Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 2352941, 11 pages
https://doi.org/10.1155/2019/2352941

Research Article

A Quality of Experience Management Framework for Mobile Users

María Jesús Algar,[1] **Isaac Martín de Diego**,[1] **Alberto Fernández-Isabel**,[1] **Miguel Ángel Monjas**,[2] **Felipe Ortega**,[1] **Javier M. Moguerza**,[1] **and Hektor Jacynycz**[1]

[1] *Data Science Laboratory at Universidad Rey Juan Carlos, Spain*
[2] *Technology and Innovation, Ericsson, Spain*

Correspondence should be addressed to Isaac Martín de Diego; isaac.martin@urjc.es

Received 10 July 2018; Revised 16 November 2018; Accepted 13 December 2018; Published 1 January 2019

Academic Editor: Oscar Esparza

Voice transmission is no longer the main usage of mobile phones. Data transmissions, in particular Internet access, are very common actions that we might perform with these devices. However, the spectacular growth of the mobile data demand in 5G mobile communication systems leads to a reduction of the resources assigned to each device. Therefore, to avoid situations in which the Quality of Experience (QoE) would be negatively affected, an automated system for degradation detection of video streaming is proposed. This approach is named *QoE Management for Mobile Users* (*QoEMU*). *QoEMU* is composed of several modules to perform a real-time analysis of the network traffic, select a mitigation action according to the information of the traffic and to some predefined policies, and apply these actions. In order to perform such tasks, the best Key Performance Indicators (KPIs) for a given set of video traces are selected. A *QoE Model* is trained to define a global QoE for the set of traces. When an alert regarding degradation in the quality appears, a proper mitigation plan is activated to mitigate this situation. The performance of *QoEMU* has been evaluated over a degradation situation experiments with different video users.

1. Introduction

During the last decade mobile networks have witnessed a continuous growth in traffic, due to the increasing number of connected devices. This growth has accelerated with the takeoff of the Internet of Things (IoT), with an estimation of 30 billion connected devices by 2020 [1].

Telephone cells have limited resources in terms of simultaneous voice connections and data bandwidth. Such resources are shared among all devices connected to the same cell. When this limit is reached, the cell cannot provide service to all connected devices leading to a congestion situation. The result is a reduction of the bandwidth assigned to each device and the impossibility for new devices to attach to the cell and, in some cases, the device may even become unable to access any data service.

In any environment, it is common to find cells with severe congestion problems while their neighbor cells are underutilized. Mobility Load Balancing (MLB) [2] is a widely used technique to balance the network load. The overload of some cells is transferred to other less congested ones to achieve a more efficient load distribution in the mobile network. This technique is used in Global System for Mobile communications (GSM) [3], Universal Mobile Telecommunications System (UMTS) [4], and Long Term Evolution (LTE) [5]. However, the use of MLB in LTE presents important limitations (see, e.g., [6]).

Mobile networks have evolved to support very high-speed data transmission, since the new generation of mobile users not only demands telecommunication services but especially data-communication services. Cisco's Visual Networking Index (VNI) Global Mobile Data Traffic Forecast predicts that mobile data traffic will grow at an annual rate of 47% from 2016 to 2021, reaching 49.0 Exabytes per month by 2021. Video traffic, which currently accounts for 60 per cent of the total mobile traffic, is predicted to reach 78 per cent by 2021 [7].

Network operators design their network capacity according to traffic estimations for resource provisioning. However, with the arrival of 5G mobile communications [8], it is expected that these networks will support a variety of applications that will increase the demand of bandwidth significantly [9]. When the connectivity is lower than a tolerable threshold, the network traffic will pause or slow down. This degradation could greatly impact the user perceived quality, also known as the Quality of Experience (QoE) [10]. In this work, the following definition of QoE from the EU Qualinet community (COST Action IC1003: “European Network on Quality of Experience in Multimedia Systems and Services) encompasses the discussed aspects and defines it as the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the users personality and current state. In the context of communication services, QoS is influenced by service, content, device, application, and context of use” [11].

The Online Network Traffic Characterization (ONTIC) project proposes to design, develop, and evaluate a novel architecture and a set of data mining techniques to characterize network traffic data streams, identifying traffic pattern evolution and proactively detecting anomalies in real time [12]. ONTIC has defined several use cases to address the network transformation. One of them is the *Adaptive QoE Control*, which implements an analytics-enhanced control loop to react to QoE degradation situations in video services and apply corrective measures [13].

The aim of this work is to consolidate an automated framework to prevent degradation situations in mobile networks. This framework will be named *Quality of Experience Management for Mobile Users* (*QoEMU*). We focus on video streaming domains where modeling distortions is a very complex issue [14]. The main goal of *QoEMU* is to detect problematic scenarios and react applying different actions to mitigate adverse effects, ensuring a minimum overall degradation of the QoE. It is known that the QoE depends on the end user’s device [15]. In this work, mobile phones were considered.

This paper is organized as follows. Section 2 presents the proposed framework in the context domain. Section 3 describes in detail the proposed framework, introducing all components and their functionality. Results from two empirical experiments to test the approach are shown in Section 4. Finally, Section 5 summarizes the conclusions of this work.

2. Background

As mentioned, one of the main problems of mobile networks is congestion situations, since they can even cause users disconnections. To address this issue, a possible solution could be readjust one of assigned resources. For instance, a scheduling based on a search scheme is presented in [16]. This approach finds the suitable number of combinations of antennas and user equipment.

Due to the network evolution from 4G to 5G it is also essential to facilitate the management of scenarios of growing network complexity [17]. Recent studies have focused on network dimension. In [18] a prediction method based on cluster analysis is proposed, whose main goal is to obtain general information about the multidimensional situation of LTE traffic to assign resources more efficiently. The network can be analyzed using only time-dependent data, applying k-means, and Support Vector Machine (SVM) methods [19]. In addition, useful information about network future behavior can be given to plan optimal access.

Other researches have centered their attention in the relation between spatial and temporal information [20, 21]. They usually apply Machine Learning (ML) techniques to combine this temporal and spatial information to improve prediction accuracy in any cellular network, even in noisy or challenging cases.

Regarding the QoE, a framework to model the mobile Internet network QoE using Big Data is proposed in [22]. Up to now, parameters such as bandwidth, loss, and delay were sufficient to evaluate Quality of Service (QoS) [23]. However, these metrics are not sufficient to describe a good quality level for real-time multimedia applications, as it is necessary to capture subjective aspects such as user satisfaction [24]. Many researches develop QoE metrics using explicit functions, which take into account parameters from the encoder or the network [25, 26]. In [27] a framework called Post Streaming Quality Analysis (PSQA) is developed to improve the QoE of dynamic adaptive streaming over HTTP video streaming over mobile networks. PSQA uses ML to generate the best adaptation logic parameters from past throughput traces training. In [28] a QoE model for HTTP adaptive streaming is developed. Parts of this model have become standardized in ITU-T P.1203.3.

In the case of Ericsson, it has introduced a highly automated and programmable architecture, the COMPA architectural model [29]. This approach consists of an evolution of classic autonomic networks. It uses control loops to obtain data from a variety of sources (for instance, network traffic in real-time situations), making automatic corrective decisions such as modifying the assigned bandwidth (depending on the analysis of these data) and executing them. The COMPA principles have been incorporated to current 5G standardization efforts such as the Open Network Automation Platform (ONAP) [30]. Figure 1 shows the COMPA model, including three different functions:

(i) A *(Analytics)* takes care of turning data into information and insights that serve as a basis for decision making and triggering actions.

(ii) P *(Policy)* is a function that governs the behavior of a telecommunication system based on the insights from A, asking the COM function to implement the desired behavior.

(iii) COM is a function that enforces the decisions made by the Policy function, including three components:

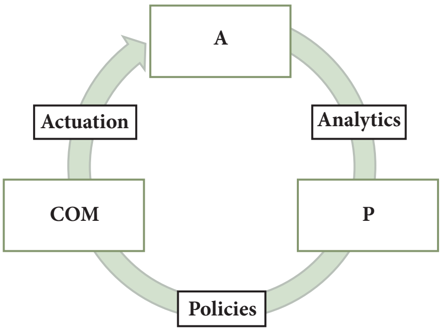


Figure 1: COMPA architectural model.

C *(Control)*: in charge of negotiating, establishing, maintaining, and terminating data/user plan connections.

O *(Orchestration)*: in charge of the organization of the loop, its deployment and automation, coordination, and communication between components and services.

M *(Management)*: it coordinates the efforts to accomplish goals and objectives using available resources efficiently and effectively.

3. The QoEMU Framework

The main goal of *QoEMU* is to detect situations suitable of deteriorating the QoE in mobile networks and apply different mitigation actions to alleviate the QoE degradation situation. Different types of mitigation actions can be adopted, for example: bandwidth regulation; access technology steering (i.e., from 4G to 3G), or different services (such as video service, web navigation, etc.) can be blocked [31]. An important point to highlight is the impossibility of simply providing more bandwidth to users. The overall available bandwidth in a cell is limited and therefore only redistribution is feasible.

Different options could be considered at this point (see, e.g., [32]). In this case, QoE degradation alleviation must deal with enhancing the average QoE, even if some of the users do not receive a better service. The proposed framework, presented in Figure 2, uses the COMPA architecture introduced in Section 2 and follows, whenever possible, the Kappa architecture paradigm, using a message broker as main communication means between the architecture modules [33].

First, a *Video Processor* module, which is a streaming video server based on *VLC* that provides video to users. Video traffic goes through the *Virtual Proxy* (the COM module). This module redirects traffic to the *VLC Video Server* component. It also collects network traces from the users traffic and is able to regulate the available bandwidth for users to play the video, based on the mitigation plans provided by the *Policy Manager* module. The *Message Broker* module, which is based on Apache Kafka [34], is used to communicate the traces from the *Virtual Proxy* to the *Degradation Detection* module. This component is based on Apache Spark Streaming [35] and works as a time-based sliding window operator [36] that continuously receives batches of information (network traces in this case). Thus, the *Degradation Detection* module accumulates the traces during the window duration (several batches). Then, it processes the accumulated traces to obtain the Key Performance Indicators (KPIs) [37] and QoE metrics based on the *QoE Model* (i.e., the ML model previously trained to organize the traces which is used by *QoEMU* to detect degradation situations). The *Policy Manager* module is in charge of receiving the insights from the *Degradation Detector* module to decide the mitigation plans that must be executed, according to a set of predefined policies. Such mitigation plans are communicated to the *Virtual Proxy* module to solve the detected degradation situations.

In Section 3.1 the architecture of the *QoEMU* framework is presented in detail. Then, the architecture built to generate the *QoE Model* is addressed in Section 3.2.

3.1. QoEMU Framework Architecture.

In accordance with the COMPA model, the *QoEMU* architecture consists of three main modules: the *Degradation Detector*, the *Policy Manager*, and the *Virtual Proxy*.

Degradation Detector. This module supports to the A function in the COMPA model. It is in charge of the real-time analysis of video traffic data from users connected to the network. Regarding the architecture of the *Degradation Detector* module, it consists of four components: the *KPI Selector*, the *QoE Analyzer*, the *Alert Notifier,* and the *QoE Model*. This latter is trained using a specific architecture that includes the *QoE Model Generator System* module. This architecture is explained in detail in Section 3.2 and its scheme is shown in Figure 4.

First, the *KPI Selector* extracts the KPIs from the collected network traces. Next, the *QoE Analyzer* uses a nonsupervised ML algorithm to split up the dataset in heterogeneous groups. The main idea is to collect traffic data with similar QoE in the same cluster and compute an average QoE. The *Degradation Detector* module is able to analyze the QoE of a set of traces collected during a time window in real time and detect a degradation situation. When the average QoE of the traces in a time window is under a threshold (δ_{deg}), a degradation situation is considered to occur. Then, the *Degradation Detector* module sends an alert to the *Policy Manager* module. The alert includes the average QoE and the location where the degradation has been detected. When an alert about a degradation situation is sent to the *Policy Manager*, a session is created in such module. Since it is possible to detect multiple degradation situations in different locations, they will be treated by the *Policy Manager* in different sessions. While a degradation situation in a given location persists, the *Degradation Detector* module keeps on periodically sending information about the evolution of the average QoE. However, it is stateless and therefore not aware of the sessions kept by the *Policy Manager*.

Policy Manager. This module supports the P function in COMPA. It consists of two components: the *Policy Engine* and the *Degradation Sessions Database*. The first one makes

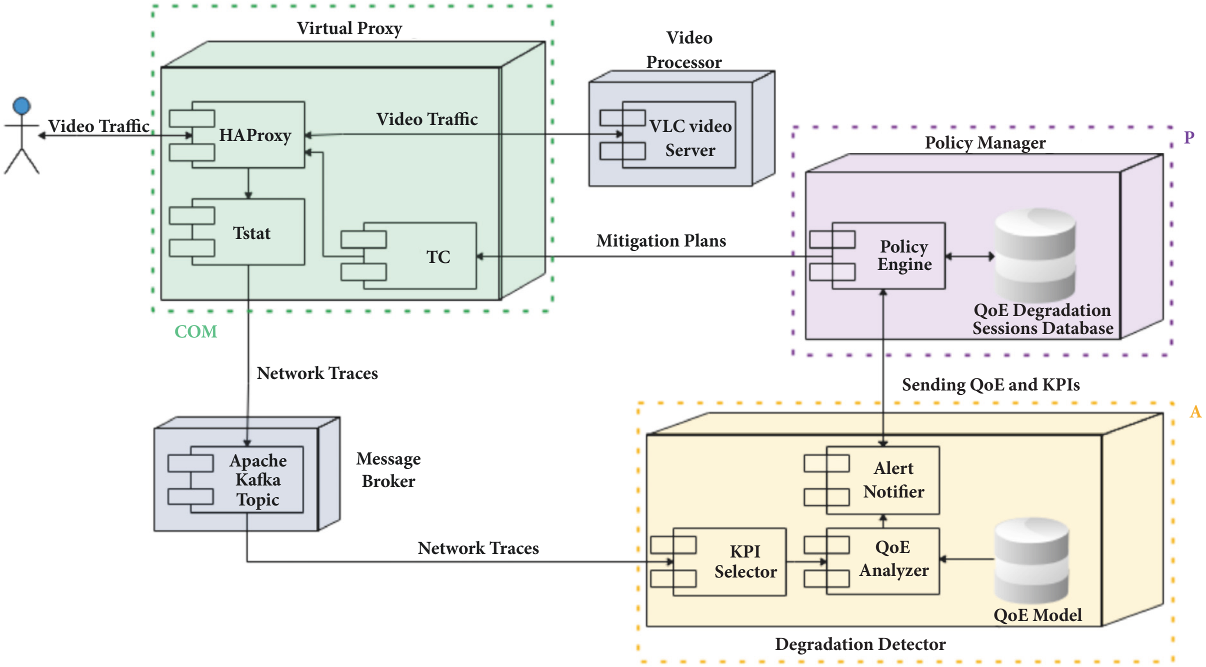


Figure 2: *QoEMU* framework.

decisions based on the alerts received from the *Degradation Detector* and decides which mitigation actions have to be enforced by the COM function. The *Degradation Sessions Database* maintains information of the active degradation sessions (mobile devices IPs). A search is performed on the database when an alert from the *Degradation Detector* is received. If a degradation session is located at the same location, it means that the degradation continues and has already been alerted (has an associated session). According to its policies, the *Policy Engine* may decide to update the mitigation plans being enforced.

During the conversation between the *Degradation Detector* and the *Policy Manager* modules, two different situations can be distinguished:

(i) If the average QoE is greater than or equal to a threshold ($\geq \delta_{deg}$), it could mean two different things. If there is a degradation session in the same location, this means that the degradation situation has just finished. On the other hand, if the locations are different the QoE is directly considered as satisfactory.

(ii) If the average QoE is strictly lower than a threshold ($< \delta_{deg}$) and a degradation session exists in the same location, the degradation continues and the *Policy Manager* module has been alerted. In contrast, a new degradation situation has been detected and an alert is sent to the *Policy Manager* module. At this point, this module generates a session identifier and stores information about the date, the time, the degradation location, and the average QoE. When a session begins, a specific mitigation plan is selected for that session.

The message exchange between the *Degradation Detector* and the *Policy Manager* is based on a Representational State Transfer (REST) interface [38]. The *Policy Manager* only provides REST interfaces and, therefore, cannot be directly plugged to the *Message Broker* component. As the *Degradation Detector* plays the role of a message producer according to a Kappa architecture, an adapter (the *Policy Manager Adapter*) is required. It plays a message consumer role when receiving alerts from the *Degradation Detector* while, at the same time, implements a REST client interface towards the *Policy Engine* component.

Figure 3 presents an example of the messages between the *Degradation Detector* module (A), and the *Policy Manager* module (P), when a degradation situation is detected. For the sake of simplicity, it is represented as an end-to-end REST exchange.

Virtual Proxy. This module supports the COM function in COMPA. It is aware of what type of user each IP address belongs to, so that it can apply different policies. It implements two main tasks. The first one handles the traffic towards the *VLC Video Server*, captures traffic information (network traces), and forwards it to the *Degradation Detector*. The second executes the mitigation actions invoked by the *Policy Manager* to alleviate the degradation.

Regarding the architecture of this module, it consists of three software components: *HAProxy*, *TC* (*Traffic Control*), and *Tstat* [39]. Once the *Policy Manager* has decided which mitigation actions have to be enforced, they are communicated to the *Virtual Proxy* for being applied. The *HAProxy* provides HTTP and TCP proxy capabilities. It allows, among

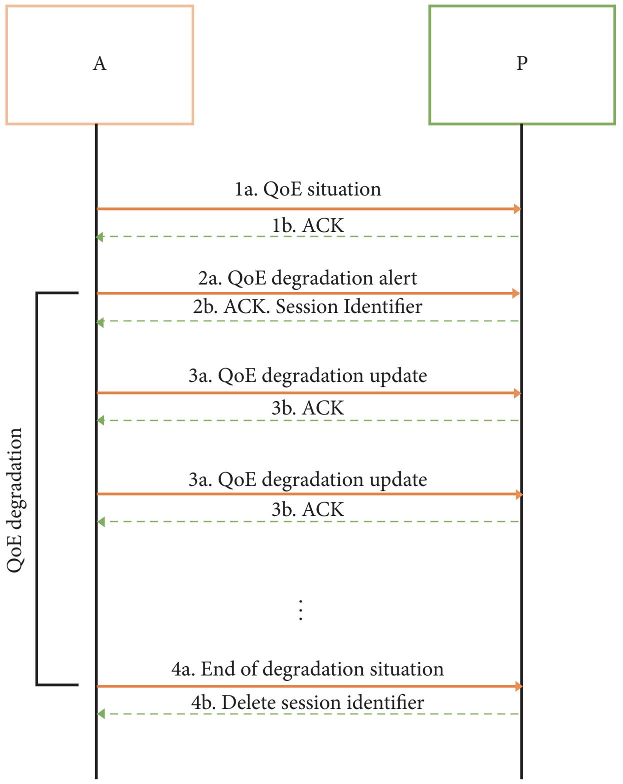

A
P
1a. QoE situation
1b. ACK
2a. QoE degradation alert
2b. ACK. Session Identifier
3a. QoE degradation update
3b. ACK
3a. QoE degradation update
3b. ACK
⋮
4a. End of degradation situation
4b. Delete session identifier
QoE degradation

Figure 3: Sequence diagram for a degradation situation.

other things, a high availability and load balancing [40]. It is in charge of rerouting the incoming HTTP traffic to the Video Server to enable the capture of network traces.

Tstat is an automated tool for passive monitoring [41]. It provides traffic monitoring up to Gigabits per second using off-the-shelf hardware and offers a passive sniffer functionality of traffic. Additionally, it is capable of analyzing these traces to provide multiple KPIs, concerning connections, at both the network and transport level [37].

The *TC* configures the Linux kernel packet scheduler which acts as a bandwidth regulator. It allows controlling the way packets are handled in network interfaces. The main feature, as per the *Virtual Proxy* which is involved, is the rate of handled packets. Thus, it is possible to characterize traffic and create rules to assign a bandwidth for each type of service or range of IPs [42]. A bandwidth limitation functionality will be used when needed for video users connected to the *Virtual Proxy*.

When the *Policy Manager* decides which mitigation actions have to be enforced and creates a degradation session, the *Virtual Proxy* module is informed to implement the mitigation plan. Message exchange between them is based on REST as well. The mitigation plan activation orders sent to the *Virtual Proxy* should include information about the start and end of the plan, the mitigation actions to perform for each type of user and the location in which the plan is going to be applied.

After applying a mitigation plan, the *Degradation Detector* module keeps on receiving network trace information and an average QoE is continuously computed. The evolution of this indicator is received by the *Policy Manager*, which evaluates whether the degradation situation continues. If that is the case, it informs the *Virtual Proxy* module to carry on the mitigation plan. However, if the *Policy Manager* module detects that the degradation has finished, it informs the *Virtual Proxy* module to cancel the plan and discards the degradation session record in its database.

3.2. QoE Model Generator System. As mentioned before, the *Degradation Detection* module of the *QoEMU* framework (see Figure 2) uses the *QoE Model*. It implements a ML model able to discriminate between network traces and compute an average QoE. This component has been built using the architecture presented in Figure 4.

A controlled experiment with several users is performed to generate the *QoE Model*. It uses an alternative architecture that comprises three modules: the *Virtual Proxy*, the *Video Local Server,* and the *QoE Model Generator System*. Notice that this architecture is not used by the *QoEMU* framework since it is only in charge of generating the *QoE Model*.

Firstly, several network traces from traffic generated when videos are watched in devices are collected. User's traffic goes through the *Virtual Proxy* module, where *Tstat* monitors the traffic to obtain traces. Secondly, the *Traffic Control (TC)* is used to generate random disturbances for each user. The objective pursued is to cause different congestion situations to obtain information about the degradation situations produced. As a result, the user has to label the QoE of the service, that is, they will be asked about their quality of experience. In the *Label Provider* component, each user evaluates three possible QoE levels: *Good*, *Medium*, and *Bad*. Then, this information is combined with the traffic traces generated by the user.

For the experiment, a total of 328, 708 video traces were obtained using the method presented above. Notice that *Tstat* generates 114 different variables. Thus, before using the *Cluster Generator* component to group the traces, it is necessary to filter the data and select those variables with relevant information. The *KPI Selector* component is used for this purpose.

Following the state-of-the-art for QoE [43, 44], the ensuing variables are selected according to the domain knowledge:

(i) PCK.ALL (*s_pkts_all* according to *Tstat*): total number of packets sent by the video server during the connection.

(ii) PCK.RES (*s_pkts_retx*): number of packets retransmitted from the server.

(iii) PCK.UNK (*s_pkts_unk*): number of packets not in sequence or duplicated.

(iv) RTT (*s_rtt_avg*): round-trip time. This is the time needed for a package to be sent plus the time needed for an acknowledgment of the package to be received.

(v) BYTES (*s_bytes_all*): total number of bytes sent.

(vi) DURATION (*durat*): timestamp of the connection.

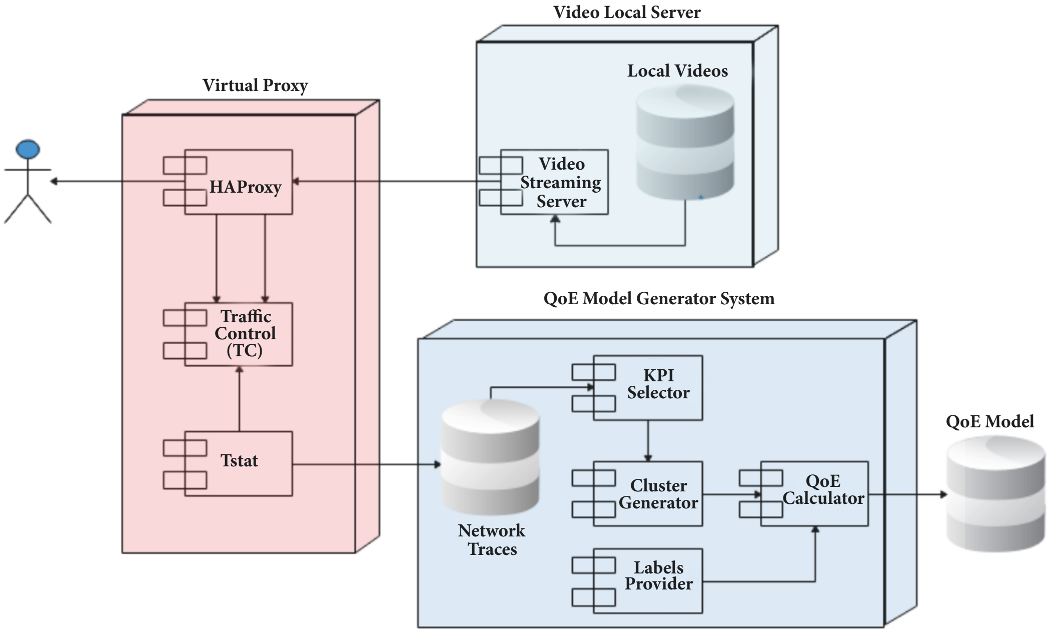


Figure 4: QoE Model Generator System.

Table 1: KPIs obtained in the data analysis.

Variable	Expression
RMP	PCK.RES/PCK.ALL
RUP	PCK.UNK/PCK.ALL
LATENCY	RTT
BANDWIDTH	BYTES/DURATION

From these variables, the following KPIs are defined (see Table 1):

(i) Ratio of Missing Packets (RMP): percentage of missing, and forwarded, packets during connection.

(ii) Ratio of Unknown Packets (RUP): percentage of packets not in sequence or duplicated which are not classified as specific events.

(iii) LATENCY: the average RTT.

(iv) BANDWIDTH: the data transmission rate, number of bytes per second that have been transmitted during the connection.

The purpose of these KPIs was as follows. RMP is necessary to being able to detect possible problems experienced by users. It is strongly related to QoE. RUP allows to detect possible packets that arrive later or disordered to the destination. As expected, it is related to situations of degradation where the packets are buffered and then suddenly liberated. LATENCY is the difference between a stimulus and the response to it. In this case, it consists of the average time a packet needs to go from the initial point to the destination. It allows to detect possible delays that affect the QoE. Finally, BANDWIDTH provides information about the performance of the connection. If it is significantly reduced, it could negatively influence the QoE.

Table 2: Summary (mean and standard Deviation) of the KPIs obtained in the data analysis.

Variable	Mean	Sd
RMP	0.18	0.14
RUP	0.12	0.10
LATENCY	3	5
BANDWIDTH	109	83

A dataset of more than 300, 000 traces, four KPIs and the corresponding QoE label from the users, are obtained. The summary of the collected KPIs is presented in Table 2.

Despite the subjective response of the users (an opinion about the QoE), an unsupervised learning method has been selected. In the *Cluster Generator* component, a *k-means* clustering algorithm [45] was trained to group the traces based on the *Euclidean* distance from the KPIs (normalized between 0 and 1). If the obtained clusters are informative, there will be a relationship between the QoE from the user and the labels in each cluster. The classical Elbow method is used to choose the optimal number of clusters in the model [46]. According to this technique, 4 clusters have been selected.

Cluster 0 is defined by high values of *RMP* and *RUP* variables. Cluster 1 is defined by high values of *RMP* and low values of *BANDWIDTH*. Cluster 2 is defined by low values of *RMP*, *RUP*, *LATENCY*, and *BANDWIDTH*. Cluster 3 is defined by high values of *BANDWIDTH*, and low values for the other variables. Thus, clusters 2 and 3 should be associated with high values of QoE, and clusters 0 and 1 should be associated with low values of QoE. At this point,

Table 3: Label distribution and QoE for each cluster in the model.

Cluster	Good	Medium	Bad	QoE
0	0.02	71.17	28.80	0.44
1	0.65	46.31	53.03	0.40
2	89.99	6.72	3.28	0.85
3	99.70	0.00	0.28	0.90

the unsupervised cluster technique is completed with the labels from the users. In clusters 2 and 3, the label *Good* is predominant. However, in clusters number 0 and 1 there is a mix between labels *Medium* and *Bad*.

Following the labels assigned by the users, a global QoE for each of the generated clusters is calculated. In order to produce such a value, a score is assigned to each label following the eleven-grade numeric quality scale proposed by [47]. *Good* is scored as 0.9, and *Bad* is scored as 0.3. The *Medium* could be scored as the mean value in the range $[0.3, 0.9]$, that is, 0.6. However, in order to bring on early detection of QoE degradation situations, *Medium* is scored as 0.5 (closer to *Bad* score than to *Good* score).

The users on charge of labeling were illustrated with the meaning of the label before watching the videos.

Table 3 presents the QoE for each cluster as the weighted average of the labels distribution in the cluster. For instance, the distribution of *Good*, *Medium*, and *Bad* traces in cluster 0 is 0.02%, 71.17%, and 28.80%, respectively. Thus, the global QoE for cluster 0 is calculated as

$$0.90 * 0.0002 + 0.50 * 0.7117 + 0.30 * 0.2880 = 0.44. \quad (1)$$

The global QoE of clusters in Table 3 will be used to define the model that will perform the QoE estimation of new video traces. That is, once the training phase has finished, this model is able to achieve the QoE of any video service.

3.2.1. QoE Calculation. The cluster model is a list with the centroids of the four clusters and their global QoE. In order to calculate a prediction for a new trace, the nearest centroid is calculated based on the *Euclidean* distance between the KPIs of the new trace and the centroids.

Let V be a set of T video traces: $\{t_i\}_{i=1}^{T}$. First, the corresponding cluster ($C(t_i)$) for each trace t_i in V is calculated using the KPIs in the trace looking for the nearest cluster centroid. For example, let one consider trace number 1. If the nearest centroid to the KPIs corresponding to trace number 1 is the centroid in cluster 0, then $C(t_1) = 0$. Next, the QoE for trace t_i ($QoE(C(t_i))$ is calculated as the QoE of the associated cluster using values in Table 3. Thus, the QoE assigned to trace number 1 will be $QoE(C(t_1)) = QoE(0) = 0.44$. Finally, it is possible to calculate the QoE for the complete set of video traces V as the average of the QoE of all the traces in the set:

$$QoE(V) = \frac{1}{T}\sum_{i=1}^{T} QoE(C(t_i)). \quad (2)$$

Table 4: Distribution of the two sets of traces from the *DellEMC* in the clusters provided by the *QoE Model*.

Cluster	Set 1	Set 2
0	0.13	0.14
1	0.45	0.53
2	0.07	0.09
3	0.35	0.24

4. Case Study

This section details the evaluation of the proposal to illustrate its viability. It consists of two experiments. The first one, presented in Section 4.1, addresses the detection of QoE degradation situations using the *QoE Model* for different sets of traces. The second experiment, detailed in Section 4.2, presents the complete process achieved by the *QoEMU* framework in order to detect a QoE degradation situation and illustrate how the system reacts activating a proposed mitigation plan to solve the situation. In both cases the degradation threshold δ_{deg} has been fixed at 0.5.

4.1. QoE Model Evaluation. In this experiment the *QoE Model* is evaluated to illustrate its capabilities to calculate the QoE value and detect possible degradation situations. For this purpose, two sets of 10,000 traces from the *DellEMC* dataset [48] have been selected. Notice that this new dataset has made possible to calculate the needed KPIs to feed the *QoE Model*.

For the first set, any of the traces is assigned to the nearest centroid using the *Euclidean* distance. The traces were mainly distributed between the clusters 1 and 3. The distribution of traces in the clusters is: 0.13, 0.45, 0.07 and 0.35, respectively (see Table 4). Thus, the resulting QoE value was $0.13 * 0.44 + 0.45 * 0.4 + 0.07 * 0.85 + 0.35 * 0.90 = 0.61$. This implies that no degradation situation was detected by the *QoE Model*. Then, it could be said that the selected traces present a normal behavior and no mitigation plan is needed at this point.

For the second set, the traces were distributed similarly to the previous one (i.e., clusters 1 and 3 receives most traces). The distribution was: 0.14, 0.53, 0.09 and 0.24 respectively for each one of the clusters. Thus, the resulting QoE value was $0.14 * 0.44 + 0.53 * 0.4 + 0.09 * 0.85 + 0.24 * 0.90 = 0.57$ (see Table 4). This implies that no degradation situation was detected by the *QoE Model*. Nevertheless, the obtained values could be a starting point for a possible degradation situation as it is close to the predefined threshold. However, the most common situation could be a temporary random problem related to the bandwidth.

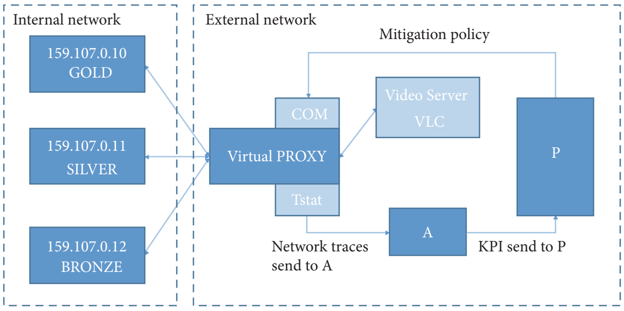


Figure 5: Excerpt of the scenario scheme in the experiment.

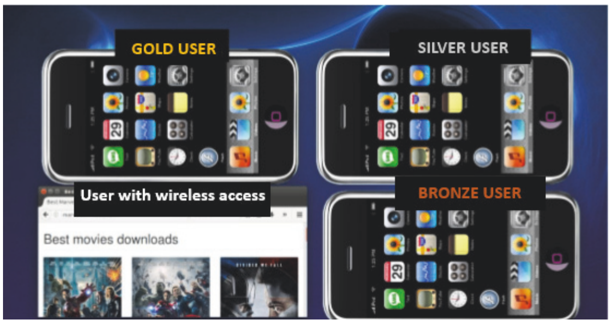


Figure 6: Users connected to an IP direction.

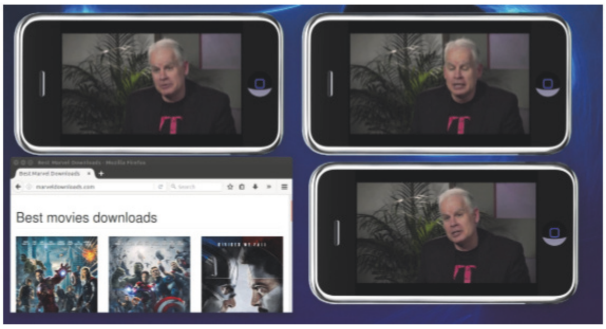


Figure 7: The three users watching the video with similar quality.

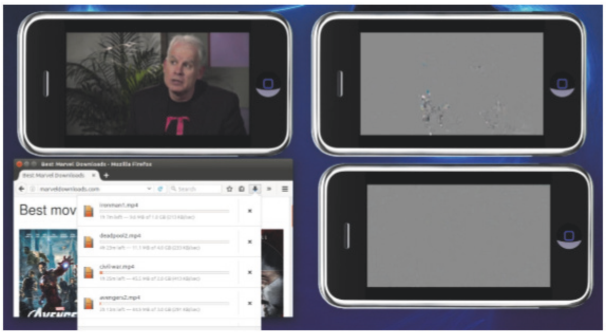


Figure 8: Users suffering a degradation.

Thus, the *QoE Model* has been tested using a different dataset of that used to train it. It was able to evaluate the QoE value detecting normal behavior and possible degradation situations in traces.

4.2. Complete Framework Evaluation. In order to evaluate the performance of the complete *QoEMU* framework, a mobile experiment has been carried out in a laboratory, in which three users watching a video in their devices, connected to different servers through a switch, are monitored. The scheme of the scenario is presented in Figure 5.

The profiles of the users are defined based on the priority they have on resource sharing in degradation situations. The category of each user is different: *Gold*, who is a premier user whose connection should be as good as possible most of the time; *Silver*, who is an user with a medium quality type of connection; and *Bronze*, who is an user with the worst type of connection that could be penalized to improve the global performance. Each user is connected to an IP address and receives the video streaming as shown in Figure 6.

In this particular case, the mitigation plan designed consists of a limitation of the user bandwidth: *Gold* is granted 3Mbps, *Silver* is granted 1Mbps and *Bronze* only receives 64Kbps. With this mitigation action, the *Gold* user will watch the video normally; the *Silver* user will experience some difficulties, although, in general, the video can be properly watched; and the *Bronze* user will have frequent problems to visualize the video.

The *QoE Analyzer* component in the *Degradation Detector* module (see Figure 2) estimates the QoE using a time-based sliding window operator. The window duration is 30 seconds and the slide duration is 10 seconds (see Section 3). That is, the QoE calculation (as explained in Section 3.2.1) is achieved in 10-second batches of traces and the window considers the last 30 seconds of traces (i.e., the current batch together with the two previous batches). An alert will be generated when the QoE value of the traces are under the previously fixed degradation threshold (i.e., $\delta_{deg} = 0.5$).

When users begin to reproduce the video, the traffic traces are sent to the *Degradation Detector* module to analyze them and obtain the corresponding KPIs, which represent the global QoE of each set of traces generated by users. In the beginning, visualization is perfect and each user enjoys watching the video. This situation is represented in Figure 7, and at first 50 seconds in Figure 10.

After 45 seconds, an artificial congestion in the traffic is generated. Thus, a limit in the bandwidth is applied to all the users. This action generates several problems and, consequently, users start to have troubles in their visualizations. The QoE decreases, as shown in Figures 8 and 10.

Next, this action implies a reduction in the QoE given by the *Degradation Detector* module under the fixed threshold (δ_{deg}). This happens 60 seconds after the start of the video.

When this information arrives at the *Degradation Detector* module, it detects a degradation situation and generates an alert. The alert is received by the *Policy Manager* module which, according to the policies it has configured, chooses a mitigation plan to alleviate the undesired situation. The plan comprises bandwidth regulation actions, which are sent to the *Virtual Proxy* module, responsible for applying it.

FIGURE 9: The mitigation plan has ended and all the bandwidth of the users will be restored.

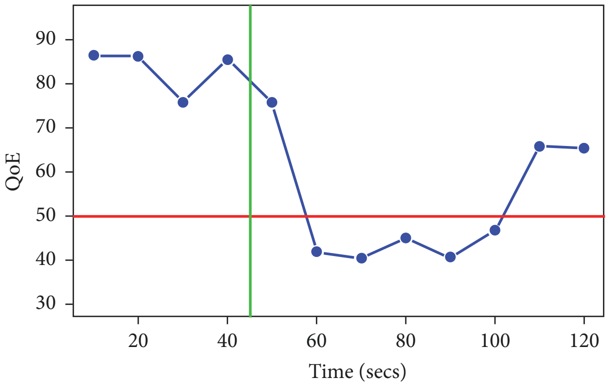


FIGURE 10: QoE for the traces in the experiment. Green line indicates the start of the artificially generated congestion. Red line indicates the QoE threshold.

When the degradation situation has finished, the mitigation action is deactivated and the three users watch the video without issues, as shown in Figure 9. This happens 110 seconds after the start of the video (see Figure 10).

Thus, a complete experiment using *QoEMU* framework has been developed applying specific mitigation actions to solve a congestion traffic issue.

Notice that the global performance of the system relies on two main configuration parameters: the threshold δ_{deg} for the QoE and the mitigation policy action. In this case, the threshold was fixed to 0.5. However, if a higher value was selected (for instance 0.75), the artificial degradation would be detected earlier. This implies increasing the risk of false alarms (false detection of degradation situations). Since the bandwidth regulation action mainly affects *Bronze* users, they will have a lot of problems to visualize the video, that could be avoided with a proper system configuration. Regarding the mitigation plan, it depends on the network capabilities. Thus, if it would be possible to grant a high bandwidth for each user profile (equals to the applied one for the *Gold* users), it is clear that the system will be restored earlier. However, this plan depends mainly on the business domain model.

5. Conclusions

One of the main problems of mobile networks are QoE degradation situations. In this paper, the *QoEMU* framework has been presented and tested. *QoEMU* is an automated system for the analysis of network traffic, the detection of degradation in the Quality of Experience of user, and the generation and application of mitigation actions to alleviate adverse effects. All these tasks are achieved using an architecture based on the COMPA model.

The architecture of *QoEMU* implies three main modules. The *Degradation Detector* module, that corresponds to the *Analytics* module in COMPA, generates alerts based on the *QoE Model* previously generated. The *Policy Manager* module corresponds to the *Policies* module in the COMPA architecture. The *Virtual Proxy* module corresponds to the *Control, Orchestration and Management* function in the COMPA model. It executes the mitigation actions to solve the detected degradation.

In order to train the *QoE Model*, a specific architecture has been used. An experiment with more than 300,000 labeled traces was performed. Using ML techniques, the *QoE Model* has been able to predict the QoE of a set of traces.

The *QoEMU* framework has been tested with several video users in laboratory experiments. A degradation in the QoE has been generated based on an artificial congestion in the traffic. The *QoEMU* framework was able to detect the generated degradation and manage it through an adequate mitigation action. Thus, a mitigation plan was designed based on the different profiles. Once the degradation situation was corrected, the plan was deactivated.

As future work, the presented *QoE Model* could be upgraded by including the opinions of a higher number of experts. In such a case, a method to integrate several opinions into a unique ground truth should be considered [49, 50]. In order to implement the *QoEMU* framework in production, different test batteries are mandatory (e.g., stress testing) and a cooperation between the Over-The-Top (OTTS) and Internet Service Providers (ISPs) is needed [51]. Furthermore, a graphical real-time statistics tool will be implemented. For this purpose, a server will be deployed to predict, in real time, the value of the QoE and communications between different components. It could be also interesting to measure the predictive capabilities of the system. In addition, more specific distances among KPIs, different to the *Euclidean* distance, could be considered for the calculation of the QoE.

Data Availability

With regard to the datasets involved in this paper, they cannot be openly provided, as they belong to Ericsson Intellectual property, and our policy does not allow undisclosing it.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Research was supported by grants from European Union under FP7 Grant Agreement no. 619633 (ONTIC Project) and the Spanish Ministry of Economy and Competitiveness, under the Retos-Colaboración program: UNIKO (Ref: RTC-2015-3521-7) and PPI (Ref: RTC-2015-3580-7).

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] T. Yamamoto, T. Komine, and S. Konishi, "Mobility load balancing scheme based on cell reselection," in *Proceedings of the Eighth International Conference on Wireless and Mobile Communications (ICWMC '12)*, pp. 1–7, 2012.

[3] M. Toril and V. Wille, "Optimization of handover parameters for traffic sharing in GERAN," *Wireless Personal Communications*, vol. 47, no. 3, pp. 315–336, 2008.

[4] J. Li, C. Fan, D. Yang, and J. Gu, "Umts soft handover algorithm with adaptive thresholds for load balancing," in *Proceedings of the 62nd Vehicular Technology Conference, VTC 2005*, pp. 2508–2512, USA, September 2005.

[5] S. Sesia, I. Toufik, and M. Baker, *LTE-The UMTS Long Term Evolution: Form Theory to Practice*, Wiley, 2011.

[6] J. M. Ruiz-Aviles, M. Toril, S. Luna-Ramirez, V. Buenestado, and M. A. Regueira, "Analysis of Limitations of Mobility Load Balancing in a Live LTE System," *IEEE Wireless Communications Letters*, vol. 4, no. 4, pp. 417–420, 2015.

[7] CISCO, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021, white paper," https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html.

[8] J. G. Andrews, S. Buzzi, and W. Choi, "What will 5G be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[9] C. H. Liu and J. Fan, "Scalable and efficient diagnosis for 5G data center network traffic," *IEEE Access*, vol. 2, pp. 841–855, 2014.

[10] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the quality of experience of HTTP video streaming," in *Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management, IM 2011*, pp. 485–492, Ireland, May 2011.

[11] K. Brunnstr, S. A. Beker, K. De Moor et al., "Qualinet white paper on definitions of quality of experience," COST Action IC 1003, 2012.

[12] Online Network Traffic Characterization: ONTIC, 2014, http://ict-ontic.eu.

[13] A. Bascuñana, M. A. Monjas, V. Martín, and F. Arias, "ONTIC Deliverable D5.5: Use Case 2 - Adaptive Quality of Experience Control," Online Network Traffic Characterization, 2017.

[14] C. G. Bampis, Z. Li, and A. C. Bovik, "Continuous prediction of streaming video QoE using dynamic networks," *IEEE Signal Processing Letters*, vol. 24, no. 7, pp. 1083–1087, 2017.

[15] M. T. Diallo, N. Marechal, and H. Afifi, "A hybrid contextual user perception model for streamed video quality assessment," in *Proceedings of the 15th IEEE International Symposium on Multimedia, ISM 2013*, pp. 518-519, USA, December 2013.

[16] Y. Arikawa, K. Kawai, H. Uzawa, and S. Shigematsu, "Practical resource scheduling in massive-cell deployment for 5G mobile communications systems," in *Proceedings of the International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2015*, pp. 456–461, Indonesia, November 2015.

[17] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 65–75, 2014.

[18] S. Zhang, D. Ma, J. Gao, J. Gu, and X. Zhang, "Poster: Cluster analysis based traffic prediction method from real traffic traces in LTE networks," in *Proceedings of the 9th International Conference on Communications and Networking in China, CHINACOM 2014*, pp. 646-647, China, August 2014.

[19] S. E. Hammami, H. Afifi, M. Marot, and V. Gauthier, "Network planning tool based on network classification and load prediction," in *Proceedings of the 2016 IEEE Wireless Communications and Networking Conference, WCNC 2016*, Qatar, April 2016.

[20] E. Lovisotto, E. Vianello, D. Cazzaro et al., "Cell traffic prediction using joint spatio-temporal information," in *Proceedings of the 6th International Conference on Modern Circuits and Systems Technologies, MOCAST 2017*, Greece, May 2017.

[21] U. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das, "Understanding traffic dynamics in cellular data networks," in *Proceedings of the IEEE INFOCOM 2011*, pp. 882–890, China, April 2011.

[22] A. W. Yusuf-Asaju, Z. B. Dahalin, and A. Ta'a, "Framework for modelling mobile network quality of experience through big data analytics approach," *Journal of Information and Communication Technology (JICT)*, vol. 17, no. 1, pp. 79–113, 2018.

[23] R. Stankiewicz, P. Cholda, and A. Jajszczyk, "QoX: What is it really?" *IEEE Communications Magazine*, vol. 49, no. 4, pp. 148–158, 2011.

[24] Series, B.: Methodology for the subjective assessment of the quality of television pictures. Recommendation ITU-R BT pp. 500–13 (2012).

[25] Y. Park, Y. Seok, N. Choi, Y. Choi, and J.-M. Bonnin, "Rate-adaptive multimedia multicasting over IEEE 802.11 wireless LANs," in *Proceedings of the 2006 3rd IEEE Consumer Communications and Networking Conference, CCNC 2006*, pp. 178–182, USA, January 2006.

[26] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Transactions on Broadcasting*, vol. 50, no. 3, pp. 312–322, 2004.

[27] Y. Liu and J. Y. Lee, "A unified framework for automatic quality-of-experience optimization in mobile video streaming," in *Proceedings of the IEEE INFOCOM 2016 - IEEE Conference on Computer Communications*, pp. 1–9, San Francisco, Calif, USA, April 2016.

[28] W. Robitza, M.-N. Garcia, and A. Raake, "A modular HTTP adaptive streaming QoE model - Candidate for ITU-T P.1203 ("P.NATS")," in *Proceedings of the 9th International Conference on Quality of Multimedia Experience, QoMEX 2017*, Germany, June 2017.

[29] G. Rune, E. Westerberg, T. Cagenius et al., "Architecture evolution for automation and network programmability," *Ericsson Review (English Edition)*, vol. 91, no. 2, pp. 46–54, 2014.

[30] "ONAP Project: Open Network Automation Platform, March 2017," https://www.onap.org/.

[31] A. Bascuñana, M. A. Monjas, M. Lorenzo, and P. Sánchez, "Adaptive Quality of Experience: A Novel Approach to Real-Time Big Data Analysis in Core Networks," in *Proceedings of the East European Conference on Advances in Databases and Information Systems*, pp. 235–242, Springer International Publishing, 2015.

[32] T. Hoßfeld, P. E. Heegaard, and M. Varela, "QoE beyond the MOS: Added value using quantiles and distributions," in *Proceedings of the 2015 7th International Workshop on Quality of Multimedia Experience, QoMEX 2015*, Greece, May 2015.

[33] J. Forgeat, *Data processing architectures Lambda and Kappa*, 2015, https://www.ericsson.com/research-blog/data-processing-architectures-lambda-and-kappa/.

[34] N. Narkhede, G. Shapira, and T. Palino, *Kafka: The Definitive Guide*, OReilly Media, Inc, 2016.

[35] A. Spark, "Spark streaming programming guide," *Apache Spark*, 2014.

[36] A. Bahga and V. Madisetti, Big Data Science & Analytics: A Hands-On Approach, VPT, 2016.

[37] A. Finamore, M. Mellia, M. Meo, M. M. Munafo, and D. Rossi, "Experiences of internet traffic monitoring with tstat," *IEEE Network*, vol. 25, no. 3, pp. 8–17, 2011.

[38] M. A. Monjas and A. Bascuñana, "ONTIC Deliverable D5.5.03: Adaptive QoE Control. Reference Point IF3-3: Policy Governance Function–Enforcement Point," Online Network Traffic Characterization, 2016.

[39] M. Mellia, A. Carpani, and R. Lo Cigno, "Tstat: TCP STatistic and analysis tool," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 2601, pp. 145–157, 2003.

[40] A. B. Prasetijo, E. D. Widianto, and E. T. Hidayatullah, "Performance comparisons of web server load balancing algorithms on HAProxy and Heartbeat," in *Proceedings of the 3rd International Conference on Information Technology, Computer, and Electrical Engineering, ICITACEE 2016*, pp. 393–396, Indonesia, October 2016.

[41] "Telecommunication Networks Group - Politecnico di Torino : TSTAT, 2014," http://tstat.tlc.polito.it.

[42] J. Vila-Carbó, J. Tur-Masanet, and E. Hernández-Orallo, "An evaluation of switched Ethernet and Linux Traffic Control for real-time transmission," in *Proceedings of the 13th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2008*, pp. 400–407, Germany, September 2008.

[43] P. Casas, B. Gardlo, M. Seufert, F. Wamser, and R. Schatz, "Taming QoE in cellular networks: From subjective lab studies to measurements in the field," in *Proceedings of the 11th International Conference on Network and Service Management, CNSM 2015*, pp. 237–245, Spain, November 2015.

[44] E. Danish, A. Fernando, M. Alreshoodi, and J. Woods, "A hybrid prediction model for video quality by QoS/QoE mapping in wireless streaming," in *Proceedings of the IEEE International Conference on Communication Workshop, ICCW 2015*, pp. 1723–1728, UK, June 2015.

[45] C. C. Aggarwal, *Data Mining: The Textbook*, Springer International Publishing, 2015.

[46] M. Bertin and I. Atanassova, "K-means and Hierarchical Clustering Method to Improve our Understanding of Citation Contexts," in *Proceedings of the the 2nd Joint Workshop on Bibliometric-enhanced Information Retrieval and Natural Language Processing for Digital Libraries (BIRNDL)*, pp. 107–112, CEUR-WS. org, Tokyo, Japan, 2017.

[47] ITU-T: Recommendation P.910. Subjective video quality assessment methods for multimedia applications, 2008, https://www.itu.int/rec/T-REC-P.910-200804-I.

[48] J. Dromard, P. Owezarski, M. López et al. et al., *ONTIC: D5. 3: ONTIC Subsystem Integration [Ph.D. thesis]*, LAAS-CNRS; SATEC; Ericsson Spain; Dell-EMC; UPM, 2017.

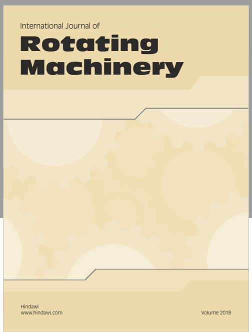
[49] A. R. Colson and R. M. Cooke, "Cross validation for the classical model of structured expert judgment," *Reliability Engineering & System Safety*, vol. 163, pp. 109–120, 2017.

[50] I. Martín de Diego, O. S. Siordia, A. Fernández-Isabel, C. Conde, and E. Cabello, "Subjective data arrangement using clustering techniques for training expert systems," *Expert Systems with Applications*, vol. 115, pp. 1–15, 2019.

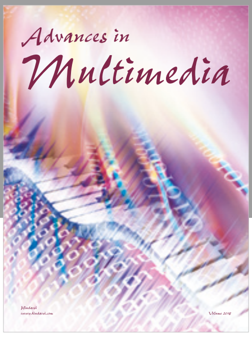
[51] A. Ahmad, A. Floris, and L. Atzori, "QoE-centric service delivery: A collaborative approach among OTTs and ISPs," *Computer Networks*, vol. 110, pp. 168–179, 2016.

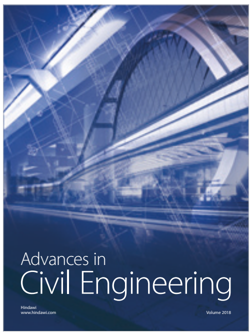
Journal of
Engineering

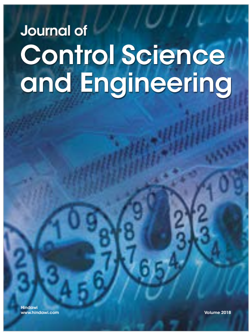
The Scientific World Journal


International Journal of
Rotating Machinery

Journal of
Sensors

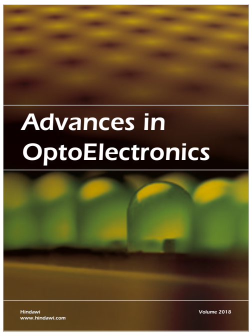

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science and Engineering

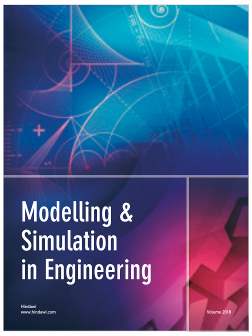
Journal of
Robotics

Journal of
Electrical and Computer Engineering

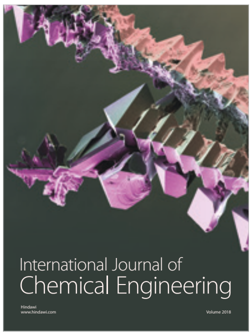

Advances in
OptoElectronics

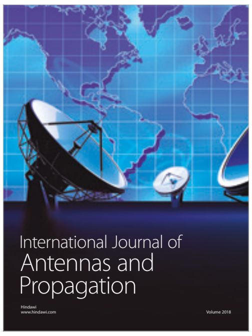
VLSI Design

International Journal of
Navigation and Observation

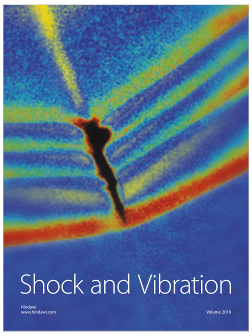

Modelling &
Simulation
in Engineering

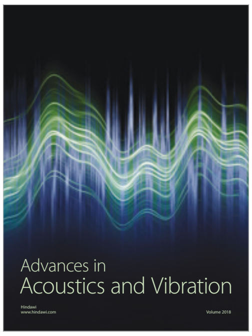
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration