The Value of Context-Awareness in Bandwidth-Challenging HTTP Adaptive Streaming Scenarios

Eirini Liotou[1(✉)], Tobias Hoßfeld[2], Christian Moldovan[2], Florian Metzger[2], Dimitris Tsolkas[1], and Nikos Passas[1]

[1] National and Kapodistrian University of Athens, Athens, Greece
{eliotou,dtsolkas,passas}@di.uoa.gr
[2] University of Duisburg-Essen, Essen, Germany
{tobias.hossfeld,christian.moldovan,florian.metzger}@uni-due.de

Abstract. Video streaming has become an indispensable technology in people's lives, while its usage keeps constantly increasing. The variability, instability and unpredictability of network conditions pose one of the biggest challenges to video streaming. In this chapter, we analyze HTTP Adaptive Streaming, a technology that relieves these issues by adapting the video reproduction to the current network conditions. Particularly, we study how context awareness can be combined with the adaptive streaming logic to design a proactive client-based video streaming strategy. Our results show that such a context-aware strategy manages to successfully mitigate stallings in light of network connectivity problems, such as an outage. Moreover, we analyze the performance of this strategy by comparing it to the optimal case, as well as by considering situations where the awareness of the context lacks reliability.

Keywords: HTTP Adaptive Streaming · Video streaming
Context awareness · Quality of Experience · Stalling probability

1 Introduction

1.1 Motivation

The rising number of smart phone subscriptions, which are expected to reach 9.2 billion by 2020, combined with the explosive demand for mobile video, which is expected to grow around 13 times by 2019, accounting for 50% of all global mobile data traffic, will result in a ten-fold increase of mobile data traffic by 2020 [1]. This explosive demand for mobile video is fueled by the ever-increasing number of video-capable devices and the integration of multimedia content in popular mobile applications, e.g. Facebook and Instagram. Furthermore, the use of video-capable devices, which range from devices with high resolution screens to interactive head mounted displays, requires a further increase of the bandwidth, so that on-demand video playback can be supported and differentiated expectations raised by the end video consumers can be satisfied.

I. Ganchev et al. (Eds.): Autonomous Control for a Reliable Internet of Services, LNCS 10768, pp. 128–150, 2018.
https://doi.org/10.1007/978-3-319-90415-3_6

Following this trend for video streaming, mobile network operators, and service providers focus on the Quality of Experience (QoE) of their customers, controlling network or application-level parameters, respectively. In parallel, from the user's side, a better QoE enhancement can be achieved if both network- and application-level information are utilized (cross-layer approaches). On top of that, greatest gains can be possible if also "context information" is used by any of these parties, complementary to the usually available Key Performance and Key Quality Indicators (KPIs and KQIs), to which the service/network providers already have access. As a general conclusion, (ideally) cross-party, cross-layer, and multi-context information is required towards devising mechanisms that will have the greatest impact on the overall user QoE.

In parallel, since most of the consumed video of a mobile data network is delivered through server-controlled traditional HTTP video streaming, the ability of such monolithic HTTP video streaming to support a fully personalized video playback experience at the end-user is questioned. To this end, this traditional technique is gradually being replaced by client-controlled video streaming exploiting HTTP Adaptive Streaming (HAS). HAS can split a video file into short segments of a few seconds each, with different quality levels and multiple encoding rates, allowing a better handling of the video streaming process, e.g. by adapting the quality level of future video segments. HAS is a key enabler towards a fully personalized video playback experience to the user, as it enables the terminal to adapt the video quality based on the end device capabilities, the expected video quality level, the current network status, the content server load, and the device remaining battery, among others.

In this chapter, our objective is to investigate how context awareness in mobile networks can help not only understand but also enhance the user experienced quality during HAS sessions. We study a scenario where users travelling with a vehicle experience bad or no service at all (i.e. a service outage). In this or similar type of scenarios, the opportunity emerges to propose novel, preemptive strategies to overcome such imminent problems, for instance by proposing proactive adaptive streaming or buffering techniques for video streaming services. This scenario has been modelled, optimized and investigated by means of simulation. Before presenting the problem under study, we first identify the need and the changes needed to move from a QoE-oriented to a context-aware network/application management.

1.2 From QoE-Awareness to Context-Awareness

QoE is defined as "*the degree of delight or annoyance of the user of an application or service*" [2], and as such, it is an inherently subjective indication of quality. Consequently, a significant amount of research efforts has been devoted to the measurement of this subjective QoE. The goal of these efforts is to find objective models that can reliably estimate the quality perceived/experienced by the end-user. To this end, subjective experiments that involve human assessors are carefully designed, with the purpose of mapping the various quality influence

factors to QoE values. In [2], these influence factors are defined as "*any characteristic of a user, system, service, application, or context whose actual state or setting may have influence on the Quality of Experience for the user*", and they are basically classified into three distinct groups, namely, *Human*, *System*, and *Context* factors. Human influence factors include any psychophysical, cognitive, psychological or demographic factors of the person receiving a service, while system influence factors concern technical parameters related to the network, application and device characteristics and parameters. Finally, context relates to any spatio-temporal, social, economic and task-related factors.

The awareness of QoE in a network is a valuable knowledge not only per se (namely for network monitoring and benchmarking purposes) but also as a useful input for managing a network in an effective and efficient way. The "QoE-centric management" of a network can be performed as a closed loop procedure, which consists of three distinguishable steps:

QoE Modelling: For the purposes of QoE modelling, key influence factors that have an impact on the network's quality need to be mapped to QoE values. To this direction, QoE models have to be used that try to accurately reflect/predict a subjective QoE estimation.

QoE Monitoring: This step provides answers on how, where and when QoE-related input can be collected. It includes the description of realistic architectures in terms of building blocks, mechanisms, protocols and end-to-end signalling in the network. Also, this procedure relates to the way in which feedback concerning QoE measurements can be provided from end-user devices and any network nodes to the responsible QoE-decision making entities in the network.

QoE Management and Control: This step includes all the possible QoE-driven mechanisms that can help the network operate in a more efficient and qualitative way. These mechanisms may include for instance power control, mobility management, resource management and scheduling, routing, network configuration, etc. All these procedures can be managed based on QoE instead of traditional Quality of Service (QoS) criteria and their impact can be assessed based on the QoE they achieve. Multiple variants of the three previous steps or building blocks can be found in the literature, such as [3,4].

"Context" may refer to "*any information that can be used to characterize the situation of an entity*" [5]. In this way, context awareness can facilitate a transition from packet-level decisions to "scenario-level" decisions: Indeed, deciding on a per-scenario rather than on a per-packet level may ensure not only a higher user QoE but also the avoidance of over-provisioning in the network. This huge potential has been recently identified in academia and as a result, research works on context awareness and context-aware network control mechanisms are constantly emerging in the literature. In [6], a context aware handover management scheme for proper load distribution in an IEEE 802.11 network is proposed. In [7], the impact of social context on compressed video QoE is investigated, while in [8] a novel decision-theoretic approach for QoE modelling, measurement, and prediction is presented, to name a few characteristic examples.

If we now revisit the three-step QoE control loop described earlier by also considering context awareness, then this is enriched as follows:

Context Modelling: Based on the earlier discussion about the QoE modelling procedure, we may observe that the *System* as well as the *Human* influence factors are directly or indirectly taken into account in the subjective experiments' methodologies, e.g. [9]. Consequently, the impact of technical- and human-level characteristics is tightly integrated into the derived QoE models. Nevertheless, the *Context* influence factors are mostly missing in these methodologies, or are not clearly captured. This happens due to the fact that the QoE evaluations are usually performed in controlled environments, not allowing for diversity in the context of use. Besides, context factors are challenging to control, especially in a lab setting, and new subjective experiment types would have to be designed. As a consequence, the mapping of context influence factors to QoE is absent from most QoE models that appear both in the literature and in standardization bodies. Therefore, novel context-aware QoE models need to be devised that are able to accurately measure and predict QoE under a specific context of use, as these context factors are (often) neglected. These context factors could either be integrated inside a QoE model directly, or, be used as a tuning factor of an otherwise stand-alone QoE model.

Context Monitoring: On top of QoE monitoring, context monitoring procedures could (and should) be implemented in the network. These procedures will require different input information from the ones used by traditional QoS/QoE monitoring techniques. The acquired context information may be used for enhancing the QoE of the users or for the prediction of imminent problems, such as bottlenecks, and may range from spatio-temporal to social, economic and task-related factors. Some of the possible context information that may be monitored in a network is the following (to give a few examples): the current infrastructure, which is more or less static (access points, base stations, neighbouring cells, etc.), the specific user's surrounding environment (location awareness, outdoors/indoors environment, terrain characteristics, presence of blind spots such as areas of low coverage or limited capacity, proximity to other devices, etc.), the time of day, the current and predicted/expected future network load, the current mobility level or even the predicted mobility pattern of users in a cell (e.g. a repeated pattern), the device capabilities or state (e.g. processing power, battery level, storage level, etc.), the user task (e.g. urgent or leisure activity), as well as application awareness (e.g. foreground or background processes), and social awareness of the end-users, among others. Moreover, charging and pricing can also be included in the general context profile of a communication scenario. It needs to be noted here that context awareness does not necessarily rely on predicting the future (e.g. future traffic demand) but also on solid knowledge that is or can be available (e.g. time of day, outage location, etc.).

Context-Aware Management and Control: Three possibilities emerge in a context-aware network. First, the network can take more sophisticated control decisions that are also influenced by context-awareness, such as for instance,

a decision to relax the handover requirements for a user in a fast-moving vehicle or a decision to connect a device with low battery to a WiFi access point. Second, the network can actualize control decisions exploiting the current context. For instance, it can exploit information about flash crowd formation to drive an effective Content Distribution Network (CDN) load balancing strategy [10] or, more generally, to take control decisions proactively based on context information about the near future. Finally, context-awareness can help to take decisions with the objective to increase the network efficiency as measured in spectrum, energy, processing resources, etc., and consequently to reduce operational expenses. For instance, context information could allow for a more meaningful distribution of the network resources among competing flows that refer to different communication scenarios.

This book chapter handles a characteristic use case of context-aware management, to showcase its potential. More specifically, we study a scenario where "context awareness" refers to awareness of the location and duration of a forthcoming outage, namely of a restricted area of very low or zero bandwidth (e.g. limited coverage due to physical obstacles or limited capacity due to high network congestion). Based on this knowledge, we devise a proactive HAS strategy that will enhance the viewing experience of a user travelling inside a vehicle towards this area.

Related work involves HAS strategies that use geo-location information ([11,12]), and evoke users to send measurements regarding their data rate, so that an overall map of bandwidth availability can be created for a certain area. Other HAS techniques rely on prediction, rather than context-awareness. For instance, [13] describes a HAS method where higher quality segment requests are a posteriori replaced with lower ones, as soon as a zero-bandwidth spatio-temporal event is identified. Moreover, similarly to our approach, [14] proposes an anticipatory HAS strategy, which requires prediction of the channel state in terms of Received Signal Strength (RSS) and proactively adjusts the user's buffer. An optimization problem is formulated that minimizes the required number of spectrum resources, while it ensures the user buffer is better prepared for an imminent coverage loss. The authors even conducted a demo of this approach in [15] that serves as a proof of concept. Our difference with this approach, is that we rely on longer-term context-awareness rather than imminent channel prediction, and that instead of manipulating the user buffer size, we proactively adapt the video quality selection. Finally, [16] combines RSS information with localization sensors from the smart phones that reveal the user's coverage state and help achieve a smoother and stabler HAS policy, called Indoors-Outdoors aware Buffer Based Adaptation (IOBBA).

2 System Analysis

2.1 System Model

The environment under study is a mobile cellular network. We consider a cell, where one base station is offering connectivity to multiple users, residing inside

the cell. Here we focus on TCP-based video streaming service users (e.g. YouTube videos) and therefore, we focus only on the Downlink (DL).

Due to the challenges introduced by the access part of the network, namely due to pathloss, shadowing, fading and penetration losses, as well as due to the mobility of the users within this cell, the channel strength and quality may fluctuate significantly from user to user, from location to location, and from time to time. The existence of an outage inside a cell poses a high risk for the viewing experience of mobile video streaming users, since it might lead to a stalling event.

In the context of this scenario and with the assistance of Fig. 1, we can mathematically represent the system model and problem statement. Assume that a video streaming user is inside a vehicle (such as a bus or train), which is travelling with a particular direction and with a specific speed. We assume, that the positioning and the length of an upcoming outage are known in advance (due to context awareness). As a result, the remaining distance between the vehicle and the outage's starting point is also available at the client side. This distance corresponds to a travelling time of t_{dist}, namely the time required until the user enters the outage region. Let b be the current buffer status of this user's HAS application; Then, during t_{dist}, this buffer level will be boosted by b_+ but also reduced by b_-. Similarly, throughout the outage duration, the buffer will be boosted by $b_{outage+}$ but also reduced by $b_{outage-}$. When the user enters (exits) the outage region, the application's buffer level will be $b_{outage-in}$ ($b_{outage-out}$), respectively, and it will hold that:

$$b_{outage-in} = b + b_+ - b_- \tag{1}$$

$$b_{outage-out} = b_{outage-in} - b_{outage-} \tag{2}$$

because $b_{outage+}$ is assumed equal to zero, namely there is negligible or no connection to the base station inside the outage region. Then, we can express the objective of the proposed HAS strategy as the following:

$$b_{outage-out} \geq b_{thres} \tag{3}$$

which means that when the vehicle is exiting the outage region, the buffer status of the HAS application should be at least equal to the minimum buffer threshold, b_{thres}, which ensures that the video playout continues uninterrupted. Note that, a stalling always occurs when $b < b_{thresh}$. The last condition can be re-written as:

$$b_+ \geq b_{thres} + b_- + b_{outage-} - b \tag{4}$$

This condition answers the question about how much should the buffer of the HAS application be pro-actively filled during t_{dist} (namely from the time of reference up to the outage starting point), so that no stallings will occur. This should be achieved despite the imminent connection disruption. Note that all the parameters on the right hand side are known to the client or can be easily estimated (b_{thres} is fixed, b is directly known to the client application, while b_-, $b_{outage-}$ can be estimated). It needs to be stressed out that all previous buffer-related variables may be expressed either in seconds, i.e. buffer playtime, or in bytes, i.e. buffer size.

Based on the previous system model we can estimate the b_+, namely the required buffer boost (in bytes or in seconds) to avoid any stalling during the outage duration. This measurement can be then further translated to a required "advance time", t_{adv}, until which the travelling user needs to be notified about the existence of the outage (namely, its starting position and duration), in order to run the proactive HAS strategy proposed here. We assume that the users switch from a standard HAS strategy to the adapted one exactly at t_{adv}. We can express b_+ as a function of t_{adv} as follows:

$$b_+ = r * t_{adv} \tag{5}$$

where r (bytes per second) is the estimated data rate by the client's application. Namely, r is the user's prediction of the available network bandwidth, as estimated by the HAS strategy. Therefore, the minimum required advance time in order to avoid any stalling would be:

$$t_{adv} \geq \frac{b_{thres} + b_- + b_{outage-} - b}{r} \tag{6}$$

To avoid a stalling, t_{adv} should be less than the remaining t_{dist}, namely the user should be notified early enough to react.

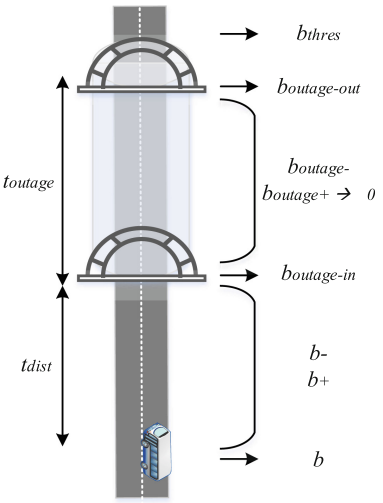


Fig. 1. Problem description using buffer status information.

2.2 Optimization Problem

The goal of this section is to formulate a problem that achieves optimal segment selection with respect to three different optimization objectives, described next. The optimization problem is formulated using the following notation[1]:

[1] [17] is used as a reference.

- τ is the length of each segment in seconds.
- T_0 is the initial delay of the video.
- D_i is the deadline of each segment i, meaning that this segment needs to be completely downloaded up to this point.

Then:

$$D_i = T_0 + i\tau, \quad \forall i = 1, ..., n. \tag{7}$$

Also:

- n is the total number of segments that comprise the video.
- r_{max} is the maximum number of available layers/representations.
- x_{ij} represents segment i of layer j.
- w_{ij} is the weighting factor for the QoE of segment i of layer j. Here, we use the quality layer value as weighting factor = {1,2,3}.
- S_{ij} is the size of segment i of layer j (e.g. in bytes).
- $b(t)$ is the total data downloaded until the point in time t. We assume perfect knowledge of $b(t)$.
- α is the weight for the impact of the quality layer and β for the impact of the switches ($\alpha + \beta = 1, \alpha > 0, \beta > 0$).

QoE studies on HAS (e.g. [18,19]) have revealed that major quality influence factors are in order of significance: *a*) the layers selected and especially the time spent on highest layer and *b*) the altitude, i.e. the difference between subsequent quality levels (the smaller the better). Other factors with less significance are: the number of quality switches, the recency time and the last quality level. Taking these findings into account, we focus on three different types of optimization objectives, which aim to maximize the positive impact of higher level selection, deducing the negative impact of quality switches and altitude. Three different versions of optimization objectives are thus formulated, as follows:

- Optimal strategy "W" accounts only for the impact of the quality layers, trying to maximize their value, so that the highest layer will be favored over the intermediate layer, which will be preferred over the lowest layer.
- Optimal strategy "W+S" additionally accounts for the number of switches, trying to minimize their occurrence.
- Optimal strategy "W+S+A" additionally accounts for the altitude effect, trying to minimize the distance between subsequent layers, thus preferring direct switches e.g. from layer 1 to layer 2 rather than from layer 1 to layer 3.

This leads us to the three different formulations of the optimization problem for one user:

- W: Maximize the quality layer values:

$$\text{maximize} \sum_{i=1}^{n} \sum_{j=1}^{r_{max}} \alpha w_{ij} x_{ij} \tag{8}$$

– W+S: Maximize the quality layer values minus the number of switches:

$$\text{maximize} \sum_{i=1}^{n} \sum_{j=1}^{r_{max}} \alpha w_{ij} x_{ij} - \frac{1}{2} \sum_{i=1}^{n-1} \sum_{j=1}^{r_{max}} \beta (x_{ij} - x_{i+1,j})^2 \tag{9}$$

– W+S+A: Maximize the quality layer values minus the number of switches and the altitude difference:

$$\text{maximize} \sum_{i=1}^{n} \sum_{j=1}^{r_{max}} \alpha w_{ij} x_{ij} - \frac{1}{2} \beta \sum_{i=1}^{n-1} \sum_{j=1}^{r_{max}} \left[(x_{ij} - x_{i+1,j})^2 + \frac{(x_{ij} - x_{i+1,p})^2}{|p - j|} \right] \tag{10}$$

where

$$p = \{1..r_{max}\} - \{j\}$$

Despite its complication, the terms in the last parenthesis of Eq. (10) represent the preference over switches between "neighbor" layers (i.e. after a layer 1 selection, layer $p = 2$ switches will be preferred/after a layer 2 selection, either layer $p = 1$ or $p = 3$ switches will be preferred/while after a layer 3 selection, layer $p = 2$ switches will be preferred).

All above optimization objectives are subject to the following constraints:

$$x_{ij} \in \{0, 1\} \tag{11}$$

$$\sum_{j=1}^{r_{max}} x_{ij} = 1, \quad \forall i = 1, ..., n \tag{12}$$

$$\sum_{i=1}^{k} \sum_{j=1}^{r_{max}} S_{ij} x_{ij} \leq b(D_k), \quad \forall k = 1, ..., n \tag{13}$$

The three constraints in this problem are interpreted as follows: x_{ij} is a binary value (Eq. (11)) meaning that a segment is either downloaded or not, each segment has to be downloaded in exactly one layer (Eq. (12)), and all segments need to have been downloaded before their deadline, so that no stalling occurs (Eq. (13)).

2.3 HAS-Based Strategy

The proposed strategy needs to avoid stallings during the outage, something which is extremely high likely to occur due to the very low network coverage. The main idea to ensure that is to pro-actively lower the requested quality level of the next segments a priori, i.e. before entering the outage area. As a consequence, the buffer at the user side when entering the outage region will be fuller than it would have been without such a scheme (see Fig. 2)[2].

As a result of this strategy, the user viewing experience will be less affected, not only because the video will continue to play without a stalling for a longer

[2] This figure is adapted from [20].

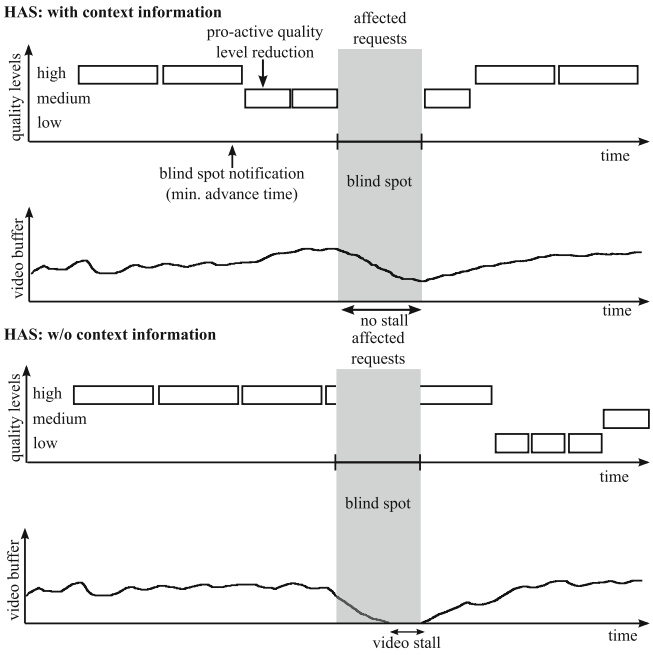


Fig. 2. Adaptive video streaming scenario with and without context awareness.

period of time, or hopefully will never stall depending on the outage duration, but also because the quality level will be *gradually* decreased and thus the user will be better acquainted with lower quality levels. Such progressive quality degradations would be preferred in comparison to sudden and unexpected quality degradations, especially if the quality level is already very high (cf. the IQX hypothesis [21]). Consequently, the objective of the proposed strategy is to compute the optimal context-based quality level selection to ensure the best QoE while avoiding any stallings.

The HAS strategy is based on the estimation of the required buffer boost b_+ as this was described in Sect. 2.1. As for the estimation of the expected downlink rate (network bandwidth prediction), this is assumed equal to the segment rate. The segment rate estimation (in bytes per second) is done over a sliding window of the past k downloaded segments as follows:

$$r = (1-w) * \frac{Size\,of\,last\,(k-1)\,segments}{Time\,to\,download\,(k-1)\,segments} + w * \frac{Size\,of\,segment\,k}{Time\,to\,download\,segment\,k} \tag{14}$$

where w is the weight (importance) given to the latest downloaded segment. Based on this rate estimation, the expected bytes that can be downloaded until the user enters the outage region is:

$$b_{+expected} = r * t_{adv}, \quad (in\,bytes) \tag{15}$$

while the minimum required buffer playtime to exit the outage region and avoid a stalling is:

$$b_+ = b_{thres} + b_- + b_{outage-} - b, \quad (in\, seconds) \tag{16}$$

Therefore, the required bytes per segment are:

$$required\, video\, rate = \frac{b_{+expected}}{b_+}, \quad (in\, bytes\, per\, second) \tag{17}$$

Note that the higher the outage duration, the larger the b_+ and thus the lower the required video rate (lower layer selection). Based on the required video rate estimation, the HAS strategy will request the highest possible representation j that fulfills this condition:

$$\frac{S_{ij}}{\tau} \leq required\, video\, rate \tag{18}$$

Namely, the layer j that will be requested will be the highest one that yields a video bit rate less or equal to this estimation. The "required video rate" estimation may be updated each time in order to account for the most recently achieved data rate r. Alternatively, an average value may be calculated in the beginning (on t_{adv}) and assumed valid until entering the outage region. In the case that the actual available data rate for this user is less than his subjective rate estimation, r, there is, however, a higher risk of stalling. We assume that the player requests the lowest layer when initialized.

2.4 QoE Models

The QoE models that are used in this work are the following:

- A QoE model for HAS, where no stallings are assumed. This model can be found in [17] and it can be described by the following formula:

$$QoE = 0.003 * e^{0.064*t} + 2.498 \tag{19}$$

  where t is the percentage of the time that the video was being played out at the highest layer (here layer 3).
- A QoE model for TCP-based video streaming, if stallings occur. This model can be found in [22] and it is described as follows:

$$QoE = 3.5 * exp(-(0.15 * L + 0.19) * N) + 1.5 \tag{20}$$

  where N is number of stalling events and L is the stalling length.

For the purposes of this scenario we combine the two aforementioned models, so that in case that no stalling has occurred, the former QoE model is used, while during and after a stalling event, we use the latter.

2.5 Realization in the Network

In this section, we provide some insights regarding the realization of the proposed scheme in a real network. Specifically, the information required so that this framework can work already is or can become easily available, namely:

- The existence and duration of an imminent outage. We assume that "Big Data" collection by the mobile operators regarding the connectivity of their subscribers can ensure the availability of this information.
- The user's moving direction and speed. This can be obtained via GPS information (current location, speed and direction combined with a map).
- The minimum advance time t_{adv} or minimum advance distance x_{adv} at which the user has to initiate the proactive HAS strategy. There are two options here: either the user knows about the outage a priori and therefore switches to the enhanced HAS mode on t_{adv} without any network assistance, or the user becomes aware of the outage existence, starting point and length on t_{adv} by the network and then switches to the enhanced HAS mode. In the first case, the user runs an internal algorithm to estimate the t_{adv}.
- Standard information required for the operation of HAS, namely video segment availability, network bandwidth estimation, and current buffer state.

As far as the need for "Big Data" mentioned before is concerned, this may take two forms: Either they could be data collected at the device itself because the user has the same travel profile every day and, therefore, learns about any coverage problems on his way, or, the data are collected at a central network point (e.g. at a base station or a server) through measurements collected by any devices passing from there. Actually, in Long Term Evolution (LTE) networks, such measurements are already available via "Channel Quality indicators - CQI". CQIs report to the LTE base station (eNB - evolved NodeB) about the quality of the received signals (SINR - Signal to Interference plus Noise Ratio) using values between 1 (worst) and 15 (best). Currently, CQIs are used only for real-time decisions such as scheduling; however, we may envision that CQIs may be collected by an eNB on a longer-term time scale (days or weeks), and be used in order to create a "coverage profile" of the cell. Following such past information, proactive measures could be taken at a cell for users travelling towards problematic areas (e.g. a physical tunnel ahead).

3 Evaluation Results

For the purposes of evaluation we use Matlab simulation. The client's buffer is simulated as a queuing model, where the "DOWNLOADED" segments are arrivals and the "PLAYED" segments are departures. To simulate the network traffic, we use real traces recorded from a network [23]. Moreover, to simulate congestion we use the parameter "bandwidth factor[3]", which is a metric of the network congestion/traffic and takes values between 0 and 1 (the higher this factor the lower the congestion).

[3] The bandwidth factor concept is extracted from [17].

The parameters used in our simulation are presented at Table 1:

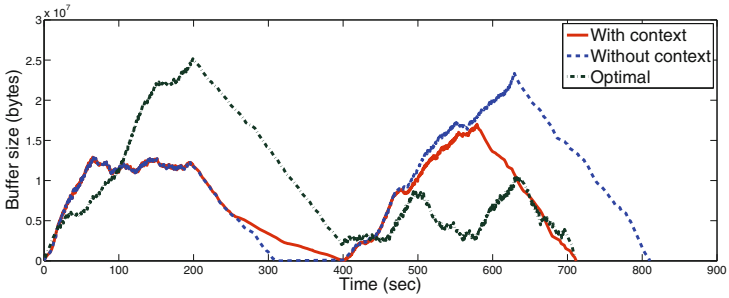
Table 1. Simulation parameters.

Parameter	Value
Segment duration	2 s
Number of video segments	350
Number of different representations (layers) per segment	3
Buffer playout threshold (initial delay)	10 segments
Outage starting point	200 s after simulation start
Outage duration	[0..400] s
HAS policy sliding window	50 segments
Bandwidth factor	0.8 (unless variable)
Replications	30, with different network traces each

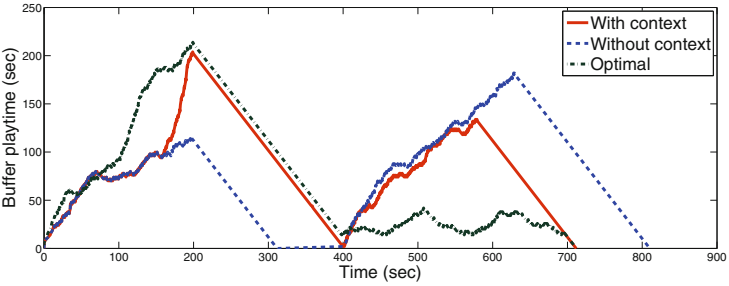
3.1 Proof of Concept

The first evaluation study mainly serves as a proof of concept of the enhanced HAS logic. The goal is to demonstrate how a context-aware HAS policy can help overcome an otherwise inevitable buffer depletion and thus, an imminent stalling event. To demonstrate that, we plot four different metrics: (a) the client buffer size in bytes, (b) the client buffer size in seconds (i.e. buffer playtime), (c) the HAS layers selected for each played out segment, and finally (d) the QoE evolution in time for the travelling user. For the latter, we make the assumption that the QoE models presented in Sect. 2.4 hold also in a real-time scale, and that the QoE model for HAS holds for the tested scenario where three different layers are available per segment. Real-time estimation of the QoE for a particular user means that QoE is estimated at every time instant t using as input accumulated information about the percentage of time that this user has already spent watching the video at layer 3 up to instant t, as long as no stalling has occurred yet, or information about the number N and duration L of stalling events since $t = 0$ up to instant t, as long as at least one stalling has occurred.

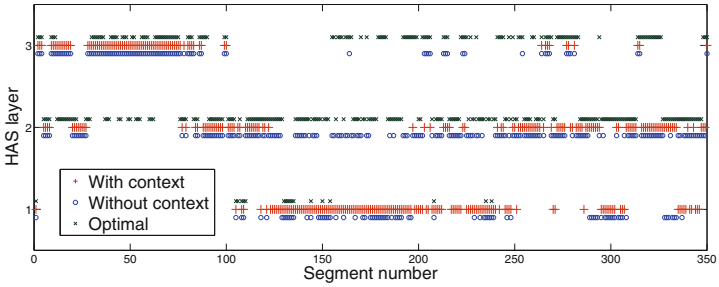
As shown in Fig. 3, three different cases are considered, namely (a) the conventional case, where no context awareness about the outage event is available, and consequently, the standard HAS strategy is implemented, (b) the case where context awareness about the starting point and duration of the outage event is available, which leads to the selection of the adapted, proactive HAS strategy, and finally (c) the optimal case (W) described in Sect. 2.2. Examining Figs. 3a and b we can see that a stalling of around 80 s is completely avoided when context awareness is deployed, or when optimal knowledge is assumed. The explanation behind the prevention of the stalling lies in Fig. 3c. In the "without context" case higher HAS layers are selected as compared to the "with context" case



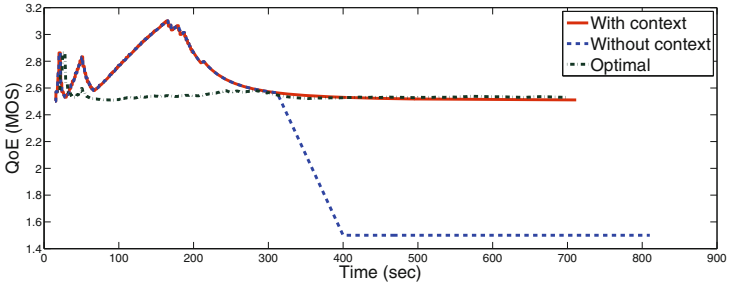
(a) Buffer size evolution over time.



(b) Buffer playtime evolution over time.



(c) Layers selected.



(d) QoE perceived over time.

Fig. 3. Client behavior with context awareness, without context awareness, and optimal behavior (W).

(mainly with layer = 2), especially around the outage occurrence, which here starts and ends at 200 s and 400 s, respectively. Having downloaded lower HAS layers in the "with context" case, the buffer of the client is fuller in terms of playtime than it would have been if higher HAS layers had been downloaded instead. The impact on QoE for all cases is also presented in Fig. 3d, where we can see that even a single stalling event of a few seconds' duration has a significantly deteriorating impact on the perceived QoE, as compared to the selection of lower HAS layers. QoE values per strategy follow the trend of layer selection: this is why the "context case" at some periods reveals higher QoE than the "optimal" case (the former requests more layer 3 segments before the outage).

Comparing now the enhanced HAS strategy with the optimal strategy, we observe that the latter does a better job in selecting higher quality layers (especially layer 2 segments) up to the point of the outage start. The reason is that the optimal strategy has full awareness of the future network conditions and thus, can take more informed decisions that lead to the highest layer selection with zero stalling risk.

3.2 Required Advance Time Estimation ("Context Time")

Next, we study how the outage duration influences the required advance time, t_{adv} and present the results in Fig. 4 (mean and standard deviation). We observe an intuitively expected trend, i.e. that the user needs to initiate the proactive HAS strategy earlier for longer outage durations (i.e. a higher t_{adv} is required). In this way, the user has more time to buffer sufficient playtime. Moreover, the standard deviation follows the same trend, indicating higher uncertainty for longer outages. The required advance time strongly depends on the achieved data rate per user, which for the purposes of simulation is a result of the network traces and bandwidth factor.

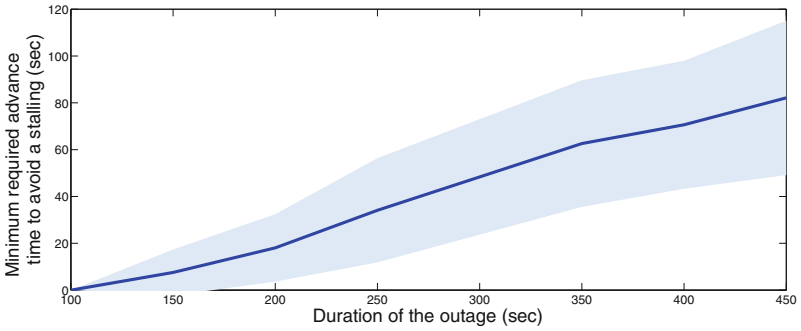
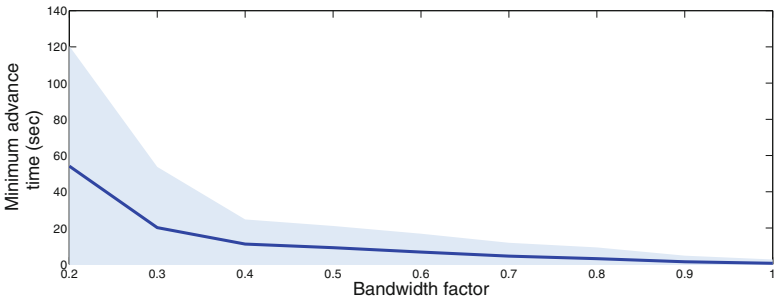


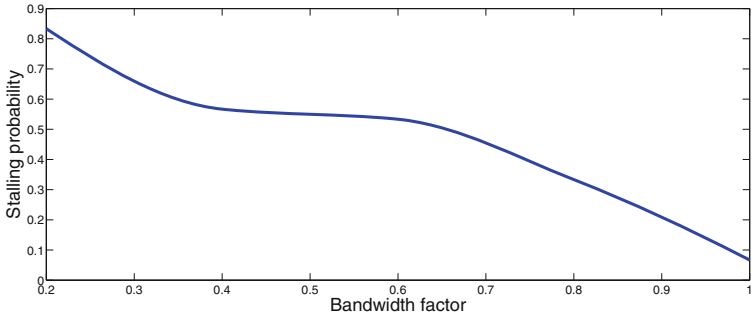
Fig. 4. The minimum required advance time (t_{adv}) to avoid a stalling event during an outage.

3.3 Comparison of Different Strategies

Next we perform a study with respect to the availability of bandwidth, in order to evaluate how HAS performs in bandwidth-challenging scenarios. Since we use real traces as input information about the data rates in the network, we can indirectly enforce a network congestion by multiplying the measured bandwidth with the aforementioned bandwidth factor.



(a) With context awareness: Minimum required advance time (t_{adv}) to avoid a stalling event in light of an outage event of 150sec for various bandwidth factors.



(b) Without context awareness: Stalling probability for various bandwidth factors.

Fig. 5. Simulation results for various bandwidth factors with and without context awareness.

The purpose of the first study with regard to the bandwidth factor is to investigate how it influences the minimum advance time t_{adv} in the case of context awareness, and how it influences the stalling probability in the conventional context unaware case. The results are presented in Fig. 5. As demonstrated in Fig. 5a, for very low data rates (e.g. a bandwidth factor of 0.2), the minimum required advance time gets higher, as the user would need a much greater time-margin to proactively fill the buffer in light of the outage, because the network is heavily congested. Moreover, the uncertainty in this case is also very high,

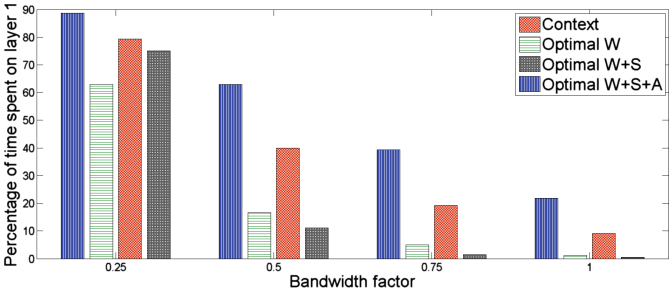
a conclusion that we have seen at the previous section as well. On the contrary, the more relaxed the network conditions, the higher the margin for an early notification about the outage, while this practically gets zero seconds (i.e., no notification is needed) when the network conditions are very relaxed (bandwidth factor = 1). Similar conclusions are drawn for the context-unaware case with regard to the stalling probabilities for different bandwidth factors, namely the less this factor, the higher the stalling probability, as expected (Fig. 5b).

Next, we compare the behaviour of the three different types of the optimal strategy (i.e. cases W/W+S/W+S+A, as described in Sect. 2.2) both among them, but also with the context-aware strategy. In Figs. 6a–d, the percentage of time spent on each of the three layers as well as the resulting number of switches are presented per strategy. All four strategies follow a similar trend as bandwidth availability increases, that is higher and higher layer 3 segments are selected, while lower and lower layer 1 segments are selected. With respect to layer 2 segments, the behaviour is different when the bandwidth factor changes from 0.25 to 0.5 (increasing layer 2 selection) from when it changes from 0.5 to 1 (decreasing layer 2 selection). Note that a bandwidth factor of 0.25 represents very high congestion and a bandwidth factor of 1 represents very low congestion.

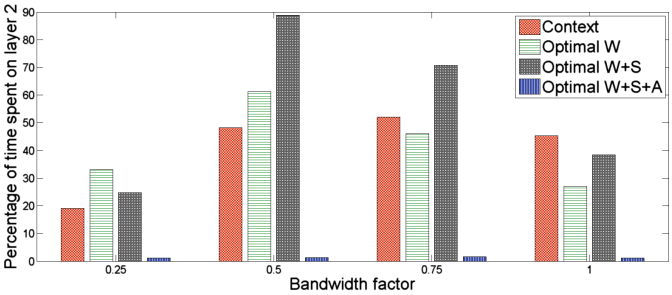
Another interesting observation is that strategy W+S+A "avoids" layer 2 segments almost completely. The reason behind that is that layer 2 in W+S+A is mostly used as a "transition step" to switch to layer 1 or layer 3, respecting the objective to keep the altitude of two sequential layers as low as possible. Equation (10) gives the same priority to staying at the same layer and to switching to a +1 or −1 layer. Perhaps, this is not necessarily the best action in terms of QoE, but there is no complete HAS QoE model to be able to build the perfect optimization function. However, the optimization goal of low altitude between successive layers holds. On the contrary, strategy W+S has a tendency to select many layer 2 segments, which is explained by its goal to minimize the switches and thus operate at a stable but safe level. We have also tested a "W+A" optimal strategy (not mentioned in Sect. 2.2), but this has been found to cause too many quality switches; therefore it was not considered for further investigation.

It is important to note that no optimal strategy is considered "better" than the other; They all represent how different optimization objectives behave under varying bandwidth conditions. However, once a validated multi-parameter QoE model for HAS becomes available in the future, the optimization problem could be revisited.

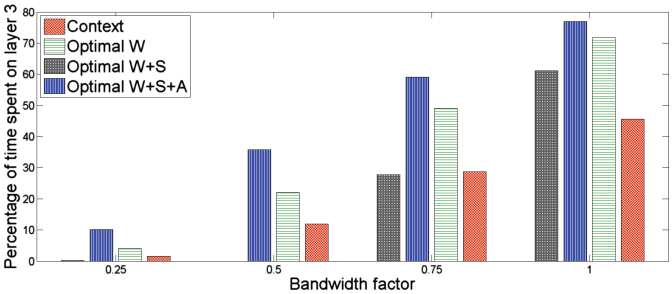
In terms of quality switches caused, which is another important QoE impairment factor, the context aware strategy and the optimal W strategy cause the highest number of switches, since they do not take measures to prevent them (see Fig. 6d). On the contrary, the optimal W+S and optimal W+S+A strategies cause the least number of switches. Between the last two, W+S+A causes more switches, as it puts equal priority to mitigating switches and keeping the altitude of any switches at a low level.



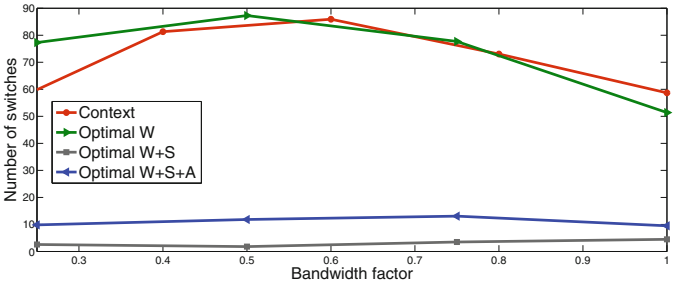
(a) Percentage of time spent on layer 1.



(b) Percentage of time spent on layer 2.



(c) Percentage of time spent on layer 3.



(d) Number of switches.

Fig. 6. Simulation results for various bandwidth factors for the three optimal cases W/W+S/W+S+A as well as the context-aware strategy.

3.4 The Impact of Unreliability of Context Information

In this section we study how unreliability in the context information influences the probability of having a stalling event. In other words, we study how risky the proactive HAS strategy is to lead to a stalling, when accurate information about the outage starting point is missing or when it is impossible to have this information on time.

For the purposes of this experiment, we assume that the buffer of the user is not limited, and therefore the user will continue to download as many bits as its connectivity to the base station allows. As a consequence, the starting point of the outage plays an important role, since the further away it is from the vehicle's current location, the fuller the buffer of the client will be under normal circumstances up to that point. Thus, also the stalling probability will be lower. Overall, this study evaluates to what extent an unexpected outage is mapped to a stalling probability.

The results under this perspective are presented in Fig. 7. As expected, the further away the outage, the less the stalling probability. However, it might be more meaningful to conduct the same study assuming a limited buffer size of the client's application, which is a more realistic assumption. In that case, we would expect that the starting point of the outage would not play such a crucial role, but the maximum size of the buffer would. Note that a normal value for an upper threshold in the number of buffered segments would be 50 segments. However, this study still provides some insights about the impact of unexpectancy regarding the outage starting point.

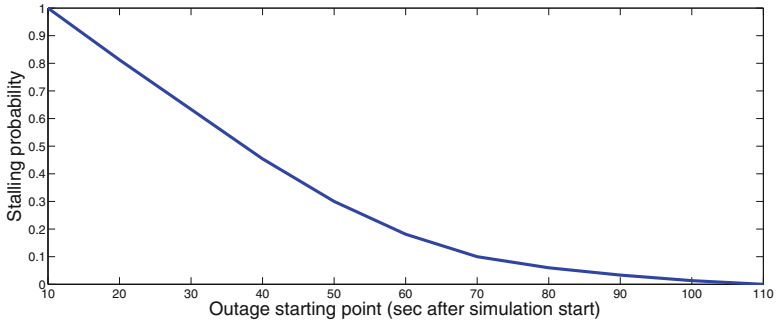
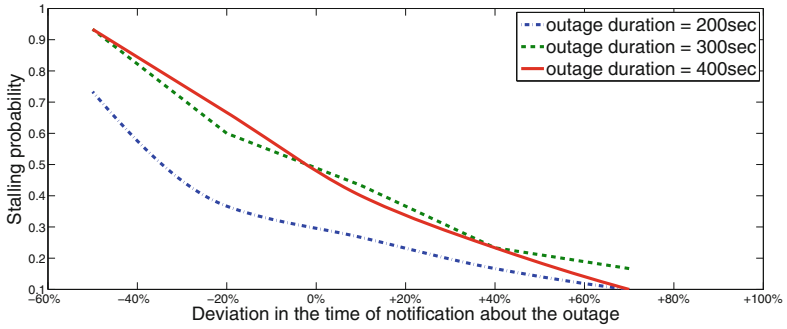


Fig. 7. The impact of the outage starting point on the stalling probability.

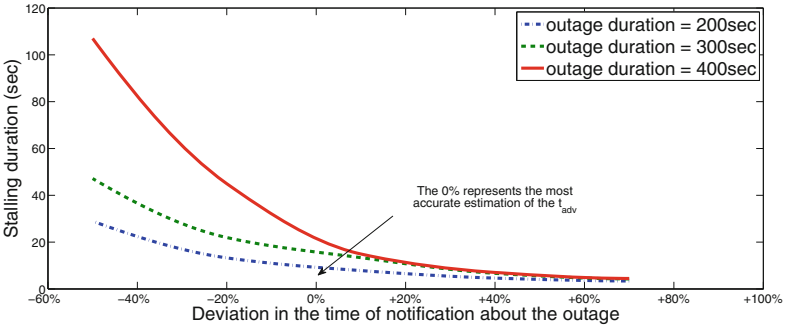
Next, we would like to investigate what happens if the context information is not communicated to the client as 100% accurate or, similarly, if it is not communicated early enough in advance (so it is accurately communicated but with some delay). Specifically, we assume that the information about the t_{adv} deviates from its mean value, as this was estimated in Sect. 3.2. This mean value is considered to represent a "0% deviation" in the following figures. From Figs. 8a and b, which represent the stalling probability and stalling duration respectively,

we draw two main conclusions. Firstly, we confirm that the mean values of t_{adv} are not enough to prevent a stalling, due to the fact that standard deviations have not been taken into account. In fact, as presented in Sect. 3.2, the standard deviations are higher for larger outage lengths and thus we observe higher stalling probabilities for the 0% values (compare the three plots per figure).

A second important conclusion, which is the emphasis of this simulation study, is that a potential uncertainty in this context information can lead to inevitable stallings. This is interpreted both in terms of stalling probabilities and stalling lengths. This emphasizes the need for accurate and timely context information, which also takes into account statistical metrics such as the standard deviation.



(a) Resulting stalling probability.



(b) Resulting stalling duration.

Fig. 8. Stalling effects when t_{adv} deviates from its mean value.

4 Conclusions

In this chapter, a novel proactive HAS strategy has been proposed and evaluated, demonstrating significant benefits as compared to the current approaches in terms of QoE and meaningful KPIs such as stalling probability. The proposed

strategy can successfully help prevent stallings at the client's HAS application during network coverage problems. The "cost to pay" is the collection and signalling of context information, which could however be realistically implemented; therefore its adoption in a real network should not be difficult.

Even though this work focused on outage conditions of zero bandwidth, we could easily extend this solution to a more general problem where bandwidth may be insufficient (but not zero). Similarly, the same problem could be adjusted for cases of an imminent service disruption such as a handover, where the aforementioned HAS strategy can help prevent stallings during the disruption period (i.e. the handover period). This may become possible by exploiting handover-hinting information, a priori. In this way, the user will be better prepared for a potential interruption in his viewing experience.

It would be also interesting as future work to study a scenario of more than one mobile video streaming users using HAS, and investigate how the decisions of one user potentially affect the others. Stability and fairness issues, together with QoE analysis would be of great interest in this case.

Finally, as a general comment, we would like to point out that this work could be revisited once a standard QoE model for HAS becomes available. In that case, we could have the opportunity not only to produce a more accurate optimization problem, but also to enhance the proposed HAS strategy, focusing on the key factors that mostly influence the end-users' QoE.

References

1. Ericsson Mobility Report: Mobile World Congress Edition, February 2015
2. Qualinet White Paper on Definitions of Quality of Experience, March 2013
3. Cuadra-Sánchez, A., et al.: IPNQSIS project, Deliverable 2.2: "Definition of requirements of the management systems to keep up with QoE expectations based on QoS and traffic monitoring" (2011)
4. Liotou, E., Tsolkas, D., Passas, N., Merakos, L.: Quality of experience management in mobile cellular networks: key issues and design challenges. Commun. Mag. IEEE **53**(7), 145–153 (2015)
5. Abowd, G.D., Dey, A.K., Brown, P.J., Davies, N., Smith, M., Steggles, 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
6. Sarma, A., Chakraborty, S., Nandi, S.: Context aware handover management: sustaining QoS and QoE in a public IEEE 802.11e hotspot. IEEE Trans. Netw. Serv. Manage. **11**(4), 530–543 (2014)
7. Zhu, Y., Heynderickx, I., Redi, J.A.: Understanding the role of social context and user factors in video quality of experience. Comput. Hum. Behav. **49**, 412–426 (2015)
8. Mitra, K., Zaslavsky, A., Ahlund, C.: Context-aware QoE modelling, measurement and prediction in mobile computing systems. IEEE Trans. Mob. Comput. **99**(PrePrints), 1 (2014)
9. Recommendation ITU-T P.800: Methods for Subjective Determination of Transmission Quality (1998)

10. 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
11. Riiser, H., et al.: Video streaming using a location-based bandwidth-lookup service for bitrate planning. ACM Trans. Multimed. Comput. Commun. Appl. **8**(3), 24:1–24:19 (2012)
12. Hao, J., et al.: GTube: geo-predictive video streaming over HTTP in mobile environments. In: Proceedings of the 5th ACM Multimedia Systems Conference, MMSys 2014, pp. 259–270. ACM, New York (2014)
13. Ramamurthi, V., Oyman, O., Foerster, J.: Using link awareness for HTTP adaptive streaming over changing wireless conditions. In: 2015 International Conference on Computing, Networking and Communications (ICNC), pp. 727–731, February 2015
14. Sadr, S., Valentin, S.: Anticipatory buffer control and resource allocation for wireless video streaming. CoRR abs/1304.3056 (2013)
15. Mekki, S., Valentin, S.: Anticipatory quality adaptation for mobile streaming: fluent video by channel prediction. In: 2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1–3, June 2015
16. Mekki, S., Karagkioules, T., Valentin, S.: HTTP adaptive streaming with indoors-outdoors detection in mobile networks. CoRR abs/1705.08809 (2017)
17. Hoßfeld, T., Seufert, M., Sieber, C., Zinner, T., Tran-Gia, P.: Identifying QoE optimal adaptation of HTTP adaptive streaming based on subjective studies. Comput. Netw. **81**, 320–332 (2015)
18. Seufert, M., Egger, S., Slanina, M., Zinner, T., Hoßfeld, T., Tran-Gia, P.: A survey on quality of experience of HTTP adaptive streaming. IEEE Commun. Surv. Tutor. **17**, 469–492 (2015)
19. Hoßfeld, T., Seufert, M., Sieber, C., Zinner, T.: Assessing effect sizes of influence factors towards a QoE model for HTTP adaptive streaming. In: 2014 Sixth International Workshop on Quality of Multimedia Experience (QoMEX), pp. 111–116, September 2014
20. Metzger, F., Steindl, C., Hoßfeld, T.: A simulation framework for evaluating the QoS and QoE of TCP-based streaming in an LTE network. In: 27th International Teletraffic Congress (ITC 27), September 2015
21. Fiedler, M., Hoßfeld, T., Tran-Gia, P.: A generic quantitative relationship between quality of experience and quality of service. Netw. IEEE **24**(2), 36–41 (2010)
22. Hoßfeld, T., Schatz, R., Biersack, E., Plissonneau, L.: Internet video delivery in youtube: from traffic measurements to quality of experience. In: Biersack, E., Callegari, C., Matijasevic, M. (eds.) Data Traffic Monitoring and Analysis. LNCS, vol. 7754, pp. 264–301. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36784-7_11
23. Müller, C., Lederer, S., Timmerer, C.: An evaluation of dynamic adaptive streaming over HTTP in vehicular environments. In: Proceedings of the 4th Workshop on Mobile Video, MoVid 2012, pp. 37–42. ACM, New York (2012)