



Context Monitoring for Improved System Performance and QoE

Florian Metzger¹(✉), Tobias Hoßfeld¹, Lea Skorin-Kapov², Yoram Haddad³, Eirini Liotou⁴, Peter Pocta⁵, Hugh Melvin⁶, Vasilios A. Siris⁷, Andrej Zgank⁸, and Michael Jarschel⁹

¹ Chair of Modeling of Adaptive Systems, University of Duisburg-Essen, Essen, Germany

{florian.metzger,tobias.hossfeld}@uni-due.de

² Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia

lea.skorin-kapov@fer.hr

³ Jerusalem College of Technology, Jerusalem, Israel

haddad@jct.ac.il

⁴ Communication Networks Laboratory, University of Athens, Athens, Greece

eliotou@di.uoa.gr

⁵ Department of Multimedia and Information-Communication Technology, Faculty of Electrical Engineering, University of Zilina, Zilina, Slovakia

pocta@fel.uniza.sk

⁶ Discipline of Information Technology, National University of Ireland, Galway, Ireland

hugh.melvin@nuigalway.ie

⁷ Department of Informatics, Athens University of Economics and Business, Athens, Greece

vsiris@aueb.gr

⁸ Faculty of Electrical Engineering and Computer Science, University of Maribor, Maribor, Slovenia

andrej.zgank@um.si

⁹ Nokia Bell Labs, Munich, Germany

michael.jarschel@nokia-bell-labs.com

Abstract. Whereas some application domains show a certain consensus on the role of system factors, human factors, and context factors, QoE management of multimedia systems and services is still faced with the challenge of identifying the key QoE influence factors. In this chapter, we focus on the potential of enhancing QoE management mechanisms by exploiting valuable context information.

To get a good grip on the basics we first discuss a general framework for context monitoring and define context information, including technical, usage, social, economic, temporal, and physical factors. We then iterate the opportunities and challenges in involving context in QoE monitoring solutions, as context may be, e.g., hard to ascertain or very situational.

The benefits of including context in QoE monitoring and management are demonstrated through use cases involving video flash crowds as well as online and cloud gaming.

Finally, we discuss potential technical realizations of context-aware QoE monitoring and management derived based on the SDN paradigm.

1 Introduction

With the ever increasing availability of media-rich, personalized, and *context-aware* services delivered over today’s networks, service and network providers are constantly battling to maintain a satisfied customer base. In addition, a paradigm shift is being witnessed in Internet service delivery, whereby we see a transition towards what has been referred to as an Internet of Services (IoS), envisioning everything on the Internet as a service [44]. Such a transition will potentially lead to new services being realized as large-scale service chains, combining and integrating the functionality of (possibly many) other services offered by third parties (e.g., infrastructure providers, software providers, platform providers). Key aspects and challenges to address will include the reliability and Quality of Service (QoS) delivery, which inherently relies on monitoring, quality estimation, and prediction mechanisms.

In light of the very competitive market, end-user Quality of Experience (QoE) will be one key differentiator between providers. In order to successfully manage QoE, it is necessary to identify and understand the many factors that can—and the specific factors that actually do in a given scenario—affect user QoE. Resulting QoE models dictate the parameters to be monitored and measured, with the ultimate goal being effective QoE optimization strategies [23]. The majority of QoE-based management approaches to date are primarily based on either network management (facilitated through monitoring and exerting control on access and core network level) or application management (e.g., adaptation of quality and performance on end-user and application host/cloud level) [52]. For example, many Web services (e.g., YouTube or Netflix), which are transparently run over various Internet Service Provider (ISP) networks, commonly implement QoE control schemes on the application layer by adapting the application to the conditions found in the network. The network and service management approaches of today’s networks are often designed to operate solely in the domain of a single stakeholder. Consequently, due to a lack of information exchange and cooperation among involved parties the effectiveness of such approaches is limited [6].

Going beyond QoE management, additional information may be exploited to optimize the services on a system level, e.g., by considering resource allocation and utilization of system resources, resilience of services, but also the user perceived quality. While QoE management chiefly targets the optimization of service delivery and currently running applications, the exploitation of *context information* by network operators could lead to more sophisticated traffic management, a reduction of the traffic load on the inter-domain links, and a reduction of the operating costs for the ISPs. *Context monitoring* in its broadest

sense aims to obtain as much information about the current system state as possible. For example, the popularity of video requests could be monitored, or the consequences of events (such as soccer matches) could be foreseen based on data collected from different sources, such as social networks (e.g., the popularity of videos could dictate caching/bandwidth demands). Such a holistic viewpoint offers the necessary input for providing enhanced service control and resource allocation functions, with clear potential for improving QoE. To date, there is still a limited understanding of the potential business models, technical realizations, and exploitation benefits of extending “traditional” QoE monitoring solutions (e.g., monitoring of network conditions, device capabilities, application specific parameters, and user-related factors) with context monitoring.

The goal of this chapter is to provide further insight into the potential of exploiting context monitoring for QoE management, both in terms of increasing QoS and QoE (by managing individual flows and users), and also in terms of improving the resilience of services (by making available information about network state to these services). We also acknowledge the fact that today’s Internet traffic mix is extremely diverse, with traffic from numerous services fused together. Context monitoring is discussed from the perspective of a single service (or class of services), as those fundamentals are important to us. The interactions of concurrent QoE management efforts in a single network are extremely interesting as well but not in the scope of this paper.

This chapter is organized as follows. Section 2 lays out a generic framework for context monitoring by proposing a classification scheme for context information and discussing the potential involvement of this information in QoE monitoring and management solutions. For demonstration purposes, example scenarios are depicted in Sect. 3 illustrating the benefits of exploiting context information in actual use cases that can ultimately lead to overall QoE improvements. Section 4 then discusses a technological solution path for enhancing QoE with Software Defined Networking (SDN) that utilizes context monitoring data. Finally, concluding remarks and future work are given in Sect. 5.

2 Generic Framework for Context Monitoring

A generic framework for context monitoring provides the means to utilize context information in order to improve a networked system. To this end, the term *context information* is defined in Sect. 2.1. A classification scheme of context information is proposed which provides the means for systematically identifying useful context information for various use cases. Section 2.2 describes how to model QoS and QoE as input for any improvement strategy. Different approaches for involving context in QoE monitoring are introduced in Sect. 2.3.

2.1 Definition and Classification of Context Information

The term *context* is very broad and several definitions exist in literature. Those definitions vary depending on the actual system under consideration. Goals that

involve context could, for example, be end user QoE improvements, QoS improvements, or cost reductions. Consequently, on one hand, context is defined to be one of three QoE influence factors in [50]. In a similar manner, the authors of [48] define context as “*anything that can be used to specify or clarify the meaning of an event*”. This kind of context information may be deployed directly during measurements, modeling, or an improvement process of QoE, as done for mobile TV [8].

On the other hand, context also relates to information that allows to determine the current state of a system. The authors of [1] define context as “*any information that assists in determining a situation(s) related to a user, network or device*”. Such information may be indirectly utilized in order to improve QoE. Additionally, it allows for direct measurements and improvements of the QoS. Since there is a strong relationship between QoS and QoE, e.g., as [18] notes, transitioning between different types of context factors is straightforward and we do not need to distinguish between them. The utilization of context information is of course strongly dependent on the actual use case.

A context space model is proposed by [42] where *situations* are determined from context attribute values, e.g., sensor data. First, these context attribute values are used to infer context states defined as “*the current state of a user, application, network or device being modeled at a time instant based on the context attributes*”. The context states are combined to determine an overall situation. In our view, the proposed context space model is not required to integrate context into QoE monitoring and management. The context information directly influences QoE, therefore the intermediate step to map the context attribute value to a context space is not always necessary.

This notion of context factors as QoE influence factors is also in line with recent works described in [50,56]. The work in [56] considers four multi-dimensional spaces: Application, Resource, Context, and User space, together dubbed “ARCU model”. Context is thereby composed of dimensions that indicate the “*situation in which a service or application is being used*”. In a similar manner, [33,50] describe context influence factors as “*factors that embrace any situational property to describe the user’s environment in terms of physical, temporal, social, economic, task, and technical characteristics. These factors can occur on different levels of magnitude, dynamism, and patterns of occurrence, either separately or as typical combinations of all three levels*”.

Based on the classes of context factors that influence QoE, as defined in [50], we generalize the classification by additionally integrating context factors from a system’s point of view as well. Thus, context information may be used to determine the user or the system situation and may be directly or indirectly mapped to QoE and QoS. Figure 1 visualizes the taxonomy of context factors and includes several examples as instantiations of those classes. Here, we divide context factors into five broad categories, with further refined—albeit non-exhaustive—sub-classes of context. Section 3 will depict different use cases that consider context factors from multiple classes. Further examples can also be found in [33,48,50,56].

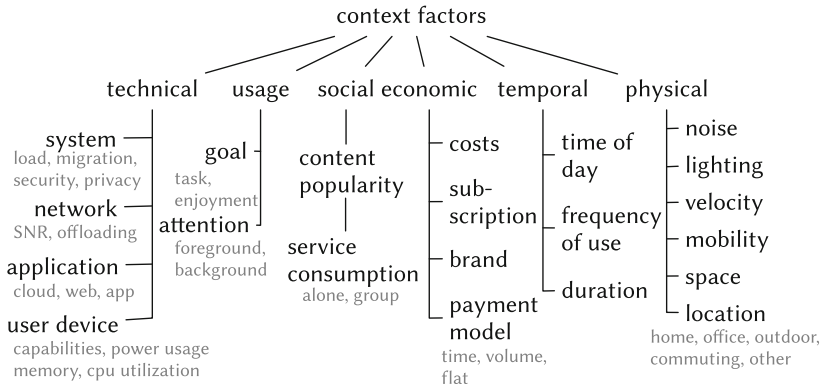


Fig. 1. Classification of context factors involved in QoS and QoE monitoring. Examples are denoted in light grey.

2.2 Towards Context-Aware QoE Modeling and Estimation Strategies

Following the overview of different categories of context factors influencing QoE, we discuss QoE modeling concepts and the potential of enhancing the quality estimation process with context awareness.

The main aim of QoE estimation models is to mimic the quality assessment process performed by the end user of a corresponding service as accurately as possible. The main output of the model is thus an estimation of the quality as perceived by the end user in the specific scenarios of interest. The aim is to achieve a high correlation between estimated quality scores and subjective scores in these scenarios. Quantifiable influence factors are mapped to quality scores commonly expressed by Mean Opinion Score (MOS) values. To this end, the data obtained from subjective quality tests is required in order to find mapping functions that best resemble the human perception of quality. Different types of quality models are currently in use for service quality estimation, quality monitoring, or even service design. We refer the prospective reader to [53], which provides a comprehensive review of different types of quality models. Reduced reference and no-reference models are generally deployed for real-time service monitoring, as they do not require access to a complete reference signal in the assessment process. In principle, both reduced reference models as well as no-reference models can be applied at different points in the network in order to better quantify the impact of packet loss and other important network parameters on QoE.

As QoE is a multidimensional concept influenced by a number of system, user, and context factors, it is important to keep in mind that QoE is highly dependent on QoS. According to [23], QoS parameters represent one of the most business-relevant parameters for network and service providers. The mathematical dependency of QoE on QoS parameters at both the network and application level can usually be characterized by logarithmic or exponential functions as they

best mimic the corresponding user quality perception. Logarithmic relationships were studied in [47, 49] and described in terms of the Weber-Fechner law [60]. In principle, this law traces the perceptive abilities of the human sensory system back to the perception of so-called “*just noticeable differences*” between two levels of a certain stimulus. For most human senses, the just noticeable difference can be represented by a constant fraction of the original stimulus size. Exponential relationships can be described by the IQX hypothesis [18, 24], which describes QoE as an appropriately parametrized negative exponential function of a single impairment factor (QoS parameter). A main assumption behind the IQX hypothesis is that a change in QoE depends on the actual level of QoE. Therefore, the corresponding relationship can be described by a differential equation with an exponential solution. Both types of relationships confirm the general observation that end users are rather sensitive to quality impairments as long as the actual quality level is good, whereas changes in network conditions have less impact on their quality perception when the quality level is low. However, they differ in terms of their basic premise. The Weber-Fechner law links the magnitude of the QoE change to an actual QoS level, whereas the IQX hypothesis assumes that the magnitude of change depends on the actual QoE level. Furthermore, the law is mostly valid when a QoS parameter of interest relates to a signal or application level stimulus directly perceivable by the end user (e.g., delay or audio distortion), while the IQX hypothesis was derived for QoS impairments on a network level, which are not directly perceivable by the end user (e.g. packet loss) [53].

At a general level, the process of estimating QoE involves relating network- and system-level Key Performance Indicators (KPIs) (e.g., delay, loss, throughput, or CPU consumption) to end-to-end application level Key Quality Indicators (KQIs) (e.g., service availability, media quality, reliability), cf. also Fig. 2. QoE estimation models may then be derived as a weighted combination of KQIs as

$$QoE = f(w_1, KQI_1, \dots, w_i, KQI_i).$$

It should be noted that the weights w_1 to w_i differ as the KQIs have different levels of impact on QoE. In other words, the weight represents a strength of the impact of the particular KQI on QoE.

As an example, consider a video streaming service whereby transmission parameters such as loss or delay result in video artifacts impacting the media quality, which may in turn be translated to QoE. In certain cases a QoE model may directly incorporate KPIs, e.g., QoE estimation modeled in terms of network delay or packet loss. Going beyond a “basic” view of the QoE estimation process, additional input to a QoE estimation model may be provided by user or context influence factors [49]. We refer to an “enhanced” context-aware QoE estimation model as potentially incorporating context data on three different levels:

- (a) Context data on system state that directly impacts KPIs and thus also indirectly QoE, e.g., network load, traffic patterns, offloading support, end user device processing capabilities.

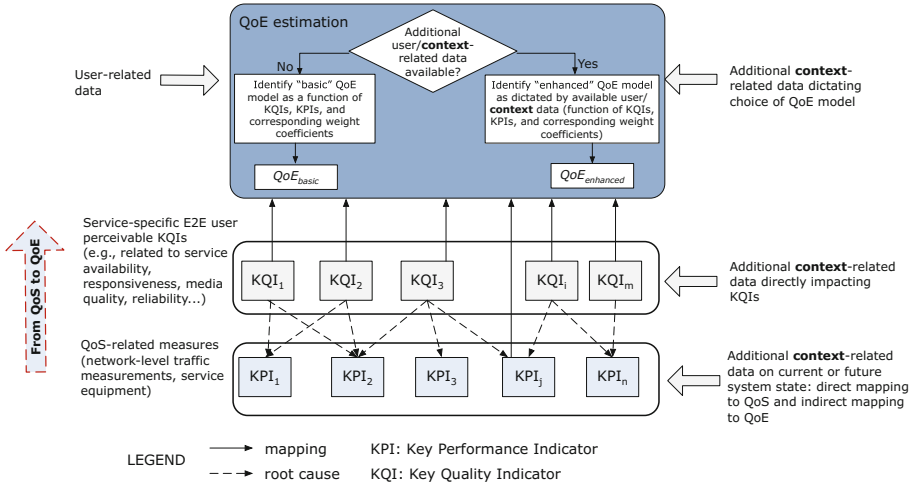


Fig. 2. Context-aware QoE estimation process, adopted with modifications from [4]. User perceivable KQIs are derived from KPIs and show a root cause relationship, while KQIs and KPIs are further mapped to QoE using QoE estimation models. Additional context data (typeset in bold) may be integrated into the process on three different levels and lead to more reliable and accurate QoE estimations.

- (b) Context data that directly impacts KQIs and in turn QoE, e.g., public events or flash crowds in a specific geographic region.
- (c) Context data that impacts the choice of QoE model, KQIs, and weight factors. Based on context data certain KQIs may or may not be included in the QoE model.

In practice, QoE monitoring applications usually run on top of general network monitoring systems that provide input QoS parameters together with a certain amount of context information. As the monitoring system is supposed to work in real-time, computational efficiency and data reliability represent the most important performance indicators. Moreover, as most of the current QoE monitoring systems also require application-specific parameters as their input to allow a reliable estimation of QoE, additional monitoring techniques, such as Deep Packet Inspection (DPI), are commonly implemented, but they can also negatively affect the system's performance. The availability of context information not only provides the opportunity to use enhanced QoE models as discussed previously, but also provides more reliable data and improves QoE estimation accuracy. In other words, complex context information provided to a monitoring system in well-defined frequent intervals can fine-tune the reliability and accuracy of QoE estimation. As QoE estimation models and context monitoring tools are usually part of the same monitoring system, they are aware of each other. If this would not be the case, system complexity would greatly increase, with only marginal QoE estimation improvements in terms of the reliability as well as accuracy to be had. Consequently, given that QoE monitoring represents a

Table 1. QoE monitoring insights for each context factor class. The given examples are non-exhaustive.

Physical	<i>Information source:</i>	User, device, network
	<i>Techniques and tools:</i>	Sensors, DPI, user feedback, GPS information, prediction techniques
	<i>Examples of opportunities:</i>	Pre-allocate network resources based on the user’s daily life pattern, location or device capabilities
Temporal	<i>Information source:</i>	Network
	<i>Techniques and tools:</i>	Operator databases, DPI
	<i>Examples of opportunities:</i>	Reconfigure the routing process based on historical information about the traffic load per hour of day
Economic	<i>Information source:</i>	ISP, operator
	<i>Techniques and tools:</i>	Data repositories (subscriber databases)
	<i>Examples of opportunities:</i>	Differentiate gold subscribers over best-effort ones
Social	<i>Information source:</i>	ISP, operator, CDN, OTT, media, social networks, user
	<i>Techniques and tools:</i>	Data analytics, prediction techniques
	<i>Examples of opportunities:</i>	Video flash crowd
Usage	<i>Information source:</i>	User, device
	<i>Techniques and tools:</i>	User feedback, application monitoring, DPI
	<i>Examples of opportunities:</i>	Prioritize foreground versus background applications
Technical	<i>Information source:</i>	Device, network
	<i>Techniques and tools:</i>	DPI, sensors, actuators, probes, embedded agents, network elements
	<i>Examples of opportunities:</i>	Control the cell selection based on the current device battery level

crucial part of QoE management systems, this leads to more efficient and reliable adaptation strategies that specify how to change parameters at different layers in order to improve QoE.

We direct the interested reader to the chapter on QoE management approaches—dealing specifically with QoE management in Future networks—and to [57,63].

2.3 Involving Context in QoE Monitoring

Specifying the QoE modeling and estimation process is an essential prerequisite to QoE monitoring. We have recently been witnessing a paradigm shift

from “traditional” QoS monitoring to QoE monitoring procedures [35]. Various approaches can be used for QoE monitoring, namely on end-user or device level, network level, or a combination of the two, see, e.g., [53]. QoE monitoring may run on the network layer to capture QoS parameters, requiring the existence of mapping functions between QoS and QoE, but also on application layer in order to obtain relevant information directly from the application in question.

Including additional context data, potentially gathered from a broad range of sources, can significantly enrich QoE monitoring by providing better information about a system’s current and future properties and state. For example, monitoring social networks’ text streams with natural language processing algorithms and machine learning approaches [28] can provide a valuable source of information regarding short-term trends in popular content or potential “high-profile” future events which allows for better caching strategies to avoid stalling in video streaming [55].

While QoE-based management is already a novel procedure when compared to traditional QoS-based management, its performance can still be enhanced further with context awareness. Therefore, in order to reap all the benefits that stem from the possibility of context-awareness in a network, viable and feasible context monitoring mechanisms need to be devised. It is expected that these mechanisms will differ per use case, to better reflect the requirements and idiosyncrasies of each scenario (as it will be also explained in Sect. 3). However, often a trade-off has to be made between the amount of information collected and the processing time required to use them properly.

In Table 1, we provide some insights about QoE monitoring in a high-level way. Specifically, following the context factor classification proposed in Sect. 2.1, we present the potential sources of information per class, the techniques or tools to extract information from the appropriate sources, and finally some new opportunities, that will arise if context monitoring, and subsequently, context- and QoE-aware management are realized. Possible sources of context information are:

- (a) The end-users themselves,
- (b) The users’ device characteristics, ranging from the device’s hardware/software up to the application layer,
- (c) The network, i.e., any intermediate network nodes in the core or the access network that are capable of providing context-related information,
- (d) The operator’s or ISP’s proprietary infrastructure,
- (e) Third party information, namely feedback from any players who do not control the information flow, but only its content and format (e.g., content and “Over-The-Top” (OTT) service providers, social channels such as Facebook, Twitter, etc., provisioning their services over the operator’s or ISP’s infrastructure).

Regarding the acquisition of technical context factors, some insights may be found in [15].

The main concerns and challenges in implementing QoE monitoring procedures on top of the already available network mechanisms mainly have to do

with the complexity and signaling overhead that would be inevitably imposed in the network. Context awareness does not come without a cost. Thus, it is important to employ it only when and where it is meaningful. A crucial point is to avoid turning the context monitoring procedure into another negative QoE influence factor, e.g., by introducing a higher congestion to the network, or draining the user device's battery faster, or even persistently requesting user feedback. Therefore, the extent of the influence of each context factor on QoE needs to be established. A further challenge in capturing the context and consequently in validating QoE monitoring techniques is the fact that these studies have to be conducted in the field. The controlled laboratory environments, which are conventionally used for QoE monitoring and modeling purposes, do not allow for capturing the diversity of the various context factors and the plethora of use cases. Some best practices for acquiring context information are provided in [51], although the focus there is mainly on user experience studies and not QoE.

It should be emphasized that context monitoring requires models, metrics, and approaches that capture the condition of the system on a network and service layer, application and service demands, and the capabilities of the end-user device. Assuming, however, that context monitoring can be realized, then the exploitation of context information may lead to more sophisticated system and traffic management, traffic load reduction on the inter-domain links, and reduction of operating costs for ISPs. However, before this can occur, effective methods need to be determined in order to incorporate the monitored context parameters into an enhanced cross-layer QoE management. A promising option might be based on SDN [31] as discussed further in Sect. 4.

3 Context Factor Examples, Use Cases and Literature

To demonstrate the potential benefits of context information in QoE monitoring and management, this section presents some practical examples, with varying involved parties and services. Table 2 also highlights the relevant factors for some scenarios. As outlined in Sect. 2.1, context is a very broad umbrella term that encompasses a wide range of elements, many of which are interrelated. To get a grasp of their influences, a general overview and related research that illustrates the idea of context factors is presented here for individual factors first, before moving on to more specific use cases.

3.1 Usage Examples for the Context Factor Categories

Usage Context Factors. The interactivity required for a certain task to be conducted in a satisfactory fashion is another usage context factor that can be essential in certain scenarios. Besides gaming this especially concerns conversational applications. In [7], the extent to which interactivity requirements of Real-time Communications (RTCs) and specifically voice applications impact QoE is examined. Of particular note is the ITU-T E-Model [29], a telecommunications planning tool that generates a QoE score. Thus they developed a

Table 2. Overview of context factors and their relevancy to the usage scenarios discussed in Sect. 3.

Physical	<i>Video Flash Crowd:</i>	Devices and sensors detecting crowds
	<i>Online and Cloud Gaming:</i>	Type of Device and Input devices
	<i>Context-Aware Billing:</i>	Wireless Channel state (SINR, ...)
Temporal	<i>Video Flash Crowd:</i>	Diurnal patterns, prior knowledge of events
	<i>Online and Cloud Gaming:</i>	Amount of time played in session, total; load at time of day, week
	<i>Context-Aware Billing:</i>	Period: peak or off peak hour
Spatial	<i>Video Flash Crowd:</i>	Prior knowledge of crowd events
	<i>Online and Cloud Gaming:</i>	Stationary, on the move
	<i>Context-Aware Billing:</i>	Ambient free Internet access
Economic	<i>Video Flash Crowd:</i>	Announced/advertised events/content and expected popularity
	<i>Online and Cloud Gaming:</i>	Data-center location and capacity, game charging model
	<i>Context-Aware Billing:</i>	Used up quota of the user
Social	<i>Video Flash Crowd:</i>	Popularity of content, social relationship of video users
	<i>Online and Cloud Gaming:</i>	Playing solo, cooperatively, or competitively
	<i>Context-Aware Billing:</i>	Social events for device-to-device communication
Usage	<i>Video Flash Crowd:</i>	Estimating popularity and crowds through individual usage history
	<i>Online and Cloud Gaming:</i>	Player's skill level and previous experiences with game and genre
	<i>Context-Aware Billing:</i>	Charging based on usage patterns
Technical	<i>Video Flash Crowd:</i>	Specific service implementation and ability to handle crowds
	<i>Online and Cloud Gaming:</i>	Game features, graphic fidelity, specific game mechanics, genre
	<i>Context-Aware Billing:</i>	Charging based on device properties

delay impairment function that depends on the nature of the conversation. For example, a conversation with strong interactivity requirements is modeled by a curve that returns a low impairment up to the 150 ms level, then increases rapidly between 150 ms to 300 ms and levels out thereafter as the conversation has essentially become half duplex anyway. The authors in [38] also examine this issue, introducing the concept of “*Tasks*”. These tasks each describe a specific set of requirements, e.g., in terms of the conversation’s interactivity. Examining individual users’ behavioral patterns and from that extrapolating the actions

of a larger crowd in order to enhance, e.g., network-wide video streaming QoE management is also an enticing aspect of this type of context factors.

Noise and Vibration Context Factors. Research by Wilk et al. [62] presents a prototype that measures smartphone background noise and screen vibration in real-time for users on the move. These factors are then used to decrease downloaded video and audio quality in order to reduce bandwidth. This strategy is based on the premise that delivering high quality audio and video under such conditions represents a waste of bandwidth and perhaps money, illustrating the potential inter-relatedness of context factors and economics.

Charging and Billing Context Factors. With usage-based pricing in mobile networks being a reality in many countries, users are incentivised to sparingly use their allotted data volume. Pursuing this line of thought, context monitoring could open up several opportunities for interactions between the operator, the network, and the user in this regard. Data-cap-aware video adaptation is examined in [10]. The work proposes to choose the best video resolution that still enables the client to stay below its data cap. Another approach presented in [21] evaluates the case of shifting elastic traffic to off-peak periods. Alternatively, one can postpone delay-tolerant traffic until it can be offloaded to free-of-charge Wi-Fi networks. With the upcoming 5G mobile networks it could also be feasible to offload certain traffic using direct device-to-device communication with someone willing to transfer the data for free assuming sufficient resource availability [9]. In this case, context information can be utilized to predict when such a suitable connection will be available. Finally, a channel-aware pricing model could also be an opportunity, such as is discussed in [19]. Herein, the operator sets the prices to a value that fits the currently available radio and network resources.

Mobility Context Factors. With the evolving mobile architectures, users increasingly expect service and application availability whilst on the move. Returning to the aforementioned ITU-T E-model [29] the advantage of application accessibility on the move is modeled by an advantage factor defined as “*the compensation of impairment factors when the user benefits from other types of access*”. However, this is specifically designed for voice communication and does not necessarily apply to the same extent to other applications such as online video games or non-RTC applications, thus further research is required. A more recent example is provided in [40]. Here, the scenario of adaptive video streaming on the move is highlighted, especially the challenges surrounding the prediction of outage events through context and appropriately modifying the buffering behavior to avoid playback stalls.

Location Context Factors. Usually location relates specifically to user location. However, for applications that are hosted remotely in the cloud could be extended to include where the application is hosted, keeping in mind the relative

distance between host and client and the impact on latency. With the growing importance of cloud services this raises an interesting interdependence of spatial, physical, and economic context factors and can have a significant impact on the QoE of RTC-applications such as communication and gaming.

A 2011 study [34] suggests a Data Center (DC) energy consumption ratio of 1.5% of the global production. This share may continue to rise as public and private cloud adoption is growing steadily. About 70% of the power usage is directly proportional to the server load, making efficient usage of the existing servers a must. The principal underlying technology, which facilitates management of workload in a DC, is virtualization. Rather than each server hosting a single application instance, a number of Virtual Machines (VMs) are running concurrently on a single physical server. These VMs may be migrated to a different local host or across the Wide Area Network (WAN) depending on a variety of strategies with significant impact on QoE.

A good example is the *follow-the-sun*-strategy that helps minimizing the network latency during office hours by placing VMs close to where they are requested most often. Such a strategy can improve QoE, with the trade-off of increased energy costs as they are typically higher during daylight time. Where latency is not a primary concern or where other factors are given precedence, there are a number of different strategies which can be applied in addition. These generally involve VMs getting shifted to locations with cheaper resource costs, for example to places where less expensive cooling is required, exploiting lower power costs during night times (“follow-the-moon”), or following fluctuating electricity prices on the open market [46].

A final reason for monitoring spatial context factors in DC environments is that of data safety. Operations related to fault tolerance, mirroring, maintenance, or disaster recovery can be the cause of VM migrations. Regardless of the motivation, migrating VMs can greatly impact application response times and thus QoE. If detailed Service Level Agreements (SLAs) are negotiated and in place, QoS parameters can be contractually covered. However, users without an SLA may then suddenly find greatly increased delays during or after a migration.

3.2 Use Case: Video Flash Crowd and QoE

The potential of context monitoring for video streaming was assessed, e.g., in simulation studies in [22,25] on which this section is based. In the papers’ scenario, a flash crowd of users watching the same video is examined. This sudden increase of popularity is a typical phenomenon of video platforms with user-generated content like YouTube. Video cascades can often emerge due to new or popular content being spread through social media challenges. Those phenomena, dubbed flash crowds, may be temporarily limited because of event-related content, spatially limited because of regional interests, or socially limited due to grouping effects in social interests [12]. The simulation provides a model for the effects of flash crowds on adaptive streaming and compares different approaches to Content Distribution Network (CDN) load balancing and video

adaptation strategies to each other to study the benefits of context information in the approaches that support it.

The context-unaware strategy has no knowledge of flash crowds and reacts too slow to such events. Hence, some users experience stalling even though enough capacity would be available to serve all users and stalling should be entirely avoidable. HTTP adaptive streaming (HAS) is necessary in order to adapt to the current network situation and to reduce the number of stalling events. If many of the users unaware of the flash crowd event request the video at its highest quality, most of the users will suffer from stalling, leading to a worse overall QoE. This topic is also covered under the term *QoE fairness* [26].

The utilization of context information by the CDN as well as by HAS improves QoE. It is difficult to properly configure the context-unaware load balancing strategies as the exact values strongly depend on the actual flash crowd. Therefore, a proper information exchange mechanism is required to make the information available across layers. The earlier the flash crowd scenario is recognized by the load balancer the better the overall system performance will be.

The results are very sensitive to the dynamics and interactions of the HAS control loop and the CDN load balancing. Thus, in practice, realistic tests and input models are required to quantify the results and to derive reasonable configurations. This serves to demonstrate the potential of exploiting relevant context data. In the given use case, the contextual information regarding the formation of a flash crowd is collected by a third party. Other studies have also addressed related improvements, such as the approach proposed in [37], which suggests a video control plane that can dynamically adapt both the CDN allocation and video bitrate midstream based on global context knowledge of network state, distribution of active clients, and CDN performance variability.

Other video scenarios addressed in related work draw similar conclusions. For example, significant work has addressed the challenges arising from multiple concurrent clients accessing HAS video in a given access network, thereby competing for bandwidth across a shared bottleneck link. Problems arise due to individual clients making adaptation decisions based on local observations, hence clients' adaptation behaviors interact with each other, which results in quality oscillations. Solutions proposed in literature involve centralized network-based solutions deployed using SDN [20], enhanced client-side adaptation to improve fairness among flows [32], and server-based traffic shaping [2]. However, in all these cases context information (in this case network state data including overall resource availability and global resource and traffic demands) could likely be utilized to control quality adaptation decisions (on a domain-wide level), consequently reducing oscillations and improving QoE.

The flash crowd scenario can illustrate the benefits of employing SDN as a centralized technological solution. With traditional IP networks, decisions are made based on local knowledge, so even if a server that conducts data mining on social networks could foresee a high download rate, it will be challenging to perform load balancing. This task would require changing the routing policy at the Border Gateway Protocol (BGP) level but may also have an impact on the

Interior Gateway Protocol (IGP) which is confined to its autonomous system. Even if the propagation of the routing updates would be implemented by a gossip algorithm, the time it takes will make the network topology change irrelevant. However, with SDN, this very complex and time consuming task is simplified. The forwarding decision is taken higher up by the central controller, so an update to the routing decision at the controller implements the load balancing. From a technical perspective, further insight is needed in regards to the information exchange between SDN controllers that are responsible for different domains, and between SDN controllers and other entities that can provide context information.

3.3 Use Case: Online and Cloud Gaming

Context monitoring can play a huge role for video games, specifically for online and cloud games. Compared to videos and watching video streams, video games add a high degree of user interactivity to the mix, which limits and alters the type of eligible evaluation and management approaches.

In the case of cloud gaming, the game server executes the game logic, renders the scene, and sends an encoded video stream to the client in real-time [27, 41]. The client is responsible for decoding the video and capturing the player's commands. Two network-level factors are critical: the bandwidth required by the video stream (influencing image quality and frame rate) [58], and the Round-Trip Time (RTT) (influencing the game's responsiveness to input commands) [13, 14]. For online games, the throughput is much less important when compared to the RTT. Nonetheless, the network path is influenced by numerous kinds of context factors, most prominently technical (type of access), spatial (user on the move), and economic (data center location and amount of available processing and GPU resources).

Speaking of the economic factor, service providers are inclined to either use a centralized (to achieve a maximum multiplexing gain especially of the costly GPU resources) or a follow-the-moon strategy on their active server locations to save energy and processing costs. However choosing a too-distant data-center location would introduce additional latency to the system, negatively impacting the player's experience. The amount of concurrent players can be deduced by a diurnal temporal context factor. Proper trade-offs between these three context factors (namely economic, spatial, and temporal) have to be determined.

Besides network aspects, the type of device used for the gaming client and its input methods play a significant role in evaluating cloud gaming. The device's native resolution determines the game's optimal rendering quality setting and the capture resolution as well as the required bandwidth to transmit the video stream. Likewise, but maybe not as obvious, the available input methods determine the effect size latency has on the game. This especially concerns touch controls, which typically lack in accuracy as well as immediacy that can be especially felt in fast-paced games such as first-person-shooters.

This argument can additionally be extended to consider the type and genre of a game as a further context factor. The range of interactivity in games is very large and diverse, sometimes even in a single game. While some games require

decision-making and inputs on a millisecond-scale (e.g., first person shooters like Counter-Strike: Global Offensive or DOOM), some games might require hundreds of input decisions every minute (StarCraft II), while others require only a single input every few seconds or even minutes with very wide timing windows (take turn-based strategy games or adventure games with logic puzzles such as the classic Monkey Island series). This technical context factor represents itself well in the way QoE management can be conducted in online and Cloud games. If the specific characteristics of a game are known in advance, the timing constraints can be determined without further measurements and put to use for management of the transmission. Also, if a game with an online component is played through a cloud gaming service, the game's lag concealment mechanisms (that are in place for almost any kind of online multiplayer game) could additionally encompass the cloud gaming service and provide telemetry on the streaming latency to the online game's server in an effort to capture the actual end-to-end latency and improve the gaming experience.

A further player-level temporal and social context factor might also be interesting for management purposes: the duration players have been playing the same game in one session as well as the total time. Both metrics might either heighten the player's sensitivity to deviations from the expected playing experience or might make her more tired and thus ignorant of quality degradations. Depending on the direction of the interaction, QoE management can either be lenient or become more stringent during the course of a game session. Previous studies have shown the link between network performance and game play duration [11]. Further studies have shown that player experience and skill are important QoE influence factors which may clearly impact a player's tolerance to performance degradations [43, 59]. Hence, experienced players could be treated differently to novice or unexperienced players by the responsible mechanism.

4 Discussion on Technical Realization Approaches of Context Monitoring

Thus far we have discussed a generic framework for context monitoring and provided use cases which outline the opportunities for exploiting concrete context data in QoE monitoring, estimation, and management solutions. In this section we discuss novel technical realization approaches and challenges.

In recent years, concepts like SDN and Network Functions Virtualization (NFV) have become key drivers of network innovation. With their emergence, the importance of software in networking has grown rapidly. This trend is lead by open-source initiatives like the OpenDaylight¹ project, the Open Network Operating System (ONOS)² project, or the Open Platform for NFV (OPNFV)³ project. The introduction of these technologies paves the way for new possibilities to control and centrally orchestrate the network in a more flexible fashion in

¹ <https://www.opendaylight.org>.

² <http://onosproject.org>.

³ <https://www.opnfv.org>.

order to achieve and maintain a high QoE standard for the users. SDN involves decoupling the data plane from the control plane [39] where the decision on how to forward a packet is taken.

One of the goals of SDN is the creation of a network OS which abstracts the network complexity and allows operators to specify a network policy based on data analytics and other sources without having to take care of the implementation details. Similarly, SDN can enable applications and services to express the required QoS expected from the network and let applications on top of the controller translate this algorithmically into optimized forwarding decision in order to satisfy QoE demands. SDN increases flexibility and efficiency by allowing the information to be aggregated at a single logically-centralized location (global SDN controller) to be subsequently accessed by multiple applications. SDN can enable a global or domain-wide view of a network and its usage, which includes network-level QoS and topology, user and application QoE, and context. Information from multiple sources can enhance the correlation and prediction of traffic demand. SDN allows the creation of dedicated monitoring solutions. The programmability of these kinds of flow-based monitoring functions supports a fine-grained adaptivity (e.g., per user, per application) of monitoring and enforcement procedures. These features can enable real-time application-aware network resource management. Combined with new distributed cloud approaches, like fog or mobile edge computing, this can bring services much closer to the user, thereby reducing latency and improving quality. Information gained from context-based QoE monitoring can thus not only be used to steer traffic, but also to dynamically instantiate edge cloud capabilities.

It should be noted, that SDN is, by far, not the only solution to context monitoring. But it might be the most prolific one in the future when speaking about operator-wide orchestrated monitoring. Other solutions might include more network-independent, end-to-end, application-layer, as well as cross-layer-focused approaches, e.g. in the fashion of [3]. While these are out of scope for this work, they merit their own separate investigations in the future. Moreover, this section should be understood of a high-level discussion of the benefits and obstacles of the SDN-based approach and not as an instruction manual on the actual implementation of context monitoring.

4.1 Monitoring with SDN

One of the benefits of SDN is that it facilitates data collection through standardized vendor-independent interfaces and makes the monitoring task more scalable by not having to place isolated monitors in multiple locations. Rather, QoS, QoE, and context information can be collected at a logically centralized location (for example the SDN controller). The interface between the controller and the network forwarding devices is generally referred to as Southbound Interface (SBI), enabling the controller to know what is going on in the network. Several SBI protocols have already been proposed (e.g., “ForCES” by IETF) although the most famous is OpenFlow (OF). While in principle the OpenFlow protocol enables monitoring, its main purpose is the control of traffic flows. Therefore, usually

dedicated monitoring protocols such as “sFlow” are used to gather information for the SDN controller.

At this stage a couple of issues already arise. These are mostly present in classic monitoring solutions as well, as long as they act as an intermediary and not on an end-to-end basis. First, application QoE measurements and context information require communication and information exchange between SDN controllers and end-user devices or content providers. Monitoring protocols are however typically designed to communicate with network elements, but for the envisaged scenario, metrics are also needed from user devices, such as terminal capabilities (e.g., CPU) or buffer state. Generally, the concept where a global SDN controller has relevant information from the full end-to-end infrastructure (core/access/end device) is tied to emerging research on time-aware applications and systems [61]. This initiative is investigating the opportunities and challenges that are presented by having precise time synchronization demands on all interconnected devices. One key benefit of this approach is in the area of QoS/QoE whereby the SDN controller will have access to precisely timestamped network information. This is especially important for delay-sensitive applications, such as multiplayer online games or teleconferencing.

Context information includes, amongst others, the available Wi-Fi networks, user and subscription information, and provider policies. For this reason, interworking and communication of SDN with entities such as 3GPP’s Access Network Discovery and Selection Function (ANDSF) and Policy and Charging Rules Function (PCRF) can be important. The PCRF and the accompanying Policy and Charging Enforcement Function (PCEF), which are traffic management and DPI nodes located in the core network, have access to all the policy and charging information related to subscribers. Having this knowledge, they control the provisioning of the QoS for each flow through the concept of bearers (a network tunneling concept). These two entities are therefore also a good candidate for controlling and enforcing QoE-based policies. A future, enhanced PCRF should then be able to enrich the bearer selection and management process based on the specific QoE requirements of each service data flow.

Accurate SDN-based QoE monitoring requires measurements from multiple domains, which in turn necessitates communication between SDN controllers (between their “westbound” and “eastbound” interfaces). In [54] different hierarchies and topologies for the deployment of controllers are proposed. The ONOS project [5] is a good example of efforts to tackle the challenge of providing a centralized coordination while avoiding performance degradation at the control plane. ONOS maintains a global view of the network while the SBI is physically distributed among multiple servers.

4.2 QoE Management with SDN

Overview of Existing Approaches. Initial commercial applications that combine QoE management with SDN already exist (e.g. for mobile nets⁴). This is

⁴ https://www.nokia.com/en_int/news/releases/2014/09/02/nokia-networks-big-data-innovation-promises-dynamic-experience-management-networksperform.

reflected in literature as well. In [36] the SDN architecture is leveraged in order to introduce a “QoE-service” specifically for OTT providers. This QoE-service takes advantage of network QoS metrics, application QoE metrics or user feedback, and by combining them with the global resource view of the network it enables elastic traffic steering with the objective to enhance the performance of OTT applications or for premium users.

The work in [45] considers a framework where the QoE of streaming video is measured at the video player running at the destination node, informing the network of the achieved QoE. If the video QoE is low, the SDN-based network identifies bottlenecks on the delivery path and assigns a new video server to deliver the requested video to the end-node. A similar approach is taken in [30] on the example of YouTube video streaming. An agent at the end devices measures the amount of buffered video data of the video and notifies an application controller if it drops below a certain threshold. The application controller instructs an SDN controller to temporarily relocate the YouTube flows of this user to a less congested link. When the video recovers, the relocation is reversed to conserve resources.

The work in [17] proposes an in-network QoE measurement framework. Unlike the previous approach, the QoE is measured inside the network and based on the video fidelity and representation switching. This is achieved by SDN replicating the video stream to a QoE measurement agent. The QoE measurements are collected by a measurement controller, which can modify the forwarding paths using SDN in order to improve the delivered QoE. Moreover, the measurement controller exposes an API to allow other applications to obtain information on the delivered QoE. The work in [16] proposes a QoE management framework for SDN-enabled mobile networks and consists of three modules: QoE monitoring, QoE policy and rules, and QoE enforcement through network management.

Example Context Monitoring Solution with SDN. Previously discussed metrics (network and system-level KPIs, application-level KQIs, context information) may be accumulated at the controller and provide input to the management plane. This information can be used by an application on top of the Network operating system (NOS) to perform optimization and control. This information is forwarded by the controller via the Northbound interface (NBI). There are several ways this information can be leveraged to perform efficient management of resources. For example, in the context of the video flash crowd use case presented in Sect. 3, it could be used at the CDN level to make the CDN aware of expected congestion and perform proactive rather than reactive load balancing. In the latter case this means that another dedicated application at the CDN will change the resource allocation through a dedicated controller.

Assuming some decision is taken, this decision has to be forwarded to the controller via an NBI, and finally from the controller to the device via the same SBI that has been used before. As such, both the NBI and SBI are used for monitoring in the uplink as well as management in the downlink. In Fig. 3 we illustrate a use case, namely a video service provider communicating with the

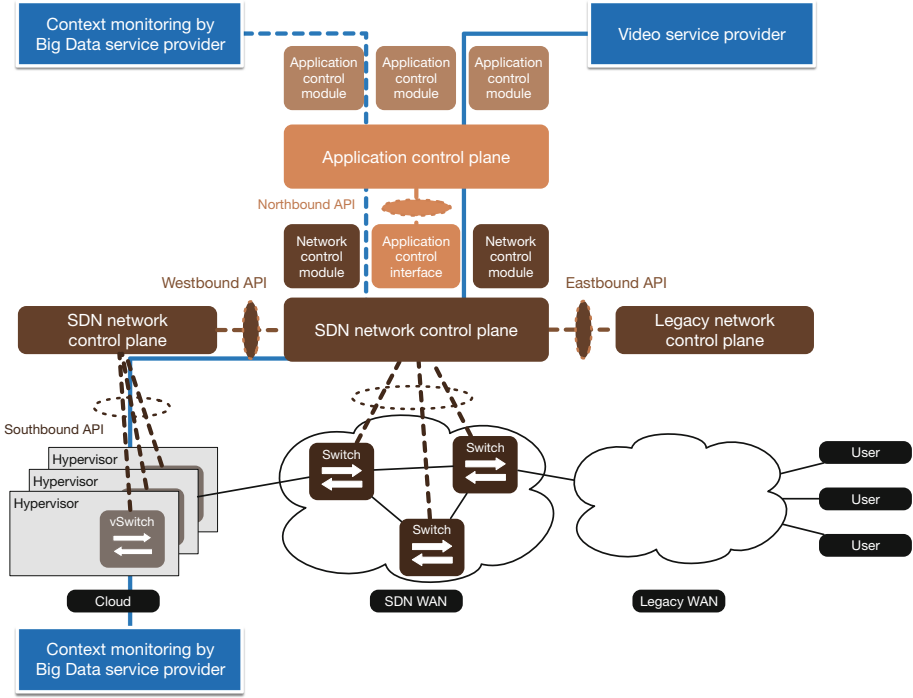


Fig. 3. An example of performing context monitoring in an SDN environment, adapted from [31] and extended to illustrate a context monitoring scenario. Other scenarios may also utilize APIs bound to different directions, e.g., context monitoring can also be conducted through the NBI.

NOS (controllers) via NBI which in turn gather the monitored data or provide the rules (i.e., how to handle the traffic) to the involved virtual and physical network elements.

5 Conclusions

This paper has provided detailed insights into how context monitoring can be beneficial for QoE management with the goal to improve both QoS and QoE. A context information classification scheme has been proposed to serve as a generic framework, which has further been applied to different context monitoring use cases with various types of services involved. The use case scenarios were selected to demonstrate the significance of context factors and the ways in which they can interrelate. The video streaming use case illustrates the benefits of having access to context data prior and during the formation of a flash crowd, and how such an event could be combated with CDN load balancing strategies. The benefits and potential exploitation of context data are further discussed in the case of on-line and cloud gaming, representing a highly demanding real-time

interactive scenario. Further use cases have highlighted some opportunities of how context monitoring can be deployed to improve QoE in future wireless and mobile networks.

Key issues to address with respect to deploying context-aware QoE monitoring and management solutions are the technical realization challenges. We focus our discussions on the SDN paradigm as it can offer a promising technical solution. We are also aware of the potential implications to and conflicts with user privacy. This topic was specifically omitted here, as it very much merits a separate discussion.

In the growing Internet of Services, QoE management will play an important role. It can be proliferated, e.g., through SDN, especially when the variety of service providers, monitored metrics and SDN controllers is considered. Finally, underlying business models will play a key role in putting an effective QoE management scheme based on enhanced monitoring into practice.

Acknowledgment. This work has been supported/partially supported by the ICT COST Action IC1304—Autonomous Control for a Reliable Internet of Services (ACROSS), November 14, 2013 – November 13, 2017, funded by European Union.

References

1. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggle, P.: Towards a better understanding of context and context-awareness. In: Gellersen, H.-W. (ed.) HUC 1999. LNCS, vol. 1707, pp. 304–307. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48157-5_29
2. Akhshabi, S., Anantakrishnan, L., Dovrolis, C., Begen, A.C.: Server-based traffic shaping for stabilizing oscillating adaptive streaming players. In: 23rd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, pp. 19–24. ACM (2013)
3. Alimi, R., Penno, R., Yang, Y., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., Woundy, R.: Application-Layer Traffic Optimization (ALTO) Protocol. RFC 7285 (Proposed Standard), September 2014. <http://www.ietf.org/rfc/rfc7285.txt>
4. Batteram, H., Damm, G., Mukhopadhyay, A., Philippart, L., Odysseos, R., Urrutia-Valdés, C.: Delivering quality of experience in multimedia networks. *Bell Labs Tech. J.* **15**(1), 175–193 (2010)
5. Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., O’Connor, B., Radoslavov, P., Snow, W., Parulkar, G.: ONOS: towards an open, distributed SDN OS. In: Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, HotSDN 2014, pp. 1–6. ACM, New York (2014). <http://doi.acm.org/10.1145/2620728.2620744>
6. Bouten, N., Latré, S., De Turck, F.: QoE-centric management of multimedia networks through cooperative control loops. In: Doyen, G., Waldburger, M., Čeleda, P., Sperotto, A., Stiller, B. (eds.) AIMS 2013. LNCS, vol. 7943, pp. 96–99. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38998-6_13
7. Boutremans, C., Le Boudec, J.Y.: Adaptive joint playout buffer and FEC adjustment for Internet telephony. In: INFOCOM 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, vol. 1, pp. 652–662, IEEE Societies, March 2003

8. Buchinger, S., Kriglstein, S., Hlavacs, H.: A comprehensive view on user studies: survey and open issues for mobile TV. In: Proceedings of the Seventh European Conference on European Interactive Television Conference, EuroITV 2009, pp. 179–188. ACM, New York (2009). <http://doi.acm.org/10.1145/1542084.1542121>
9. Chavez-Santiago, R., Szydeko, M., Kliks, A., Foukalas, F., Haddad, Y., Nolan, K.E., Kelly, M.Y., Masonta, M.T., Balasingham, I.: 5G: the convergence of wireless communications. *Wirel. Pers. Commun.* pp. 1–26 (2015). <https://doi.org/10.1007/s11277-015-2467-2>
10. Chen, J., Ghosh, A., Magutt, J., Chiang, M.: QAVA: quota aware video adaptation. In: Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies, CoNEXT 2012, pp. 121–132. ACM, New York (2012). <http://doi.acm.org/10.1145/2413176.2413191>
11. Chen, K.T., Huang, P., Lei, C.L.: How sensitive are online gamers to network quality? *Commun. ACM* **49**(11), 34–38 (2006)
12. Cheng, X., Li, H., Liu, J.: Video sharing propagation in social networks: measurement, modeling, and analysis. In: IEEE INFOCOM 2013, pp. 45–49. IEEE (2013)
13. Claypool, M., Finkel, D.: The effects of latency on player performance in cloud-based games. In: Proceedings of 13th Annual IEEE/ACM NetGames, pp. 1–6 (2014)
14. Clincy, V., Wilgor, B.: Subjective Evaluation of Latency and Packet Loss in a Cloud-Based Game. In: 2013 Tenth International Conference on Information Technology: New Generations (ITNG), pp. 473–476. IEEE (2013)
15. Cuadra-Sanchez, A., Cutanda-Rodriguez, M., Perez-Mateos, I., Aurelius, A., Brunnstrom, K., Laulajainen, J., Varela, M., de Vergara, J.: A global customer experience management architecture. In: 2012 Future Network Mobile Summit (FutureNetw), pp. 1–8, July 2012
16. Eckert, M., Knoll, T.M.: QoE management framework for internet services in SDN enabled mobile networks. In: Bauschert, T. (ed.) EUNICE 2013. LNCS, vol. 8115, pp. 112–123. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40552-5_11
17. Farshad, A., Georgopoulos, P., Broadbent, M., Mu, M., Race, N.: Leveraging SDN to provide an in-network QoE measurement framework. In: IEEE INFOCOM 2015 2nd Workshop on Communication and Networking Techniques for Contemporary Video, April 2015
18. Fiedler, M., Hoffeld, T., Tran-Gia, P.: A generic quantitative relationship between quality of experience and quality of service. *IEEE Netw.* **24** (2010)
19. Fund, F., Hosseini, S., Panwar, S.: More bars, more bang for the buck: channel-dependent pricing for video delivery to mobile users. In: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 565–570, April 2014
20. Georgopoulos, P., Elkhatib, Y., Broadbent, M., Mu, M., Race, N.: Towards network-wide QoE fairness using openflow-assisted adaptive video streaming. In: ACM SIGCOMM 2013 Workshop on Future Human-Centric Multimedia Networking, pp. 15–20. ACM (2013)
21. Ha, S., Sen, S., Joe-Wong, C., Im, Y., Chiang, M.: TUBE: time-dependent pricing for mobile data. In: Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM 2012, pp. 247–258. ACM, New York (2012). <http://doi.acm.org/10.1145/2342356.2342402>

22. Hoßfeld, T., Moldovan, C., Schwartz, C.: To each according to his needs: dimensioning video buffer for specific user profiles and behavior. In: IFIP/IEEE Workshop on QoE Centric Management 2015 (QCMAN), May 2015
23. Hoßfeld, T., Schatz, R., Varela, M., Timmerer, C.: Challenges of QoE management for cloud applications. *IEEE Commun. Mag.* **50**(4) (2012)
24. Hoßfeld, T., Tran-Gia, P., Fiedler, M.: Quantification of quality of experience for edge-based applications. In: Mason, L., Drwiega, T., Yan, J. (eds.) *ITC 2007*. LNCS, vol. 4516, pp. 361–373. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72990-7_34
25. Hoßfeld, T., Skorin-Kapov, L., Haddad, Y., Pocta, P., Siris, V., Zgank, A., Melvin, H.: Can context monitoring improve QoE? A case study of video flash crowds in the internet of services. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 1274–1277, May 2015
26. Hoßfeld, T., Skorin-Kapov, L., Heegaard, P.E., Varela, M.: Definition of QoE fairness in shared systems. *IEEE Commun. Lett.* **21**(1), 184–187 (2017)
27. Huang, C.Y., Chen, K.T., Chen, D.Y., Hsu, H.J., Hsu, C.H.: GamingAnywhere: the first open source cloud gaming system. *ACM Trans. Multimedia Comput. Commun. Appl.* **2**(3), 1–26 (2010)
28. Hürriyetoglu, A., Kunneman, F., van den Bosch, A.: Estimating the time between twitter messages and future events. In: Eickhoff, C., de Vries, A.P. (eds.) *DIR, CEUR Workshop Proceedings*, vol. 986, pp. 20–23 (2013). CEUR-WS.org
29. ITU: The E-model: a computational model for use in transmission planning (ITU-T Recommendation G.107 (02/14)). International Telecommunications Union, February 2014
30. Jarschel, M., Wamser, F., Höhn, T., Zinner, T., Tran-Gia, P.: SDN-based application-aware networking on the example of YouTube video streaming. In: 2nd European Workshop on Software Defined Networks (EWSN 2013), Berlin, Germany, October 2013
31. Jarschel, M., Zinner, T., Hoßfeld, T., Tran-Gia, P., Kellerer, W.: Interfaces, attributes, and use cases: a compass for SDN. *IEEE Commun. Mag.* **52**(6), 210–217 (2014)
32. Jiang, J., et al.: Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE. *IEEE/ACM Trans. Netw.* **22**(1), 326–340 (2014)
33. Jumisko-Pyykkö, S., Vainio, T.: Framing the context of use for mobile HCI. *Int. J. Mob. Hum. Comput. Interact.* **2**(4), 1–28 (2010). <https://doi.org/10.4018/jmhci.2010100101>
34. Koomey, J.: Growth in data center electricity use 2005 to 2010. A report by Analytical Press, completed at the request of The New York Times, p. 9 (2011)
35. Liotou, E., Tsolkas, D., Passas, N., Merakos, L.: Quality of experience management in mobile cellular networks: key issues and design challenges. *IEEE Commun. Mag. Netw. Serv. Manag. Ser.* **53**, 145–153 (2015)
36. Liotou, E., Tseliou, G., Samdanis, K., Tsolkas, D., Adelantado, F., Verikoukis, C.: An SDN QoE-service for dynamically enhancing the performance of OTT applications. In: 7th International Workshop on Quality of Multimedia Experience (IEEE QoMEX) (2015)
37. Liu, X., Dobrian, F., Milner, H., Jiang, J., Sekar, V., Stoica, I., Zhang, H.: A case for a coordinated internet video control plane. In: *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM 2012*, pp. 359–370. ACM, New York (2012). <http://doi.acm.org/10.1145/2342356.2342431>

38. Markopoulou, A., Tobagi, F., Karam, M.: Assessment of VoIP quality over Internet backbones. In: INFOCOM 2002, Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 1, pp. 150–159. IEEE (2002)
39. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: OpenFlow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.* **38**(2), 69–74 (2008)
40. Metzger, F., Liotou, E., Moldovan, C., Hößfeld, T.: TCP video streaming and mobile networks: not a love story, but better with context. *Comput. Netw.* **109**(Part 2), 246–256 (2016). <http://www.sciencedirect.com/science/article/pii/S1389128616302134>, traffic and Performance in the Big Data Era
41. Metzger, F., Rafetseder, A., Schwartz, C.: A comprehensive end-to-end lag model for online and cloud video gaming. In: PQS 2016 5th ISCA/DEGA Workshop on Perceptual Quality of Systems, pp. 20–24 (2016). <https://doi.org/10.21437/PQS.2016-5>
42. Mitra, K., Zaslavsky, A., Ahlund, C.: Context-aware QoE modelling, measurement, and prediction in mobile computing systems. *IEEE Trans. Mobile Comput.* **14**(5), 920–936 (2015)
43. Möller, S., et al.: Towards a new ITU-T recommendation for subjective methods evaluating gaming QoE. In: Proceedings of the 7th International Workshop on Quality of Multimedia Experience (QoMEX 2015), pp. 1–6 (5 2015)
44. Moreno-Vozmediano, R., Montero, R., Llorente, I.: Key challenges in cloud computing: enabling the future internet of services. *IEEE Internet Comput.* **17**(4), 18–25 (2013)
45. Nam, H., Kim, K.H., Kim, J.Y., Schulzrinne, H.: Towards QoE-aware video streaming using SDN. In: 2014 IEEE Global Communications Conference, pp. 1317–1322, December 2014
46. Qureshi, A., Weber, R., Balakrishnan, H., Gutttag, J., Maggs, B.: Cutting the electric bill for internet-scale systems. *SIGCOMM Comput. Commun. Rev.* **39**(4), 123–134 (2009). <http://doi.acm.org/10.1145/1594977.1592584>
47. Reichl, P., Egger, S., Schatz, R., D’Alconzo, A.: The logarithmic nature of QoE and the role of the Weber-Fechner law in QoE assessment. In: 2010 IEEE International Conference on Communications (ICC), pp. 1–5, May 2010
48. Reichl, P., Egger, S., Möller, S., Kilkki, K., Fiedler, M., Hossfeld, T., Tsiaras, C., Asrese, A.: Towards a comprehensive framework for QoE and user behavior modelling. In: 7th International Workshop on Quality of Multimedia Experience (QoMEX), Costa Navarino, Greece, September 2015
49. Reichl, P., Tuffin, B., Schatz, R.: Logarithmic laws in service quality perception: where microeconomics meets psychophysics and quality of experience. *Telecommun. Syst.* **52**(2), 587–600 (2013). <https://doi.org/10.1007/s11235-011-9503-7>
50. Reiter, U., Brunnström, K., De Moor, K., Larabi, M.-C., Pereira, M., Pinheiro, A., You, J., Zgank, A.: Factors influencing quality of experience. In: Möller, S., Raake, A. (eds.) *Quality of Experience: Advanced Concepts, Applications and Methods*. TSTS, pp. 55–72. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-02681-7_4
51. Roto, V., Vääätäjä, H., Jumisko-Pyykkö, S., Väänänen-Vainio-Mattila, K.: Best practices for capturing context in user experience studies in the wild. In: Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek 2011, pp. 91–98. ACM, New York (2011). <http://doi.acm.org/10.1145/2181037.2181054>

52. Schatz, R., Fiedler, M., Skorin-Kapov, L.: QoE-based network and application management. In: Möller, S., Raake, A. (eds.) *Quality of Experience: Advanced Concepts, Applications and Methods*. TSTS, pp. 411–426. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-02681-7_28
53. Schatz, R., Hößfeld, T., Janowski, L., Egger, S.: From packets to people: quality of experience as a new measurement challenge. In: Biersack, E., Callegari, C., Matijasevic, M. (eds.) *Data Traffic Monitoring and Analysis: From Measurement, Classification, and Anomaly Detection to Quality of Experience*. LNCS, vol. 7754, pp. 219–263. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36784-7_10
54. Schmid, S., Suomela, J.: Exploiting locality in distributed SDN control. In: *Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, HotSDN 2013*, pp. 121–126. ACM, New York (2013). <http://doi.acm.org/10.1145/2491185.2491198>
55. Seufert, M., Burger, V., Wamser, F., Tran-Gia, P., Moldovan, C., Hößfeld, T.: Utilizing home router caches to augment CDNs toward information-centric networking. In: *European Conference on Networks and Communications (EuCNC)*, Paris, France, June 2015
56. Skorin-Kapov, L., Varela, M.: A multi-dimensional view of QoE: the ARCU model. In: *2012 Proceedings of the 35th International Convention MIPRO*, pp. 662–666. IEEE (2012)
57. Stankiewicz, R., Jajszczyk, A.: A survey of QoE assurance in converged networks. *Comput. Netw.* **55**(7), 1459–1473 (2011). <http://www.sciencedirect.com/science/article/pii/S138912861100051X>, recent Advances in Network Convergence
58. Suznjevic, M., Beyer, J., Skorin-Kapov, L., Moller, S., Sorsa, N.: Towards understanding the relationship between game type and network traffic for cloud gaming. In: *2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pp. 1–6, July 2014
59. Suznjevic, M., Skorin-Kapov, L., Matijasevic, M.: Impact of user, system, and context factors on gaming QoE: a case study involving MMORPGs. In: *Proceedings of the 12th ACM SIGCOMM Workshop on Network and System Support for Games (2013)*
60. Weber, E.H.: *De Pulsu, resorptione, auditu et tactu. Annotationes anatomicae et physiologicae, auctore, Koehler (1834)*
61. Weiss, M., Eidson, J., Barry, C., Goldin, L., Iannucci, R.A., Lee, E., Stanton, K.: *The Case for Cross Disciplinary Research on Time Aware Applications, Computers and Communication Systems (TAACCS) (2013)*. Whitepaper
62. Wilk, S., Schönherr, S., Stohr, D., Effelsberg, W.: EnvDASH: an environment-aware dynamic adaptive streaming over HTTP system. In: *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video, TVX 2015*, pp. 113–118. ACM, New York (2015). <http://doi.acm.org/10.1145/2745197.2745205>
63. Zhang, J., Ansari, N.: On assuring end-to-end QoE in next generation networks: challenges and a possible solution. *IEEE Commun. Mag.* **49**(7), 185–191 (2011)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

