.1	2.3	4.8	4.3	4
<i>Urban Office</i>	3.8	4.7	3.3	4.8	4.5	4.3
<i>Urban Pedestrian</i>	3.9	2.6	2	4.7	4.4	4
	3.5	3.6	2.3	4.7	4.4	4.1

DASH implementation of the video streaming client under test is able to adapt to the changing conditions of the network, maintaining an acceptable rate of video cut-off, rebuffering times, and resources usage.

The final score in each domain is obtained by averaging the synthetic-MOS values from all the tested network scenarios. Figure 7 shows the spider diagram for the three domains tested. In the User Experience domain, the score obtained is lower than the other domains, due to the low synthetic-MOS values obtained for the video resolution.

The final synthetic MOS for the use case Content Distribution Streaming is obtained as a weighted average of the three domains, representing the overall QoE as perceived by the user. The final score for the Exoplayer version 1.516 and the features tested (Noninteractive Playback and Play and Pause) is 4.2, which means that the low score obtained in the video resolution is compensated with the high scores in other KPIs.

If an application under test has more than one use case, the next steps in the TRIANGLE mark project approach would be the aggregation per use case and the aggregation over all use cases. The final score, the TRIANGLE mark, is an estimation of the overall QoE as perceived by the user.

In the current TRIANGLE implementation, the weights in all aggregations are the same. Further research is needed to appropriately define the weights of each domain and each use case in the overall score of the applications.

## 7. Conclusions

The main contribution of the TRIANGLE project is the provision of a framework that generalizes QoE computation

and enables the execution of extensive and repeatable test campaigns to obtain meaningful QoE scores. The TRIANGLE project has also defined a methodology, which is based on the transformation and aggregation of KPIs, its transformation into synthetic-MOS values, and its aggregation over the different *domains* and *use cases*.

The TRIANGLE approach is a methodology flexible enough to generalize the computation of QoE for any application/service. The methodology has been validated testing the DASH implementation in the Exoplayer App. To confirm the suitability of the weights used in the averaging process and the interpolation parameters, as well as to verify the correlation of the obtained MOS with that scored by users, the authors have started experiments with real users and initial results are encouraging.

The process described produces a final TRIANGLE mark, a single quality score, which could eventually be used to certify applications after achieving a consensus on the different values of the process (weights, limits, etc.) to use.

## Data Availability

The methodology and results used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

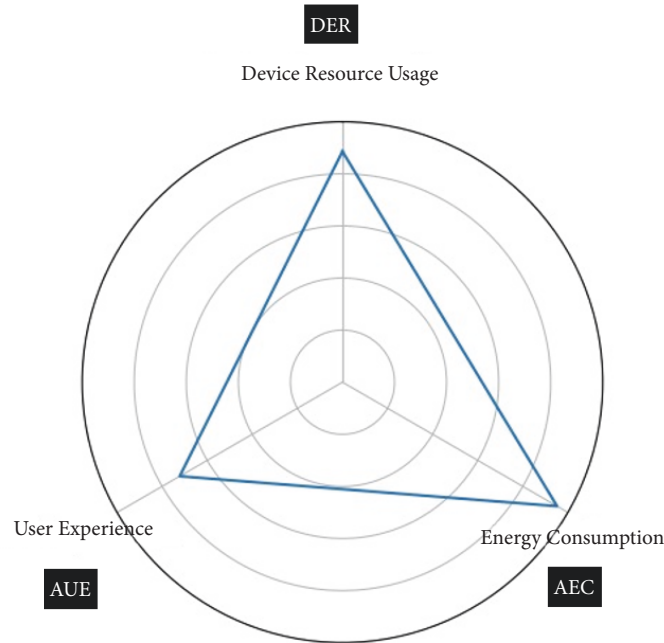


FIGURE 7: Exoplayer synthetic-MOS values per domain.

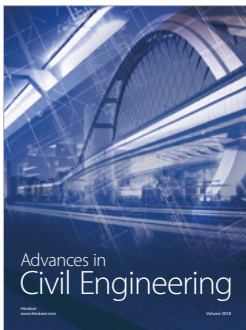
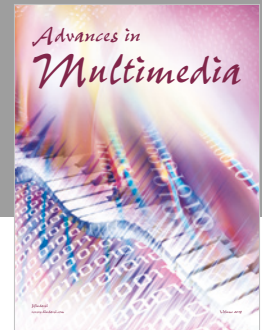
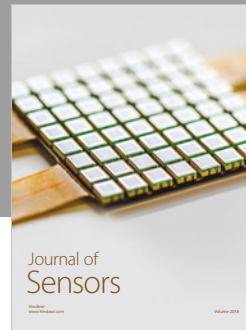
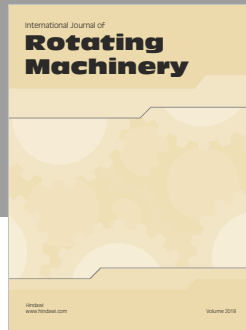
## Acknowledgments

The TRIANGLE project is funded by the European Union's Horizon 2020 Research and Innovation Programme (Grant Agreement no. 688712).

## References

- [1] ETSI, "Human factors: quality of experience (QoE) requirements for real-time communication services," Tech. Rep. 102 643, 2010.
- [2] ITU-T, "P.10/G.100 (2006) amendment 1 (01/07): new appendix I - definition of quality of experience (QoE)," 2007.
- [3] F. Kozamernik, V. Steinmann, P. Sunna, and E. Wyckens, "SAMVIQ - A new EBU methodology for video quality evaluations in multimedia," *SMPTE Motion Imaging Journal*, vol. 114, no. 4, pp. 152–160, 2005.
- [4] ITU-T, "G.107 : the E-model: a computational model for use in transmission planning," 2015.
- [5] J. De Vriendt, D. De Vleeschauwer, and D. C. Robinson, "QoE model for video delivered over an LTE network using HTTP adaptive streaming," *Bell Labs Technical Journal*, vol. 18, no. 4, pp. 45–62, 2014.
- [6] S. Jelassi, G. Rubino, H. Melvin, H. Youssef, and G. Pujolle, "Quality of Experience of VoIP Service: A Survey of Assessment Approaches and Open Issues," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 491–513, 2012.
- [7] M. Li, C.-L. Yeh, and S.-Y. Lu, "Real-Time QoE Monitoring System for Video Streaming Services with Adaptive Media Playback," *International Journal of Digital Multimedia Broadcasting*, vol. 2018, Article ID 2619438, 11 pages, 2018.
- [8] S. Baraković and L. Skorin-Kapov, "Survey and Challenges of QoE Management Issues in Wireless Networks," *Journal of Computer Networks and Communications*, vol. 2013, Article ID 165146, 28 pages, 2013.
- [9] ITU-T, "G.1030: estimating end-to-end performance in IP networks for data applications," 2014.
- [10] ITU-T, "G.1031 QoE factors in web-browsing," 2014.
- [11] EU H2020 TRIANGLE Project, *Deliverable D2.2 Final report on the formalization of the certification process, requirements and use cases*, 2017, <https://www.triangle-project.eu/project-old/deliverables/>.
- [12] Q. A. Chen, H. Luo, S. Rosen et al., "QoE doctor: diagnosing mobile app QoE with automated UI control and cross-layer analysis," in *Proceedings of the Conference on Internet Measurement Conference (IMC '14)*, pp. 151–164, ACM, Vancouver, Canada, November 2014.
- [13] M. A. Mehmood, A. Wundsam, S. Uhlig, D. Levin, N. Sarrar, and A. Feldmann, "QoE-Lab: Towards Evaluating Quality of Experience for Future Internet Conditions," in *Testbeds and Research Infrastructure*, Korakis T., Li H., Tran-Gia P., and H. S. Park, Eds., vol. 90 of *TridentCom 2011, Lnicst*, pp. 286–301, Springer, Development of Networks and Communities, Berlin, Germany, 2012.
- [14] D. Levin, A. Wundsam, A. Mehmood, and A. Feldmann, "Berlin: The Berlin Experimental Router Laboratory for Innovative Networking," in *TridentCom 2010. Lnicst*, T. Magedanz, A. Gavras, N. H. Thanh, and J. S. Chase, Eds., vol. 46 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 602–604, Springer, Heidelberg, Germany, 2011.

- [15] K. De Moor, I. Ketyko, W. Joseph et al., “Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting,” *Mobile Networks and Applications*, vol. 15, no. 3, pp. 378–391, 2010.
- [16] R. Sanchez-Iborra, M.-D. Cano, J. J. P. C. Rodrigues, and J. Garcia-Haro, “An Experimental QoE Performance Study for the Efficient Transmission of High Demanding Traffic over an Ad Hoc Network Using BATMAN,” *Mobile Information Systems*, vol. 2015, Article ID 217106, 14 pages, 2015.
- [17] P. Oliver-Balsalobre, M. Toril, S. Luna-Ramírez, and R. García Garaluz, “A system testbed for modeling encrypted video-streaming service performance indicators based on TCP/IP metrics,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, 2017.
- [18] M. Solera, M. Toril, I. Palomo, G. Gomez, and J. Poncela, “A Testbed for Evaluating Video Streaming Services in LTE,” *Wireless Personal Communications*, vol. 98, no. 3, pp. 2753–2773, 2018.
- [19] A. Álvarez, A. Díaz, P. Merino, and F. J. Rivas, “Field measurements of mobile services with Android smartphones,” in *Proceedings of the IEEE Consumer Communications and Networking Conference (CCNC '12)*, pp. 105–109, Las Vegas, Nev, USA, January 2012.
- [20] NGMN Alliance, “NGMN 5G white paper,” 2015, [https://www.ngmn.org/fileadmin/ngmn/content/downloads/Technical/2015/NGMN\\_5G\\_White\\_Paper\\_V1\\_0.pdf](https://www.ngmn.org/fileadmin/ngmn/content/downloads/Technical/2015/NGMN_5G_White_Paper_V1_0.pdf).
- [21] “Infrastructure and Design for Adaptivity and Flexibility,” in *Mobile Information Systems*, Springer, 2006.
- [22] J. Nielsen, “Response Times: The Three Important Limits,” in *Usability Engineering*, 1993.
- [23] NGMN Alliance, “Definition of the testing framework for the NGMN 5G pre-commercial networks trials,” 2018, [https://www.ngmn.org/fileadmin/ngmn/content/downloads/Technical/2018/180220\\_NGMN\\_PreCommTrials\\_Framework\\_definition\\_v1\\_0.pdf](https://www.ngmn.org/fileadmin/ngmn/content/downloads/Technical/2018/180220_NGMN_PreCommTrials_Framework_definition_v1_0.pdf).
- [24] 3GPP TS 26.246, “Transparent end-to-end Packet-switched Streaming Services (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH),” 2018.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

