H2020-ICT-688712

Project: H2020-ICT-688712

Project Name:

5G Applications and Devices Benchmarking (TRIANGLE)

Deliverable D2.7

QoE Evaluation: The TRIANGLE Testbed Approach

Date of delivery:	04/06/2019	Version:	1.1
Start date of Project:	01/01/2016	Duration:	36 months

Deliverable D2.7

QoE Evaluation: The TRIANGLE Testbed Approach

Project Number:	ICT-688712
Project Name:	5G Applications and Devices Benchmarking
Project Acronym	TRIANGLE

Document Number:	ICT-688712-TRIANGLE/D2.7
Document Title:	QoE Evaluation: The TRIANGLE Testbed Approach
Lead beneficiary:	DEKRA Testing and Certification, S.A.U.
Editor(s):	DEKRA Testing and Certification, S.A.U.
Authors:	Keysight Technologies Belgium (Michael Dieudonne), Universidad of Malaga (Pedro Merino, Almudena Diaz), DEKRA (Janie Baños, Carlos Cárdenas)
Dissemination Level:	PU
Contractual Date of Delivery:	31/10/2018
Work Package Leader:	DEKRA Testing and Certification, S.A.U.
Status:	Final
Version:	1.1
File Name:	TRIANGLE_Deliverable_D2.7_v1.1 FINAL

Abstract

This deliverable 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.

Keywords

QoE, QoS, mark, NGMN, scope

Document history

V1.0	Initial release of the document
V1.1	Improvement to section 3 around the explanation of figure 2, 3 & 4 to better understand the correlation between the 3 figures.

Executive summary

The success of 5G (the fifth generation of mobile communications), and to some extent that of 4G, depends on its capability 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]) adds confusion to the subject: QoE (Quality of Experience), QoS (Quality of Service), QoSD (Quality of Service Delivered/achieved by service provider), QoSE (Quality of Service Experience/Perceived by customer/user), etc. 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 on the typical MOS (Mean Opinion Score). Even here, 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. Technical parameters, which can be monitored, and where its 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 [6] 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 [7] or video streaming [8], [9]. This deliverable 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 non-controlled, 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 [10] and G1031 [11], 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 deliverable 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.

Contents

List of Figures

List of Tables

List of Abbreviations

2G	Second generation wireless technology
3G	Third generation wireless technology
3GPP	3rd Generation Partnership Project
4G	Forth generation wireless technology
5G	Fifth generation wireless technology
CO	Confidential
D	Deliverables
eNB	Evolved Node B
ETSI	European Telecommunications Standards Institute
E-UTRAN	Evolved UTRAN
EVM	Error Vector Magnitude
FDD	Frequency Division Duplex
FEC	Forward Error Correction
GCF	Global Certification Forum
GPRS	General Packet Radio Service
GSM	Global System for Mobile communications
HTC	Human Type Communications
ICT	Information and Communications Technology
IEEE	Institute of Electrical and Electronics Engineers
IMT	International Mobile Communications
IoT	Internet of Things
IP	Intellectual Property
IPR	Intellectual Property Rights
IR	Internal report
ITS	Intelligent Transport System
ITU	International Telecommunication Union
ITU-R	International Telecommunication Union-Radio
KPI	Key Performance Indicator

LAN	Local Area Network
LBT	Listen Before Talk
LPWAN	Low Power Wide Area Networks
LTE	Long Term Evolution
LTE-A	Long Term Evolution-Advanced
LTE-M	Long Term Evolution For Internet of Things
M	Milestones
Mbps	megabits per second
Mo	Month
MEC	Mobile Edge Computing
MGT	Management
MIMO	Multiple-Input Multiple-Output
MMC	Massive Machine Communication
M2M	Machine to Machine
MTC	Machine Type Communications
NB-IoT	Narrow Band Internet of Things
NFV	Network Function Virtualization
NFVO	NFV Orchestrator
NR	New Radio (temporary denomination for new 5G radio)
OCF	Open Connectivity Foundation
OEM	Original Equipment Manufacturer
OIC	Open Interconnect Consortium
QoE	quality of experience
QoS	Quality of Service
RAN	Radio Access Network
RAT	Radio Access Technology
UE	User Equipment
UL	Uplink
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network
V2V	Vehicle-to-Vehicle
WLAN	Wireless Local Area Network
WP	Work Package

1 State of the Art

Modelling and evaluating QoE in current and next generation of mobile networks is an important and active research area [9]. 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 [12] tool is closely related to the TRIANGLE testbed, since its main purpose is the evaluation of mobile apps QoE in an accurate, systematic a 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 th causes of a degraded QoE, but it is no 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 multi-purpose 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. Similarly 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, 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, etc.

De Moor et al., [15] proposed a user-centric methodology for the multi-dimensional 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.), 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 analyses 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. Secondly, 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 its 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 set-up 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 pre-configured 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

2 Overview of TRIANGLE Test Bed

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.

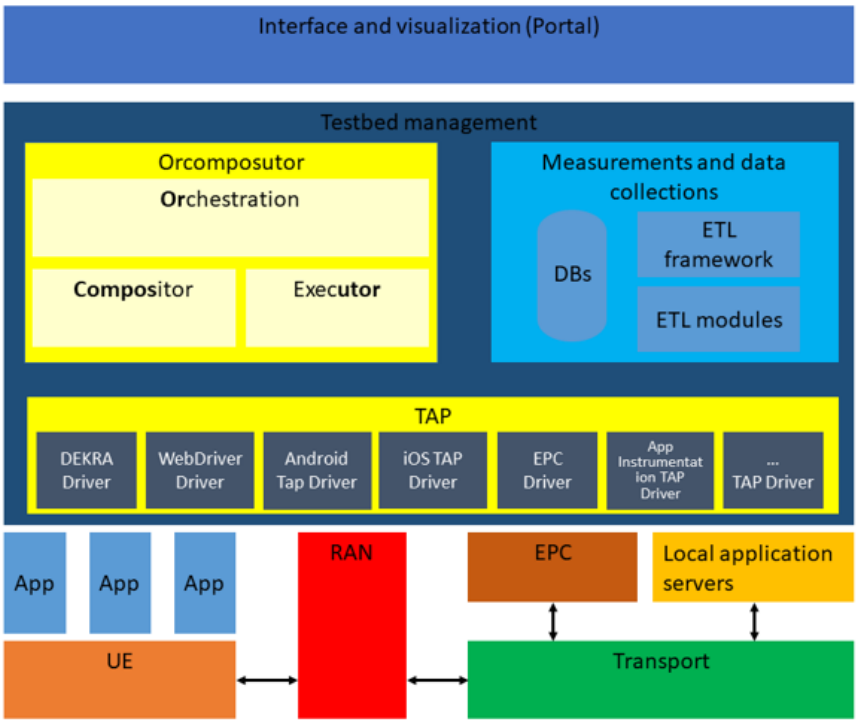

Figure 1. TRIANGLE testbed architecture

To facilitate the use of the TRIANGLE testbed for different objectives (testing, benchmarking, 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, their 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 device (UE, LTE User Equipment) on which the application under test runs, include DEKRA Agents and the TestelDroid [19] tool from UMA. TRIANGLE also provides an instrumentation library so that appdevelopers can deliver measurement outputs, which cannot otherwise be extracted and must to 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 a 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.

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, i.e., 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 EPDG (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 of fully web based. This tool is also used to prerecord the apps 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 conducted 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.

3 TRIANGLE Approach

The TRIANGLE testbed is an end-to-end framework devoted to testing and benchmarking of 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 the TRIANGLE 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 [5] . 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 type of the QoE evaluation aspects needs to be addressed. The final use cases categorization was defined in [20] and encompasses both the services normally accessible via mobile phones (UEs) and the ones that can be integrated in, e.g., 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.

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

Table 2. TRIANGLE domains

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

To produce data to evaluate the QoE, test cases that can be run on the TRIANGLE testbed have been designed, developed and implemented. 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. i.e., the test campaign with the list of applicable test cases, is automatically generated.

The sequence of user actions (type, swipe, touch, 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 of 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, etc., and the collection of measurements to take. Each test case is executed under certain network scenarios relevant to the use case. The network scenarios define the properties of the radio channel (e.g., channel model, Doppler frequency, received signal power) and the network conditions (e.g., amount of available frequency/time domain resources for scheduling), which have great impact on the measurement results. Because of the statistical nature of the dynamics introduced in the network scenario, the measurement results may vary from one iteration to another. Thus, it is necessary to run multiple executions (iterations) for each network scenario to make a good estimation of the measurement results. In each iteration, the measurements specified in the test case are collected. The KPIs specified also in the test case are computed aggregating the measurements from all the iterations, se note (1) in Figure 2. As individual KPIs are measured in different dimensions and scales, they are normalized into a standard 1-to-5 scale, referred to as synthetic mean opinion score (MOS), a terminology that has been adopted from previous works [8], [21], for further QoE computations. The synthetic-MOS values are weight averaged/aggregated to produce a synthetic-MOS score in each scenario, see note (2).. A final

aggregation of the synthetic MOS obtained in all the scenarios is done to obtain the final one, see note (3) in Figure 2. The final synthetic MOS is the result of the test case execution,

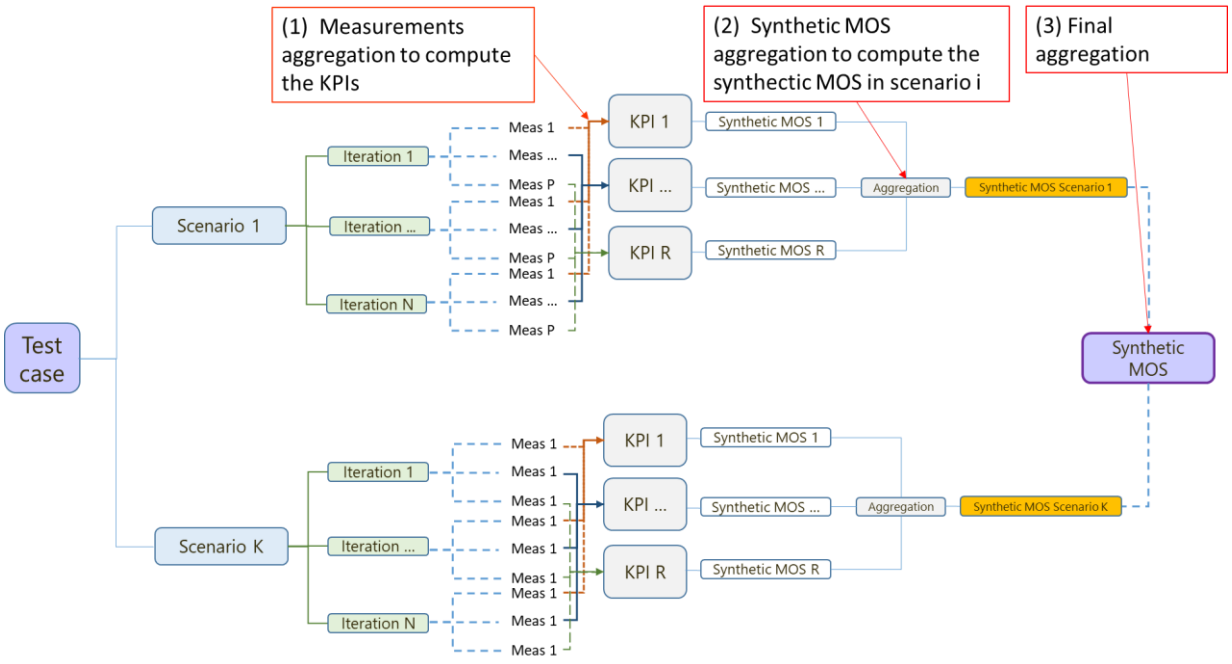

Figure 2. The process to obtain the “synthetic MOS score” in a TRIANGLE test case

Figure 2 shows the procedure of obtaining a synthetic MOS score for just one test case. For obtaining the final TRIANGLE mark for an app, usually, several test cases are executed, because the test cases are oriented to the testing of the different features provided by the apps. The features are classified into the uses cases shown in Table 1, for example the feature “Play and Pause” belongs to the “Content Distribution Streaming Services use case”. Additionally, for the same feature, there are different test cases focused on the different domains identified in Table 2 (Apps Energy Consumption, Device Resources Usage, etc.). Figure 3 depicts the aggregation procedure applied to compute the final TRIANGLE mark when executing several test cases, In particular, Figure 3, depicts the execution of two test cases for the domain A and the use case X, two test cases for the domain B and the use case X, one test case for the domain A and the use case Y and one test case for the domain B and the use case Y.

Accordingly, in the case of executing several test cases, to obtain the final TRIANGLE mark, the synthetic-MOS scores obtained in each test case as illustrated in Figure 2, are weighted averaged per domain, see note (2) in Figure 3. The synthetic MOS values in each domain of an use case are further weighted averaged to provide a single synthetic MOS value per use case, see note (3) in Figure 3. The final TRIANGLE mark is the aggregation of the synthetic- MOS obtained for the use case X and the use case Y.

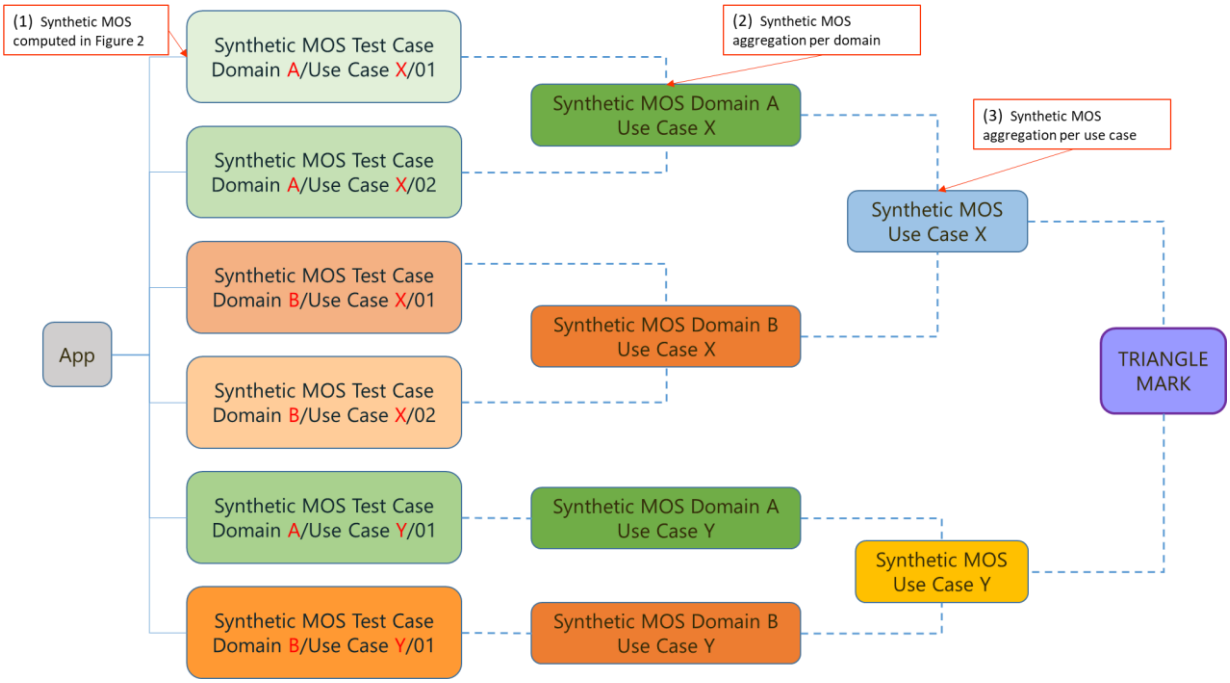

Figure 3. The process to obtain the TRIANGLE mark

As a summary, for an application under test, there could be multiple use cases associated with it (e.g., live streaming service, virtual reality, social networking, etc.), depending on the design objective of the application. Each use case can be evaluated in one or more domains. Domain is a categorization of the KPIs related to a specific subject, such as user experience, resource usage, energy consumption, reliability, etc. Within each domain, one or several test cases are defined. And within each test case, one or more KPIs are defined for the measurement. The TRIANGLE testbed provides a common framework for testing/benchmarking applications. It allows the application tester to configure the appropriate use cases, domains, test cases, KPIs, and network scenarios for which the application should be under test. The testbed will execute the configured test campaign, and perform the post-processing of the measurement results, i.e., KPI normalization and synthetic MOS value aggregation as illustrated in Figure 2 and Figure 3, to obtain the final TRIANGLE mark. This approach provides a common framework for testing applications, for benchmarking applications, or eve, for certifying disparate applications.

Figure 4 illustrates the overall process for computing the synthetic MOS for a test case which belongs to the Content Distribution Streaming Services use case and Apps User Experience Domain. The test case will be executed in all the scenarios using the app user flow needed to stimulate the feature under test. After the execution of the test case the procedure explained in Figure 2 is applied to compute the synthetic MOS.

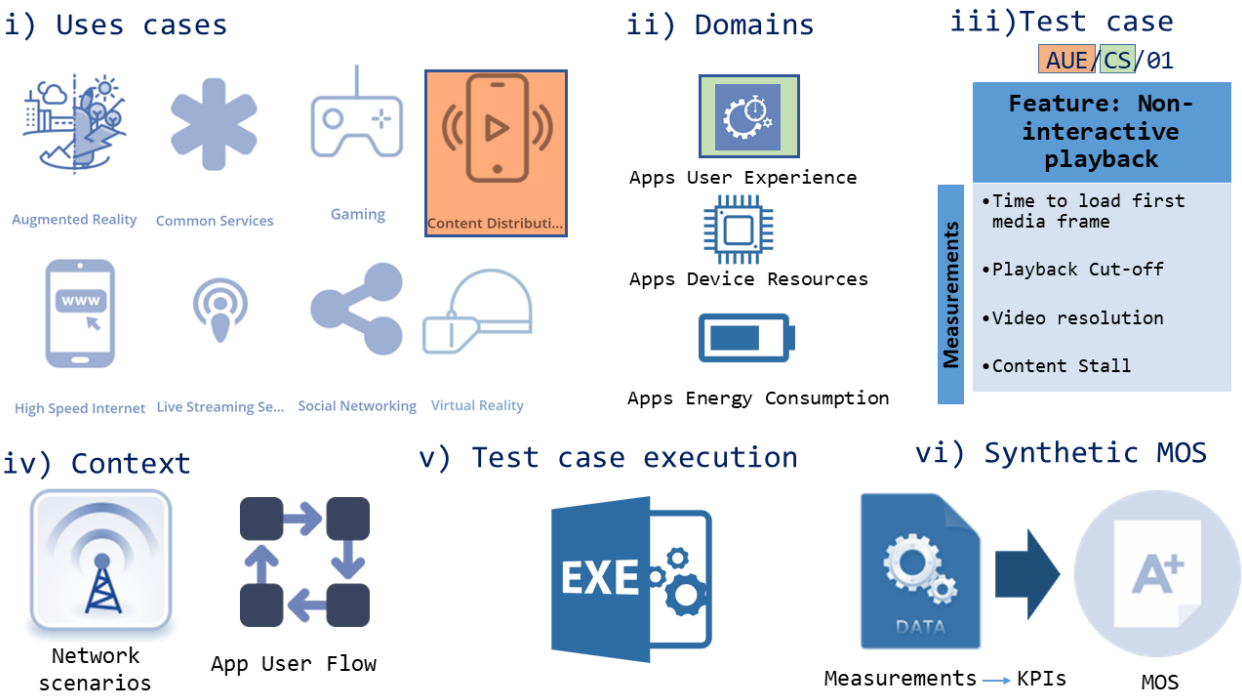

Figure 4. QoE computation steps

4 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.

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 [20] 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 G1030 “Estimating end-to-end performance in IP networks for data applications” [10] and ITU-T G1031 “QoE factors in web-browsing” [11] have been considered in TRIANGLE. In particular, [11] identifies the following context influence factors: Location (Cafeteria, office, home), Interactivity (High level interactivity vs low level interactivity), Task type (business, entertainment, etc.), and Task urgency (urgent vs 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 Cafe Off Peak; Suburban - Shopping Mall Busy Hours; Urban – Pedestrian; Urban – Office; High speed train – Relay; etc.) and in the test cases specified in [20].

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.

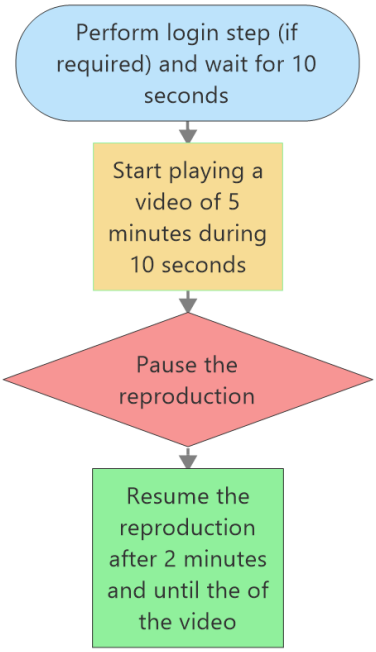

Figure 5. App user flow used in the “AUE/CS/02 Play and Pause” test case

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

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

This function is to be used for KPIs that will be transformed by 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, it will 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 [10] 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:

$$v' = \frac{5.0 - 1.0}{\ln((a * worst_{KPI} + b)/worst_{KPI})} \cdot (\ln(v) - \ln(a * worst_{KPI} + a)) + 5$$

The default "a" and "b" values in TRIANGLE correspond to the simple web search task case (a = 0,003 and b = 0,12) [10], [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 weighted averaged. In the averaging process, the first step is to average over the network *scenarios* considered relevant for the use case and domain. This provides the synthetic MOS output value for the test case. Results of test cases in each domain are weighted averaged to provide one single synthetic MOS value per domain.

To provide a single use case synthetic MOS score, the MOS score of the different test cases corresponding to different domains are weighted averaged. The final process is to average the synthetic MOS scores over all use cases supported by the application. This provides the final score, i.e., the TRIANGLE mark.

5 A practical case: Exoplayer under test

For better understanding, the complete process of obtaining the TRIANGLE mark for a specific application, the Exoplayer, 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 project has concentrated in testing, just two of the Exoplayer features: "Non-Interactive Playback" and "Play and Pause". This results 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.

Table 3. AUE/CS/002 test case description

Identifier	**AUE/CS/002 (*App User Experience/Content Streaming/002*)**
Title	Play and pause
Objective	Measure the ability of the AUT to pause and the resume a media file.
Applicability	(ICSG_ProductType = Application) AND (ICSG_UseCases includes CS) AND ICSA_CSPause
Initial Conditions	AUT in in [AUT_STARTED] mode. (Note: Defined in D2.2 Appendix 4)
Steps	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.
Postamble	• Execute the Postamble sequence (see section 2.6 in D2.2Appendix 4)
Measurements (Raw)	• Playback Cut-off: Probability that successfully started stream reproduction is ended by a cause other than the intentional termination by the user. • Pause Operation: Whether pause operation is successful or not. • 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)*

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 4. Measurement points associated to test case AUE/CS/002

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

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 [20], 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 to each test case: average current consumption, average time to load first media frame, average CPU usage, etc. The processes applied are detailed in Table 5.

The results of the initial process, i.e., 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 V. 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 pre-commercial Trials document [23].

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

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%

Table 6. Synthetic MOS values per test case / scenario for “Non-Interactive Playback”

	AUE domain			AEC domain	RES domain	
Scenario	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	AverageRAM Usage
HighSpeed Direct Passenger	2.1	3.1	2.3	4.7	4.3	4.2
Suburban Festival	3.8	4.7	3.1	4.8	4.3	4.1
Suburban shopping mall busy hours	3.7	3.7	1.3	4.8	4.4	4.1
Suburban shopping mall off-peak	3.6	3.1	2.3	4.8	4.3	4.1
Suburban stadium	3.8	2.9	2.1	4.7	4.4	4.1
Urban Driving Normal	2.6	3.9	2.8	4.7	4.4	4
Urban Driving Traffic Jam	3.4	3.7	1.6	4.8	4.4	4
Urban Internet Café Busy Hours	3.8	3.7	1.9	4.8	4.4	4
Urban Internet Café Off Peak	3.8	3.1	2.3	4.8	4.3	4
Urban Office	3.8	4.7	3.3	4.8	4.5	4.3
Urban Pedestrian	3.9	2.6	2	4.7	4.4	4
	3.5	3.6	2.3	4.7	4.4	4.1

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 is the RES domain 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 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 [20]. 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 TRIANGE testbed is not just to verify that the video streaming client actually adapts to the available maximum throughput, but also to check this adaptation improves the users' experience quality.

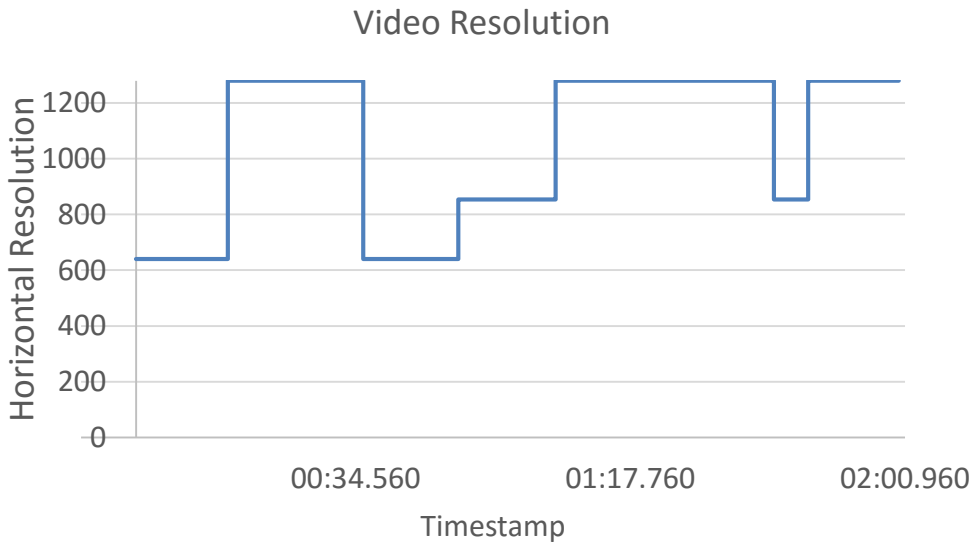

Figure 6. Video Resolution evolution in the Driving Urban Normal scenario

Table 5 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 to the Video Resolution shows low scores in some on 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 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.

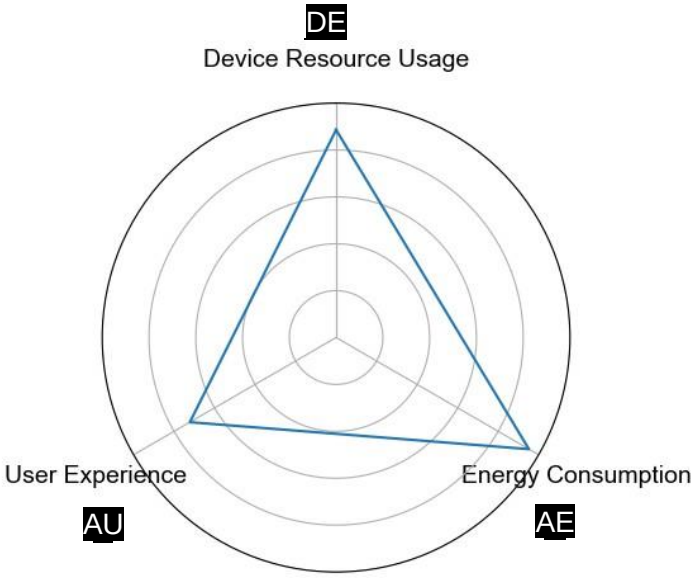

Figure 7. Exoplayer Synthetic MOS values per domains

A 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 tested domains. In the User Experience domain the score obtained is lower, 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 weight average of the three domains, representing the overall quality of experience as perceived by the user. The final score for the Exoplayer version 1.516 and the features tested (Non Interactive-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 tests.

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.

6 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 process produces a final TRIANGLE mark, a single quality score that could eventually be used to certify applications. The approach developed in TRIANGLE 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 media player Exoplayer.

7 References

[1] ETSI, “TR 102 643 Human Factors (HF); Quality of Experience (QoE) requirements for real-time communication services,” 2010.

[2] ITU-T, “Recommendation P.10/G.100 (2006) Amendment 1,” 2007.

[3] V. S. P. S. a. E. W. F. Kozamernik, “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] NGMN, February 2015. [Online]. Available: http://www.ngmn.org/fileadmin/ngmn/content/downloads/Technical/2015/NGMN_5G_White_Paper_V1_0.pdf.

[6] D. D. V. a. D. C. R. J. De Vriendt, “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.

[7] G. R. H. M. H. Y. a. G. P. S. Jelassi, “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.

[8] C.-L. Y. a. S.-Y. L. Mingfu Li, “Real-Time QoE Monitoring System for Video Streaming Services with Adaptive Media Playout,” *International Journal of Digital Multimedia Broadcasting,* p. 11, 2018.

[9] S. a. S.-K. L. Baraković, “Survey and Challenges of QoE Management Issues in Wireless Networks,” *Journal of Computer Networks and Communications,* p. Article ID 165146, 2013.

[10] ITU-T, “G.1030 Estimating end-to-end performance in IP networks for data applications,” 2014.

[11] ITU-T, “G.1031 QoE factors in web-browsing,” 2014.

[12] Q. A. L. H. R. S. e. a. Chen, “QoE Doctor: Diagnosing Mobile App QoE with Automated UI Control and Cross-layer Analysis,” *In Proceedings of the 2014 Conference on Internet Measurement Conference (IMC '14).,* pp. 151-164, 2014.

[13] W. A. U. S. L. D. S. N. F. A. Mehmood M.A., “QoE-Lab: Towards Evaluating Quality of Experience for Future Internet Conditions,” *Korakis T., Li H., Tran-Gia P., Park HS. (eds) Testbeds and Research Infrastructure. Development of Networks and Communities. TridentCom ,* 2011.

[14] D. W. A. M. A. F. A. B. Levin, “The Berlin Experimental Router Laboratory for Innovative Networking,” *TridentCom,* vol. 46, pp. 602-604, 2010.

[15] K. K. I. J. W. e. a. De Moor, “Proposed framework for evaluating quality of experience in a mobile, testbed-oriented living lab setting,” *Mobile Netw Appl ,* 2010.

[16] R. C. M. D. R. J. J. a. G.-H. J. Sanchez-Iborra, “An Experimental QoE Performance Study for the Efficient Transmission of High Demanding Traffic over an Ad Hoc Network Using BATMAN,” *Mobile Information Systems,* 2015.

[17] P. T. M. L.-R. S. e. a. Oliver-Balsalobre, “A system testbed for modelling encrypted video-streaming service performance indicators based on TCP/IP metrics,” *Wireless Com Network ,* 2017.

[18] M. T. M. P. I. e. a. Solera, “A testbed for evaluating video streaming services in LTE,” *Wireless Personal Communications,* 2018.

[19] D. A. M. P. R. F. Álvarez A, “ield measurements of mobile services with Android smartphones,” IEEE Consumer Communications and Networking Conference, 2012.

[20] E. H. T. project, “Deliverable D 2.6. Final Test Scenario and Test Specifications,” 2018.

[21] B. Pernici, “Infrastructure and Design for Adaptivity and Flexibility,” Mobile Information Systems, 2016.

[22] J. Nielsen, “Response Times: The Three Important Limits,” *Usability Engineering,* 1993.

[23] N. Alliance, “Definition of the testing framework for the NGMN 5G Pre-commercial Netwrok Trials,” Enero 2018. [Online]. Available: 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.