Received xx xxxx 2018; | Revised xx xxxx 2018; | Accepted xx xxxx 2018

DOI: xxx/xxxx

ARTICLE TYPE

Impact of VNF Placements on QoE Monitoring in the Cloud

Lam Dinh-Xuan* | Christian Popp | Valentin Burger | Florian Wamser | Tobias Hoßfeld

[1]Institute of Computer Science, University of Würzburg, Würzburg, Germany

Correspondence
*Lam Dinh-Xuan
Email: lam.dinh-xuan@informatik.uni-wuerzburg.de

Present Address
Chair of Communication Networks,
Am Hubland, 97074, Würzburg, Germany

Summary

The fast growth of video streaming is responsible for a huge amount of traffic over the past few years. Due to the variety of video content on the Internet, a potential market is emerging for video providers. However, a rapid increase in video traffic and users poses challenges for network operators maintaining user expectation. To address this problem, network operators need a mechanism to monitor the Quality of Experience (QoE) perceived by the user. This allows reacting to quality degradation to improve the service. With the paradigm of Network Function Virtualization, network operators are able to deploy such a video monitoring function in the cloud. In this work, we investigate the feasibility of deploying a Virtual Network Function (VNF) for video buffer and QoE monitoring on the Amazon Web Service cloud.
To this end, we implement a VNF to analyze video flows in the network by using deep packet inspection. We investigate the influence of different points of presence (PoP) in the cloud and mobile network on the performance of the VNF for monitoring video buffer and QoE. Our findings show that the accuracy of the QoE monitoring decreases with the distance of the PoP to the client. This is not only due to the delay and bottleneck between the monitoring point and the client but also due to the client mobile access network.

KEYWORDS:
NFV, VNF, QoE Monitoring, AWS Cloud, Network Measurement

1 | INTRODUCTION

The rapid growth of video streaming offers video providers a potential increase in revenue, but also presents challenges for network operators. In 2021, video streaming is predicted to account for about 82 % of total Internet traffic [1]. In the mobile context, mobile video traffic accounts for more than 78 % of all data traffic. In the fixed Internet, the trend is significant since video streaming is one of the most important and most data-intensive services. As a result, video streaming is one of the key services for network operators and requires special consideration, also with regard to the user and his experienced quality. The main goal for the network operator is the smart management of the mass of video data traffic and the satisfaction of the user.

In order to assess the quality of video streaming on the end user side, a video provider can utilize feedback given by the player. A network operator however needs his own monitoring mechanism to rate the actual video quality at the end user device. The question is how to monitor the perceived quality and how to do this in a scalable way for the mass of users. One possible solution to the scalability problem might be the instantiation of the monitoring within a Virtual Network Function (VNF) in the cloud. With this approach it is possible to provide the monitoring on demand and in a reactive way by instantiating the function on a data center server. In addition to simple instantiation, efficient use in the cloud brings improved scalability and cost savings. This approach is also a step towards unifying Network Function Virtualization (NFV)[2] and Cloud computing technologies, which is promising to be even more agile and flexible.

However, despite of its advantages, this approach also faces several problems. Since the main idea of this approach is to monitor the video quality using a VNF placed in the cloud, the first question is how to deploy the function on an existing cloud infrastructure? Secondly, since the VNF for video monitoring might be placed at different Points of Presence (PoP) in the cloud, the video quality measured at these locations may result in different degrees of accuracy. It is due to the fact that the network QoS may vary due to distance delay and possibly network congestion among others. Moreover, a high mobility client access network may also influence the accuracy of video quality estimation. The questions is how the location of different PoPs and mobility of clients influences the accuracy of the video quality estimation? Lastly, to be aware of the user satisfaction with the video service, Quality of Experience (QoE)[3] for video streaming must be measured accurately. Since the QoE is estimated mainly by considering stalling frequency and length, these key influence factors might be overestimated due to different placements of the VNF. The questions is how the placement of the VNF affects the estimation of key influence factors on video QoE?

To understand these problems, we investigate in this article the feasibility of placing a video buffer monitoring function in the cloud. First of all, we propose an architectural design of the deployment of VNF monitoring in the cloud. Then, we describe the monitoring process in the form of a VNF. We provide results that show the feasibility of monitoring application layer parameters within video traffic with Deep Packet Inspection (DPI). The monitored parameters have a high correlation with the user-perceived QoE. We compare the performance of VNF monitoring at different PoPs at the Amazon Web Services (AWS) cloud with regard to the actual quality obtained from the client device. Next, we add an investigation where the user is streaming a video in a high mobility scenario within a mobile network.

The contribution of this article is threefold. First, we propose a cloud-based NFV architecture on the example of deploying the QoE monitoring VNF on top of the AWS cloud. Second, we study the influence of different placements and a high mobility access network on the accuracy of the VNF for monitoring the video buffer and the QoE. In this case, we deploy the VNF at two regions of AWS cloud and the high mobility access network is emulated in a testbed. We also investigate technically the impact of packet delay on the accuracy of the estimation. Lastly, we validate our conclusions with experiments in a real scenario with a typical moving user on a vehicle. All in all, this article may help network operators to learn the pitfalls and drawbacks of a cloud-based monitoring approach.

The rest of the article is structured as follows. After the introduction, in Section 2, background of the study and related work are presented. Section 3 introduces our architectural design for VNF monitoring in the cloud. Subsequently, Section **??** describes the research methodology and video estimation algorithm. We present different measurement setups in Section 5. In Section 6, the results are presented. Finally, Section 7 concludes this work.

2 | BACKGROUND OF THE STUDY

In this section, we first present an overview of QoE assessment methods in Section 2.1. Thereafter, QoE monitoring approaches at different layers are introduced in Section 2.2 to Section 2.4. Subsequently, several works on QoE monitoring for HTTP Adaptive Video Streaming (HAS) and NFV cloud are introduced in Section 2.5 and Section 2.6, respectively.

2.1 | QoE Assessment Methodologies

As defined in[3], "Quality of Experience (QoE) is the degree of delight or annoyance of the user of an application or service". The definition is possibly suitable in the context of multimedia application or service. In this context, QoE is the level of user satisfaction or enjoyment with an application or a service. Being able to assess QoE helps network providers to react on the network quality degradations, since they may reduce the user satisfaction with the service. The following sub sections introduce subjective and objective assessment methods for QoE.

2.1.1 | Subjective QoE Assessment

Subjective QoE is the degree of expectation and/or satisfaction with a service experienced by human[4,5,6]. In HAS, subjective QoE is the feeling or perception of a viewer (or a *subject*) with the quality of image, stalling frequency, or stalling length. Subjective QoE assessments are psychophysical experiments where participants are asked to designate their opinion on the quality of a specific application or service (e.g., a video) through a given set of stimuli. The most popular method to quantify subjective

QoE is to use Mean Opinion Score (MOS) defined in[7]. The MOS can take the following values: (1) bad; (2) poor; (3) fair; (4) good; (5) excellent, which represent the corresponding degree of user satisfaction with the service. Subjective QoE assessment is considered to have high reliability results, since the tests are implemented in real scenarios with recruited participants. This method, however is expensive and time consuming. In addition, the set of stimuli must be well-defined and test sessions also require organization efforts that increases the cost of tests. In[8], Hossfeld et al. introduce an alternative to subjective testing, called QoE crowdtesting. This method leverages the advantage of crowdsourcing[9] by submitting assessment tasks to a global worker pool through a web-based aggregator platform such as Microworkers[1] or Amazon Mechanical Turk[2]. The forefront advantage of this method is reducing cost. This is achieved from a virtual laboratory with a low capital expenditure for a single test and limited number of participants. Thanks to a large diversity and high availability of participants, a test can be done or repeated quickly in one or different aggregator platforms. However, one of the major disadvantages of crowdtesting is unreliability of the user rating. This is due to the fact, that cheat users may only try to maximize their payments with minimum effort. Additionally, the test conditions and environments of the workers are unknown in most of the cases. Thus, the impact of test conditions on the result is different from worker to worker.

2.1.2 | Objective QoE Assessment

Objective QoE assessing refers to an attempt to predict the user behavior based on analytical or statistical models[10]. Similar to traditional subjective QoE assessment, this method also outputs the quantitative result reflecting the user expectation and satisfaction with an application or a service under test. The derived QoE model is especially useful for service quality monitoring in the network where providers can easily estimate the QoE perceived by the end user from validated input parameters (e.g.,[11,12,13,14,15]). However, in contrast to the ease of use, objective QoE can only provide estimated results with a specific correlation with perceptual quality measured by subjective assessment.

In[16], the authors categorize objective assessment methodologies for IPTV into five types of models. These models are media-layer, parametric packet-layer, parametric planning, bit stream layer models, and hybrid. Each type of model can exploit the input parameters for QoE models at different network layers. For instance, the media-layer model uses video signals to estimate QoE, while the bit stream layer model captures the packet header and payload to derive input parameters for a QoE model. In this work, we use the combination of the parametric packet-layer model and the bit stream layer model in the form of a VNF to monitor video buffer and QoE for HAS.

2.2 | QoE Monitoring at User Layer

Monitoring QoE for a service at the user layer refers to statistical information collected actively or passively from the user while he or she is using the service.

2.2.1 | Active Monitoring

Active QoE monitoring at the user layer can either be done by dedicated subjective user studies or by customer feedback. QoE monitoring which utilizes subjective assessment methods can be implemented in laboratories, field studies, or using crowdsourcing as described above. The more popular active QoE monitoring method is the integration of customer feedback dialog into the service. Such quality feedback integrator is widely used in speech and video services where the user is asked to rate the quality of a conversation after they hang up the phone (e.g., a Skype call). This method is considered to be lower cost and easy to deploy for different applications and services. However, the feedback or rating might be less accurate, since the user may be annoyed when receiving the feedback dialog at each time he uses the service. In addition, the reasons for a good or bad rating is hard to be traced. Nevertheless, the combination of the active monitoring with other methods may increase the accuracy of QoE monitoring.

2.2.2 | Passive Monitoring

In contrast to active monitoring, passive QoE monitoring at the user layer does not interact directly with the customers, but probes their behavior passively through different measurement techniques. For example, providers can investigate the ratio of

[1] https://microworkers.com
[2] https://www.mturk.com

user engagement, the volume of sale products, or the number of hotline complaint calls, etc., to estimate the degree of the user satisfaction with the service. This monitoring method is completely transparent with the customer activities. Moreover, some existing information of the customers can also be obtained from business administration department which can be analyzed to estimate the customer behavior with offered services. However, QoE estimated by this method has limited accuracy since the user engagement or sales volume are not only influenced by the customers but also by other objective causes. The reasons for a high or low QoE is hard to be traced as well.

2.3 | QoE Monitoring at Application Layer

QoE monitoring at application layer can be done either inside a service or at the end user device. *In-service QoE monitoring* is ordinarily implemented by service providers who can monitor QoE influence factors within the service. These influence factors are served as input parameters for a QoE model. Afterward, QoE score is signaled back to the provider to adapt the service in order to meet the user expectation. *End-device QoE monitoring* relies on a monitoring function installed at the end user device. This additional software is used to collect the performance indicators of a service or application which can be translated into QoE scores. The monitored information might be useful for the user or can be signaled to network operator. Based on this, traffic management can be applied to ensure a high QoE level perceived by the users.

2.4 | QoE Monitoring at Network Layer

At the network layer, QoE monitoring can be categorized into active and passive methods, in which an active monitoring mechanism uses probe nodes to perform measurement. Whereas, in passive monitoring, network traffic passing through a monitoring point is captured and analyzed.

2.4.1 | Active Monitoring

To monitor QoE actively for a specific service or application at network layer, one or multiple probe nodes are distributed across end points in the network. These nodes send extra data traffic to servers and measure the service quality based on pre-defined QoE metrics. The QoE scores are then signaled to a collector server or directly to providers to adjust services in order to improve user experience. Active QoE monitoring mechanisms depend on the services. In[17,18], the authors present a method for monitoring the quality of video streaming using a Pseudo Subjective Quality Assessment (PSQA) technology. This active monitoring method uses probe nodes distributed in the network. At first, the authors subjectively evaluate several distorted samples that caused by different parameters (i.e., I, B, P video frames loss rate). Then, the authors use the results of this evaluation to train a learning tool (i.e., Random Neural Networks). This training process allows to map the values of the set of parameters above into perceived quality. After that, several experiments were done in a testbed where simulated probe nodes periodically send statistical information about frame loss rate and mean loss burst size to a data collector server. Based on these collected parameters, the QoE for a video delivery network is estimated at run time using trained data that obtained from the training process. The accuracy of the platform is determined by the loss rate of video frames and mean size of loss bursts. This method is limited in accuracy and efficiency. This depends on the fact, that the number of probe nodes is finite and that these nodes produce more overhead into the network or might change the outgoing data traffic for probing purposes.

2.4.2 | Passive Monitoring

Passive QoE monitoring approach can use DPI technique to capture data traffic passing through a monitoring point in the network. Payload data is then parsed for service performance indicators. In HAS, payload data contains video request time, segment download time, Media Presentation Description (MPD), which can be used to estimate video buffer and other video quality influence factors. Passive QoE monitoring using DPI technique does not change the data traffic or require any end device. This method is promising to be highly accurate, since QoE is measured directly from service key influence factors extracted from payload data. A softwarized monitoring function can be deployed at any PoP in the network. However, to be able to measure QoE, one must thoroughly understand the service and its communication in the network. In addition, it is more difficult to monitor QoE for encrypted data traffic. In this situation, a *service flow classification* method can be used. This method can leverage machine learning to classify traffic flows of a service. By doing this, network layer parameters such as packet size, packet inter-arrival time, or throughput are monitored. These network layer parameters can be used to directly estimate QoE or to estimate application layer parameters. Thereafter, a QoE model is used to map application layer parameters to QoE. An example of *service flow classification* is presented in[19].

2.5 | QoE Monitoring for HTTP Adaptive Video Streaming

To assess the quality of video at the network layer, either active and passive monitoring methods can be used. As described in the previous subsection, an active video monitoring can use the PSQA technique that is presented in[17,18].

Regarding the passive monitoring method, the authors in[20,21] introduce QoE monitoring for YouTube video using end device application, namely YoMoApp. This application layer monitoring method is promising to have high accuracy in QoE measuring, since it can extract video quality parameters directly from video player on the user device. This method however only work at the client side, while our approach is deployed in the network. Concerning video monitoring based on packet analyzing, our study has similar method to the works[22,23,24,25] to some extent, where the authors analyze the YouTube video flows in the network using DPI to estimate the QoE based on extracted packet traces. However, the authors do not consider the influence of VNF placement in the cloud on the accuracy of QoE estimates.

In[26,27,28,29], the authors introduce methods and present the results of evaluating adaptive video streaming in mobile environments. The authors observe that most video streams in mobility scenarios will gain a smooth playback if the adaptation process is improved. Similar to their study regarding to the experiments of moving clients, our study however focuses more on evaluating the accuracy of VNF monitoring deployed in an AWS cloud.

As described in the next sections, a part of this article has been published in[30] (in particular, the algorithm presented in Section 4). However, in previous work, we deploy the VNF monitoring in a local testbed and the results primarily show that our VNF is able to estimate video buffer at the client video player. The additional value of this article is that we show the feasibility of deploying the VNF monitoring in the cloud and the evaluation of influence of high mobility environment on its accuracy.

2.6 | NFV Cloud

In the NFV paradigm, network functions are virtualized and deployed on top of the virtualization layer which can also be provided by an existing cloud management systems like AWS. In[31], Oechsner et al. discuss about possible deployments of the NFV paradigm in a cloud architecture. The authors present a mechanism to organize the infrastructure based on an adapted zone concept and prove this mechanism by providing an example of a VNF placement on OpenStack[3]. In[32], Carella et al. propose a possible deployment of IP Multimedia Subsystem (IMS) software on top of the OpenStack cloud. The idea of leveraging a telco cloud environment to manage service functions (SFs) is presented in[33]. The authors introduce the Cloud4NFV platform that is built on cloud, Software Defined Networking (SDN), and WAN technologies to manage SFs as a service. This platform is promising to improve the management of SFs in the cloud. However, the use of open source platforms like OpenStack or OpenDaylight[4] for the virtual infrastructure management plane will need more assessment to ensure the security and feasibility in practice.

In[34], Yu et al. introduce the network function-enabled cloud computing (NeFuCloud). This platform was proposed with the idea of separating the control plane from the data plane underlying cloud infrastructure which is the principle of SDN. The authors believe that NeFuCloud, with separated control plane will gain better performance and flexible management. However, the control plane might be centralized or distributed across the network. A centralized control plane may lead to a bottleneck when the network is scaled out and a distributed control plane increases the control overhead. In our concept, we propose to build NFV on top of an existing cloud platform (i.e., AWS) that provides flexible management and implementation.

3 | ARCHITECTURE FOR QOE MONITORING USING VNF

In this section, we propose an architecture of QoE monitoring that utilizes an existing cloud infrastructure to deploy VNFs on the top. This architecture is used to implement our QoE monitoring measurement described in Section 5. Figure 1 gives an overview of the QoE monitoring VNF architecture in the cloud.

[3] http://www.openstack.org/

[4] http://www.opendaylight.org/

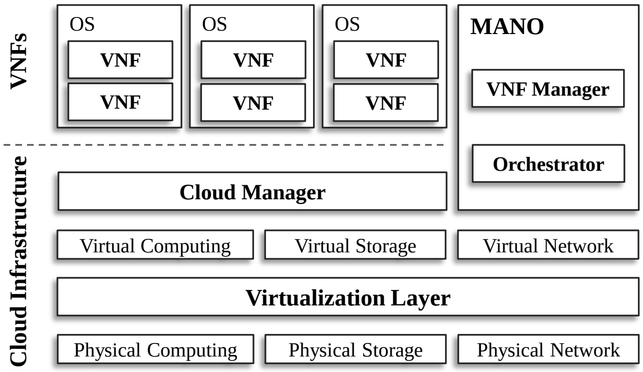


FIGURE 1 Overview of QoE Monitoring VNF Architecture

The design can be divided into two main blocks, the Cloud infrastructure plane and the NFV management plane. This architecture is aligned with the ETSI-NFV architecture framework[35]. However, VNFs, Management and Orchestration (MANO) are built on top of an existing cloud infrastructure.

3.1 | NFV Management Plane

The NFV management plane provides the environment and management for deploying VNFs. Aligning with the ETSI architecture, it consists of MANO and VNFs.

VNF Instance: In the NFV management plane, one or more VNFs are installed in a virtual machine or a container (e.g., Docker[36]) and are managed by the MANO. Each block that consists of a virtual machine with operating system (OS) built-in and one or more VNFs installed, is called a *VNF Instance*. An instance is allocated with sufficient resources at run time and can be dynamically scaled-up or out by an orchestrator. In our measurement setup, the VNF for video quality monitoring is installed on a virtual machine. This VNF instance is initiated at different working regions of the AWS cloud.

MANO: In this architecture, the MANO is more lightweight compared to the ETSI-MANO due to the handover of infrastructure management to a cloud manager. It consists of VNF manager and orchestrator. Aligning with the ETSI standardization, the *VNF Manager* is in charge of VNFs life cycle management. The *Orchestrator* is responsible for automated provisioning necessary resources and network for the VNFs. The MANO can also work with the Cloud Manager via a REST API to initiate a new instance of VNFs from templates. By doing this, the VNFs can be scaled out or migrated across different working regions. In this work, we only present functional overview of the architecture and leave implementation for future work.

3.2 | Cloud Infrastructure Plane

The underlying Cloud infrastructure plane provides all necessary hardware and software resources to build a virtual environment where VNFs are deployed.

Virtualization Layer: Virtual resources consist of virtual computing, storage, and network which are created by a virtualization layer. This layer ensures the isolation of virtual machines from the underlying physical resources. Thus, many virtual machines or containers can share limited physical resources using a resources sharing scheme.

Cloud Manager: This module is built to manage the cloud infrastructure and allocate resources for VNF instances. For example, a new VNF instance will be allocated with initial virtual resources like computation, storage, memory, and networking. In addition, a specific working geographical region and other security policies of the instance are also configured before launching. Then, the instance is initiated to host a virtual machine or a container. The instance can be automatically scaled-up with additional resources without any interruption. It can be duplicated and migrated to another working region as well.

In this work, we choose Amazon Web Service (AWS) as the Cloud infrastructure for QoE monitoring VNF, since AWS provides an IaaS platform with multiple management options. AWS is a collection of different online services and a subsidiary of Amazon.com, Inc[37]. The most popular services are Elastic Compute Cloud (EC2)[5] and Simple Storage Service (S3)[6]. S3 is mainly used to store data in AWS datacenters. While, EC2 provides computing nodes on demand, where user can launch instances with different hardware configurations depending on his requirements. Instances are multiple virtualized containers that can be installed on the same physical machine. In addition, AWS provides a Virtual Private Cloud with security groups for each user, this ensure the isolation of the containers from the other users.

When launching an instance, the user can select a desired operating system and a region where the instance is operating. AWS provides EC2 services located in 18 regions that are totally independent from each other. In each region, there are availability zones that are also isolated and connected through low-latency connections. In this work, we choose an EC2 service to deploy our VNF for video monitoring at two different regions, namely *eu-central-1b* that is located in Frankfurt, Germany, central Europe and *us-west-2b* that is located in Oregon, USA.

4 | VIDEO ESTIMATION ALGORITHM

The video estimation algorithm is specifically designed for the evaluation of performance and accuracy of video streaming with unencrypted HAS traffic. Although there are other techniques available for encrypted traffic, we focus on the investigation of the accuracy of such an VNF-based monitoring and the side-effects of network QoS and VNF placement in the network.

In[38], Seufert et al. describe the QoE influencing factors of HAS, which are initial delay, stalling, quality adaptation among the others. To achieve the goal of monitoring QoE for the video, we design an algorithm to estimate these parameters based on the extracted packets of the video flows. To feed the algorithm, the sniffing task of the VNF outputs all necessary information such as packet receiving time, source and destination IP addresses, source and destination ports, TCP acknowledgment numbers, TCP sequence numbers as well as the length of packet and HTTP payload. Figure 2 schematically depicts the algorithm of estimating video buffer based on analyzing video flows.

The top of the figure shows an example of transferring four video segments in sequence. The video flows are detected by matching pre-defined keywords contained in HTTP payload. After a video request from the client has been sent, the streaming server provides a Media Presentation Description (MPD) which contains video segment name and the respective durations. This information is stored for further analysis. Based on the MPD, the client seamlessly requests segments in sequence. The downstream packets that compose one segment are grouped and analyzed based on their common TCP acknowledgment number. Thereby, the timestamps of fully downloaded segments are recorded.

From the left side of the figure, t^r is the time when the client requests the video, t_i^d is the time when segment i is fully downloaded, which is represented by each vertical arrow. The solid rectangles with faded colors depict the corresponding segment length `seg_len`(i) extracted from the MPD. The solid zig-zag line represents the amount of estimated video buffer `eVideoBuff` over playback time. It drops steadily over video playback and is filled up when a new segment is fully downloaded.

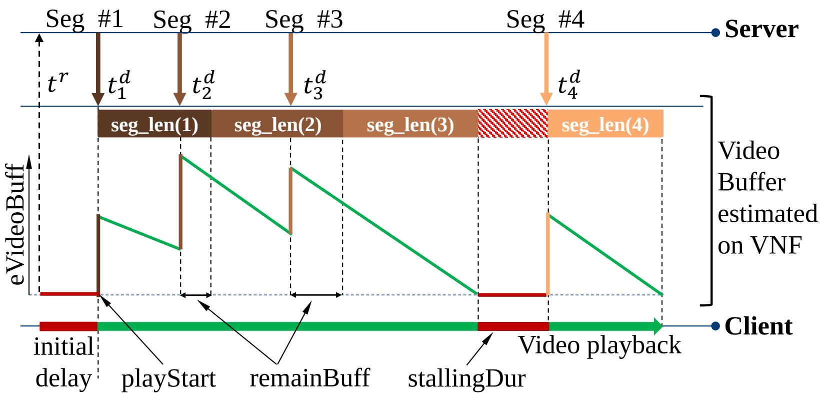


FIGURE 2 Graphical View of Video Estimation Algorithm

[5] https://aws.amazon.com/en/ec2/
[6] https://aws.amazon.com/en/s3/

The `initial_delay` is the amount of time measured from the client requesting the video at the time t^r until the video starts to play out at the time `playStart`, while `playStart` $\approx t_1^d$. In fact, through measurements on the client, we observe that the client starts the playback almost instantaneous after the first segment is fully downloaded. The `remainBuff` is the amount of video time remaining in the buffer before filling up with the next segment. If the next segment arrives later than expected, then an interruption of the video playback occurs. This is called a stalling event. In[15], the authors state that the longer stalling occurs, the lower QoE perceived by the user. In Algorithm 1, the `stallingDur` is the length of stalling event which serves as an input parameter for QoE model. The simplified algorithm is presented in the following.

Algorithm 1 Estimating Video Buffer Algorithm

```
Define samplingRate and the time when video starts the playback and sampling video buffer
timestamps = playStart = $t_1^d$
Available playback time is built from the first segment
avlPlayBackTime = segmentLength(1)
for Next segment do
    Amount of video has been played out when segment i arrives
    playOut = $t_i^d$ − playStart − stallingTotal
    Remaining video buffer before next segment arrives
    remainBuff = avlPlayBackTime − playOut
    if remainBuff <= 0 then
        Record duration of one stalling event
        stallingDur($i$) = $t_i^d - t_{s(i-1)}^{end}$
        stallingTotal + = stallingDur($i$)
        Record stalling states with pre-defined sampling rate
    else
        Amount of video has been played out when segment $(i-1)$ arrives
        ite = hasBeenPlayedOut
        Estimate video buffer and timestamps cumulatively from the time $t_{i-1}^d$
        while ite <= playOut do
            Timestamps of sampling video buffer
            timestamps + = samplingRate
            Video buffer state is sampled every samplingRate
            eVideoBuff = avlPlayBackTime − ite
            ite + = samplingRate
            Save one record [timestamps, eVideoBuff]
        end while
    end if
    hasBeenPlayedOut = playOut
    Update available playback time
    avlPlayBackTime + = segmentLength(i)
    $t_{s(i)}^{end} = t_i^d$ + segmentLength(i)
end for
```

In the algorithm, the `samplingRate` variable is defined at first. Sampling rate is the amount of time in millisecond between two sampled video buffer values. Since our algorithm estimates the video buffer based on segment arrival time, to avoid missing a buffer value, the sampling rate must be smaller than the download time of a segment. Through a local experiment and based on analysis at packet level presented in Section 6.1, we observe that the average download time of a segment is about 350 ms. Based on this analysis, we choose a sampling rate of 100 ms at the VNF and the client to ensure all buffer values are captured. Next, the algorithm estimates `playStart` with the time when the first segment arrives. The `avlPlayBackTime` is the available playback time which is built up by the duration of segments obtained from MPD, named `segmentLength`. This amount of time is cumulatively summed by every new segment that is fully downloaded. To estimate the video buffer, the amount of video that has been played out must be calculated first. `playOut` is calculated by the difference of the time between a new segment arrived and the `playStart` subtracting with `stallingTotal`. The video buffer `remainBuff` is calculated by the difference of `avlPlayBackTime` and the amount of video which has been played out. This parameter plays a role as a trigger to detect whether a stalling event occurs. If the video buffer is depleted, we record one stalling event and its length. Then, we sample the stalling state with the corresponding variable `timestamps`. Conversely, if there remains video in the buffer, we record the video buffer values and its timestamps to the variable `eVideoBuff` and the `timestamps`, respectively. The variable `timestamps` is increased until it reaches the last millisecond of the video.

5 | MEASUREMENT SETUP

In this section, we first present our method to conduct the study. Subsequently, we describe the measurement setups of testbed and real scenarios at both local testbed and in the AWS cloud. The installed hardware, network configuration, and method of implementation are described.

5.1 | Research Methodology

Our study aims to monitor the quality of video streaming to a client device in the network. To this end, we first choose AWS EC2 as cloud environment where we place the VNF at a PoP. The video traffic from the streaming server is routed to pass this PoP and is captured by our VNF. The VNF is a plain software which uses a Python library, namely Scapy[7]. This open source software provides real time packet sniffing and decoding. In fact, we can parse the video flows on the fly. In our measurements however, we do the analyzing task afterwards to avoid the interference with the performance of the sniffing task. To analyze the video flows, we design an algorithm which eventually provides us with an estimated video buffer based on download timestamps of video segments. To validate the estimated values, we simultaneously sample the video buffer at the client browser by using a Javascript-based web API. The discrepancy between the estimated and actual video buffer shows the level of accuracy of the VNF monitoring.

To evaluate the influence of different placements on the accuracy of the VNF in high mobility user access networks, we deploy our function at two AWS instances. One located in the central of Europe which is near the client, the other located at the West of the USA which is far away from the client. Both of these locations are close to the streaming server or its Content Delivery Network (CDN) server. For the high mobility user access network, we divide experiments into two scenarios, *testbed* and *real scenarios*. In the *testbed scenario*, we emulate a wireless environment based on pre-defined LTE bandwidth traces measured in real life. With the network emulation, we can evaluate the behavior of the VNF running in the cloud in multiple experiments with low cost. In this scenario, we provide the results of VNF accuracy in estimating video buffer and QoE for video streaming using a reference QoE model. In the *real scenario*, we use a vehicle to drive the client machine around a city and in country side regions. The results provided in this scenario are used for the validation of our simulation.

5.2 | AWS Cloud Infrastructure Setup

As mentioned in Section 5.1, we divide experiments with the VNF deployed in the AWS cloud into two scenarios. In the *testbed scenario*, the user access network is emulated by a network emulator. Meanwhile, in the *real scenario*, the user access network is provided by a German telecommunication operator. Nevertheless, in both testbed and real scenarios, the VNF is installed in an AWS cloud instance. The instance acts as a PoP for the VNF where a proxy program is used to route video traffic passing through the PoP to the client. During the measurement, the location of the client is varied. In the testbed scenario, the client is placed in the local testbed, while during the measurements of real scenario, the client is placed in a vehicle moving around the city and country side. Thereby, this scenario is also used to investigate the influence of the mobile network on the accuracy of the VNF. In both scenarios, the VNF is installed in an AWS *t2.micro* server with Ubuntu Server 16.04 LTS as operating system. Each instance has one virtual CPU clocked up to 3.3 GHz, 1GB of RAM and low-to-medium network performance. These instances are located in different regions as mentioned in Section 3.2. The aim of this location distribution is to evaluate the influence of different VNF placements on its accuracy. Table 1 shows the average round trip time (RTT) and the network throughput in between the AWS regions and the client in the testbed. The mean value is calculated over 100 samples.

[7] http://www.secdev.org/projects/scapy/

TABLE 1 Mean RTT and Throughput of AWS Cloud Regions

AWS Region	RTT	STD	Throughput	STD
eu-central-1b	6 ms	0.50 ms	258 Mbit/s	14.00 Mbit/s
us-west-2b	170 ms	0.11 ms	109 Mbit/s	11.60 Mbit/s

Table 1 shows that the mean RTT between the client and the *eu-central-1b* region is only 6 ms and it is much smaller than the *us-west-2b* region. The mean throughput in *eu-central-1b* region also doubles the *us-west-2b* region. This is due to a longer distance between the client and the *us-west-2b* region that may have undesired network congestion compared to a shorter distance between the *eu-central-1b* region and the client. Regarding the streaming server, all measurements are conducted with the use of the video streaming platform *metacafe.com*[8]. This platform has the benefit that it provides unencrypted traffic for HAS which is easier to process. An analysis of the MPD of a video taken on a street[9] shows that the video qualities are available in up to three levels, which are 240p, 360p and 720p, representing the resolutions of 426x240, 640x360, and 1280x720 pixels. Table 2 shows an example of video information extracted from MPD. The table shows that if the bandwidth between the client and the server is constant and larger than the specified value (e.g., 0.46 Mbit/s), the client should be able to watch the video seamlessly without any stalling events[39,40].

TABLE 2 Video Stream Information from MPD File

Quality	Bandwidth	Resolution
240p	0.46 Mbit/s	428x240
360p	0.65 Mbit/s	570x320
720p	1.14 Mbit/s	1280x720

5.3 | Measurement Setup for Testbed Scenario

The testbed scenario measurements are conducted in a testbed located at the University of Würzburg. Figure 3 shows the topology of the testbed. In Figure 3 , the *Client* is a Fujitsu PC with 1 Gbit/s NICs. Ubuntu Desktop 14.04 LTS is used as operating system. The *NetEm* is a SUN FIRE X4150 server with Ubuntu Server 16.04 LTS as operating system. The testbed is connected to the Internet via the German research network backbone for universities (Deutsches Forschungsnetz-DFN). For the sake of traffic tracing, we deploy the VNF monitoring at both the client and the AWS cloud. This allows us to compare the

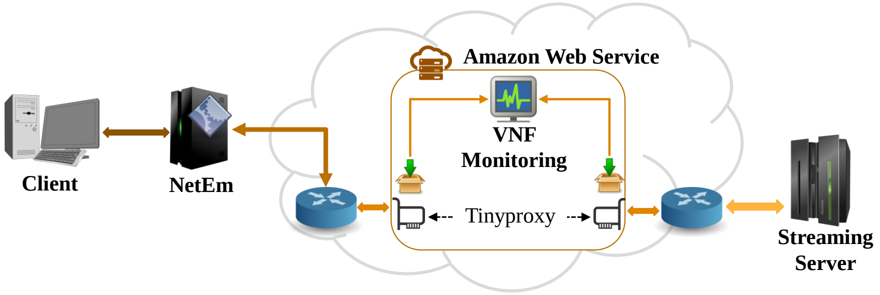


FIGURE 3 Overview of Testbed Scenario Topology

[8] http://www.metacafe.com
[9] http://www.metacafe.com/watch/11419883/homeless-video-mp4

QoE estimation in the cloud and at the client device as well as analyze the traffic at both end points. In addition, the *Tinyproxy*[10] software is used to route video traffic passing through the PoP from the streaming server to the client. To secure the reliability in timestamps calculation, we synchronize the system clock of all testbed devices with the same NTP server.

In this scenario, the *Client* is fixed in the testbed and measurements are implemented during daytime. We simulate homogeneously a single user watching a video in every experiment replication. The interval between two replications is 60 s. To this end, we use a Python library, called Selenium Webdriver[11] for Google Chrome to automatically browse the video. In the setup, we emulate a high mobility user access network with a network emulator (NetEm)[41]. This Linux-based software can adjust different network configurations to evaluate the impact of network QoS on services or applications in general. To start the measurement, at the *Client* we use a Python script to remotely activate the VNF at the AWS instance through SSH protocol. Then, the *Client* automatically browses the video after the *NetEm* is configured with pre-defined bandwidth. This ensures the network is shaped before the client starts the video playback. The bandwidth traces were measured from a real mobile scenario where it was sampled every 1 s. The duration of the bandwidth traces is about 500 s. In the measurements, we configure the *NetEm* continuously to ensure different network condition in each replication. Video stalling events therefore only occur sporadically over all replications. The bandwidth measurements were performed in LTE networks along several routes in and around the city of Ghent, Belgium, during the period of Dec. 2015 to Feb. 2016 (more details can be found in[42] and the below website [12]). This setup allows us to produce enough experiments to evaluate the influence of wireless network on the accuracy of the cloud-based VNF as well as QoE estimation for video streaming.

5.4 | Measurement Setup for Real Scenario

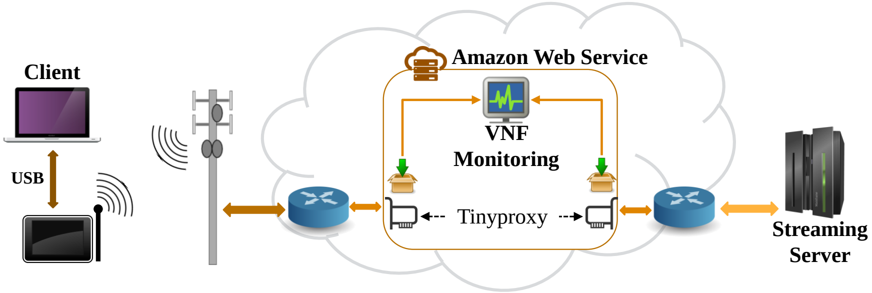


FIGURE 4 Overview of Mobility Client Scenario Topology

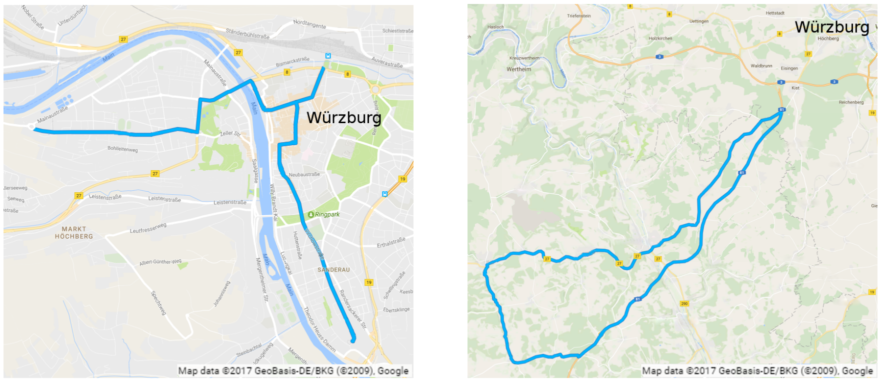
In the real scenario, we use the mobile network for Internet access. Figure 4 shows the network topology of this scenario. The *Client* is a notebook with Ubuntu 14.04 LTS as underlying operating system. The network connection is provided by a tablet via a USB hub. The tablet connects to the Internet using its built-in LTE modem which supports up to 150 Mbit/s downstream and 50 Mbit/s upstream. The network signal is provided by a major German telecommunication operator with a maximum downstream bandwidth up to 300 Mbit/s. The measurements in this scenario are implemented in two different configurations of location. Figure 5 shows the routes of the vehicle in both city and country side regions.

In the first configuration, as shown in Figure 5 a, we implemented the measurements while taking the public transport system (electric tram) as our vehicle in the city center of Würzburg. In the second configuration, as shown in Figure 5 b, we pre-defined a fixed route over country roads and the highway A81 and we used car as the vehicle.

6 | RESULTS

Based on the research method presented in Section 5.1 and measurement setup in Section 5, we have implemented experiments between January and February 2017 at both the University and the city of Würzburg. The video we present in the following

[10] https://tinyproxy.github.io/
[11] http://www.seleniumhq.org
[12] http://users.ugent.be/~jvdrhoof/dataset-4g


Würzburg
SANDERAU
Ringpark
MARKT HÖCHBERG
Map data ©2017 GeoBasis-DE/BKG (©2009), Google
Würzburg
Map data ©2017 GeoBasis-DE/BKG (©2009), Google

(**a**) City

(**b**) Countryside

FIGURE 5 Routes of Vehicle in Mobility Scenario

measurement results was taken by GoPro camera for a sport activity[13]. It has total length of 200 s. The evaluations consider the accuracy of the network function, where the accuracy is measured as the difference between estimated video buffer by the VNF and actual video buffer obtained by the client browser. To minimize the number of missed video buffer states, we use a sampling rate of 100 ms at the VNF and the client as described in Section 4. Since the video buffer is measured and estimated at different devices and locations, we choose video starting time calculated by the VNF as the time reference for all graphs that have video playback time x-axis.

6.1 | Video Quality Monitoring in the Testbed Scenario

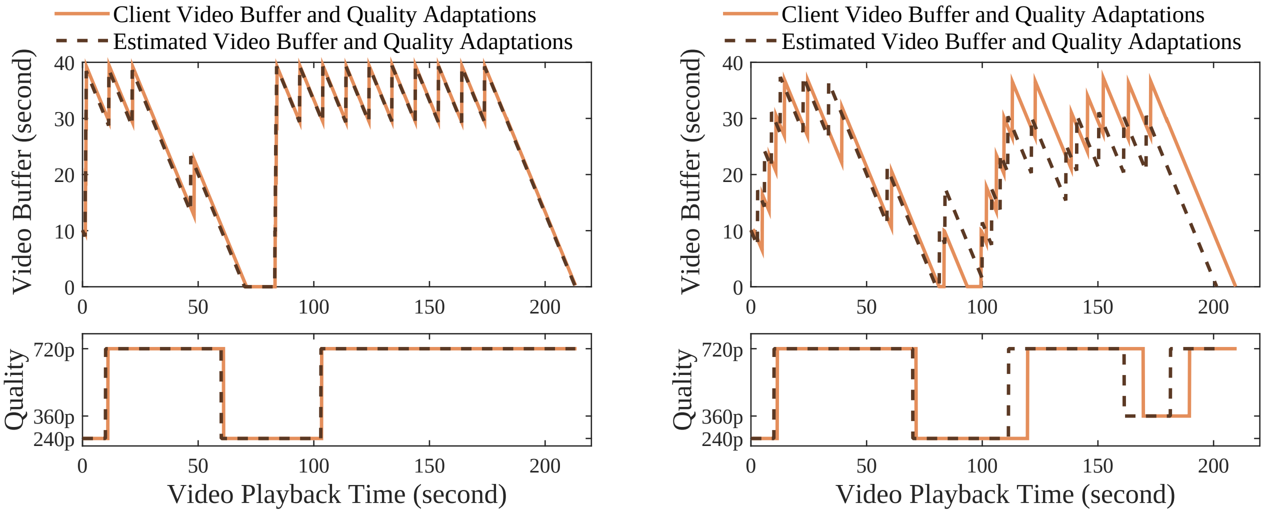
The measurement results in this section are used to evaluate the impact of VNF placement and high mobility environment on the accuracy of video buffer and QoE estimation. The client device is located in a testbed where the mobile network environment is emulated by a network emulator as described in Section 5.3. The monitoring VNF is placed at different regions of the AWS cloud as described in Section 3.2. These points of presences are so-called EU-PoP and US-PoP. For the sake of traffic analysis, we also deploy a similar VNF at the client machine to capture all incoming packets from the streaming server. This VNF is referred to as the *local VNF*. To increase statistical significance of the measurements, we produce 60 replications at each PoP.

6.1.1 | The Influence of VNF Placement on Video Buffer Estimation

To intuitively compare video buffer and stalling events estimated by the VNF and obtained from the client, we select a typical video experiment that consists of different video states (i.e., high buffer, stalling, quality adaptations). Figure 6 shows the behavior of the video buffer and quality in the testbed scenario. In all sub-figures, the x-axes show video playback time in seconds. In the upper parts of the sub-figures, the y-axes indicate the amount of video buffered in the memory. In the lower parts of the sub-figures, the y-axes show the video quality adaptations to different network conditions. The solid lines depict the behavior of video buffer and quality sampled by the client browser, while the dashed lines represent video buffer and its quality estimated by the VNF.

The results show that in both PoPs, there is a correlation between the quality adaptations and video buffer dynamics. The client requests a low quality video segment at first, due to the unawareness of network conditions. As shown in Figure 6 a, between the playback time of 71 s and 83 s, the video is stalling. This is a result of a connection loss. The video buffer is therefore depleted before new video segment arrives. The video quality also drops, since the client requests a smaller segment size. This typical behavior of HTTP adaptive video streaming is also described in[40,38]. Considering the accuracy of the video buffer estimation, the upper part of Figure 6 a shows a good fit of estimated and actual video buffer as well as detecting stalling events. The VNF estimates the video quality adaptations with high accuracy as shown in the lower part of the figure. In contrast to this, the VNF placed at the US-PoP estimates the video buffer with higher error as shown in Figure 6 b. In this case, there is a time shifting at the beginning of the video playback and after the stalling event. The video buffer estimation is fluctuated and less accurate.

[13] http://www.metacafe.com/watch/11419867/let-me-clear-my-summer/



(a) EU-PoP

(b) US-PoP

FIGURE 6 Video Buffer and Quality Monitoring at Different VNF Placements

6.1.2 | The Accuracy of Video Buffer Estimation

As described above, there is a time shifting in the estimated video buffer and quality adaptations compared to the ones obtained from the client. Indeed, to evaluate the discrepancy between the estimated and actual video buffer at both PoPs, we calculate the Root Mean Squared Error (RMSE) as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}, \tag{1}$$

where n is the number of video buffer samples, $\hat{y}_i$ and y_i are video buffer sampled by the VNF and client browser, respectively. Figure 7 depicts the cumulative distribution function (CDF) of RMSE between estimated and actual video buffer. The lines with different colors represent the CDF of RMSE at different PoPs.

Figure 7 shows that the estimated values at the US-PoP have errors larger than 3 s. The EU-PoP errors are smaller than 0.7 s in 90 % of the cases. Since the placement of the VNF at EU-PoP near the client, the difference in arrival time of video flows is negligible at both machines. In case of packet loss, the retransmitted packets are received almost instantaneously at both machines as well. Therefore, high accuracy estimations are obtained at the Eu-PoP which is geographically closer. Conversely, the longer distance in between the client and the VNF at the US-PoP may cause delay and possibly network congestion. This produces a time shifting in estimating the video buffer when the VNF is placed at this location. Note that, the client here is permanently located in Europe next to the EU-PoP and the CDN streaming server as well. Whereas, when the VNF is installed at the US-PoP, the request is forwarded to another streaming server in USA. As a consequence, packets are transfered in a longer distance that significantly increases the end to end latency.

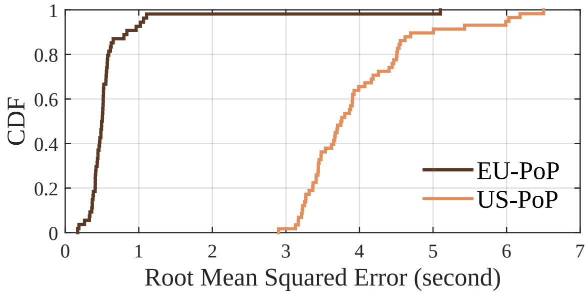


FIGURE 7 RMSE Between Estimated and Actual Video Buffer

6.1.3 | Packet Delay Analysis

In fact, the monitoring function may add a delay caused by the forwarding and processing of the proxy. This overhead is relatively small compared to the delay caused by the distance induced by the placement of the monitoring function, which is the scope of our study. Therefore, we do not consider the overhead of the video buffer estimation process in this work. To understand the influence of different VNF placements on its accuracy, we analyze the delay of video segment arrival through a scheme as depicted in Figure 8 .

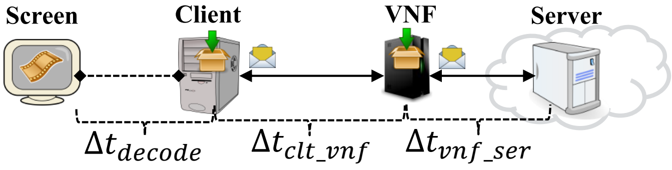


FIGURE 8 Overview of Packet Delay Scheme

Although packets traveling across the network are delayed by many factors like queuing delay or processing delay, we only calculate the download time and decoding time of each video segment. In addition, we analyze the download time of video segment as measured by the VNF. Analyzing the download time of individual packets is not straight forward. Because of the TCP congestion control algorithm[43], packets passing through the proxy at the AWS PoPs may be re-encapsulated to adapt the network condition in between the client and the AWS PoPs. As shown in Figure 8 , Δt_{decode} is the amount of time between a segment is fully captured at the client and it is filled up to the video buffer. The interval Δt_{clt_vnf} is the segment download time from the VNF to the client. Note that, as described above, we also deploy a similar local VNF at the client device for traffic analysis purpose. The duration Δt_{vnf_ser} is the segment download time from the streaming server to the VNF. This network delay, however, is not considered in this study. Figure 9 shows the average decoding and download time measured at different monitoring points.

In Figure 9 , the x-axis shows monitoring points at the EU-PoP and the US-PoP. The y-axis indicates the delay time. The bars with different colors depicts the mean of decoding and download time over 60 replications with a 95 % confidence interval. The result shows, that the decoding time is negligible in the measurements at both PoPs. However, the average download time of video segment measured at the US-PoP is much higher than at the EU-PoP. This shows, that a long distance between the client and the US-PoP with additional unstable user mobile access network are the main factors influencing the accuracy of the VNF for video buffer monitoring.

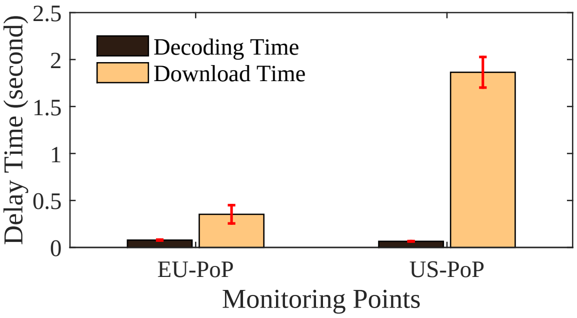


FIGURE 9 Average Decoding and Download Time at Different Monitoring Points

6.2 | The Influence of VNF Placement on QoE Estimation

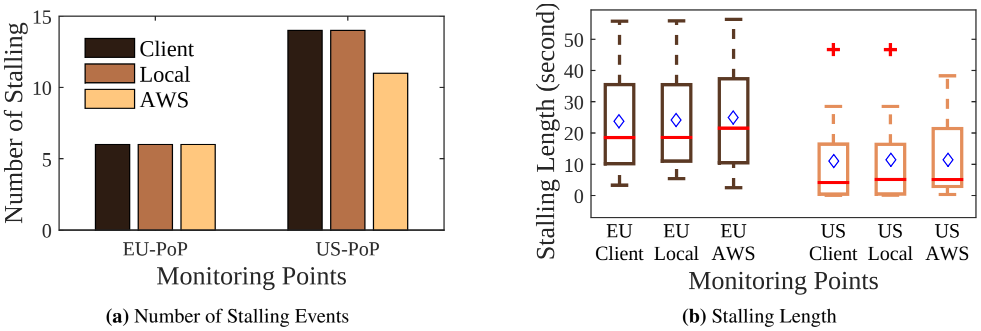
In[44], Hoßfeld et al. present a YouTube video study, in which they investigate the impact of network degradations on QoE for video streaming. The authors conclude that the combination of stalling frequency and length has exponential influence on the QoE perceived by the users. They propose a fitting function to estimate the QoE based on the duration and the number of stalling events as follows.

$$f_{MOS}(L, N) = 3.50 \cdot e^{-(0.15L+0.19)N} + 1.50, \tag{2}$$

where f_{MOS} is the function of the Mean Opinion Score (MOS) given by the average number of stalling events N and stalling length L. The equation shows that the QoE for video streaming only considers the number of stalling events and length as main influence factors. In the following, we present experimental results that show the behavior of QoE estimation under the influence of VNF placement and mobility environment.

6.2.1 | Stalling Frequency and Length Analysis

Figure 10 shows an overview of the video stalling behavior over the course of measurements at the EU- and US-PoP.



(a) Number of Stalling Events

(b) Stalling Length

FIGURE 10 Number of Stalling and Length at Different Monitoring Points

In Figure 10 , the x-axes show the monitoring points, which are at the client, at the local VNF and at the VNF deployed within the AWS cloud instance. The y-axes indicate the total number of stalling events in Figure 10 a and the stalling length in Figure 10 b, respectively. In Figure 10 b, each box shows the maximum, median and minimum durations of stalling events at a measurement point. The bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers. The outliers are plotted individually using the '+' symbol. The diamond symbol in the middle of the box shows its mean value.

The result shows that in the measurements at the EU-PoP, less stalling events occur than at the US-PoP. In fact, when the VNF is placed at the EU-PoP, the requests from the client will be sent to the CDN streaming server which is also placed in Europe. Similar to the other scenario, when we place the VNF at the US-PoP, the client will request the video at a central streaming server in the USA. Thus, this explains that we receive more stalling events at the US-PoP scenario due to a longer distance and possible network impairments in between the client and the streaming server. Considering the stalling length in the EU-PoP scenarios, due to the lower probability of stalling occurrence than at the US-PoP, the average stalling length in the EU-PoP scenario seems to be higher. In fact, in the US-PoP scenario, we also observe several stalling events with long duration. However, due to a higher number of stalling events with several short stalling lengths, the average stalling length in the US-PoP is lower. Nevertheless, the most important result is that the local VNF always delivers an exact estimate for both stalling frequency and length. Whereas, the VNF deployed in the AWS cloud shows estimation errors. In the next sub section, we present the QoE estimation result based on stalling frequency and length described above.

6.2.2 | QoE Estimation

Figure 11 shows the QoE estimation for video streaming over the course of the measurements and at different locations. For comparison between the QoE measured at the client and estimated at the VNF, we only present the measurement replications that have stalling events, since stalling frequency and length are the main metrics for QoE estimation. In Figure 11 , the x-axis indicates various monitoring points where QoE is measured and the y-axis shows the Mean Opinion Score (MOS) representative the QoE levels. Each box shows the maximum, median in the red horizontal line and minimum MOS values. The bottom and top edges of the box plot indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers. The outliers are plotted individually using the '+' symbol. The diamond symbol in the middle of each box represents the average MOS value. The MOS is calculated from Eq. (2) based on stalling frequency and length as presented above. Note that, since the MOS is obtained from the objective QoE model presented in[44], it takes only one value per measurement run in out experiment. Therefore, the average MOS mentioned below is calculated over the course of the measurements. In addition to this, since stalling event does not occur in every replications, only measurement run has stalling is considered. The MOS can take the values: (1) bad; (2) poor; (3) fair; (4) good; (5) excellent.

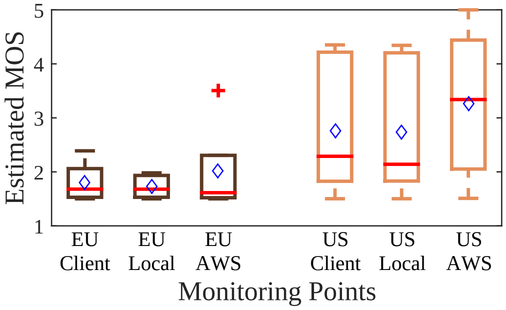


FIGURE 11 QoE Estimation at Different Monitoring Points

Figure 11 shows that the estimated MOS in the EU-PoP is smaller than in the US-PoP. It is due to the fact that, the stalling length in the EU-PoP is longer than in the US-PoP as indicated in Figure 10 b. Note that, the occurrence of stalling events is mainly dependent on the emulated network. Nevertheless, the main objective of this work does not focus on the comparison of QoE behavior between the EU and the US scenarios, but the accuracy of QoE estimation in each PoP itself. The figure shows that the local VNF can estimate the QoE with high accuracy in both the EU- and US-PoP. It is reasonable since the video flows arrive at the local VNF and show up almost instantly on the web browser due to a negligible decoding time as shown in Figure 9 . In the EU-PoP, the VNF can estimate QoE accurately in most of the cases. Due to the VNF placement at the EU-PoP near the client, the difference in arrival time of video flows is negligible at both machines. Additionally, although the mobile network can cause packet loss that induces video stalling, lost packets are retransmitted to the machines almost at the same time. This explains a high accuracy of estimated QoE at the EU-PoP.

Conversely, we observe an overestimation of QoE measured at the US-PoP as indicated in the most right box in Figure 11 . In this monitoring point, the VNF estimates a higher average MOS compared to the actual value obtained from the client and from the local VNF as well. In fact, at the US-PoP, the VNF receives the video segments faster to fill up the virtual video buffer (i.e., eVideoBuff variable in Algorithm 1). Due to the congestion of the user mobile access network and a long distance between the US-PoP and the client, the video flows arrive at the client slower. In the worst case, the video buffer at the client has been depleted while the video segment is on the fly. In the next sub-section, we investigate the influence of video segment arrival time error on the accuracy of QoE estimation.

6.2.3 | Impact of Video Segment Arrival Time on QoE Estimation

Figure 12 shows an analysis of the video segment arrival time error between the US-PoP and the client. The x-axis shows video playback time in second. The left y-axis shows the video buffer and the right y-axis indicates the segment arrival time error measured from an analysis at packet level. Note that, the arrival time of the last packet of a segment is considered as the arrival time of that segment. The saw-tooth lines show estimated and actual video buffer as described in the previous section. The black line shows the segment arrival time error, which is the difference between the estimated and actual segment arrival time. The figure shows, that the segment number 10 of the video arrives at the VNF 15.6 s faster than at the client. Thus, it fills up the virtual video buffer of the VNF earlier where a stalling event is avoided, while the client video buffer has been empty before it arrives. This is a typical reason of QoE overestimation at the VNF in this scenario. Based on this, we believe that the

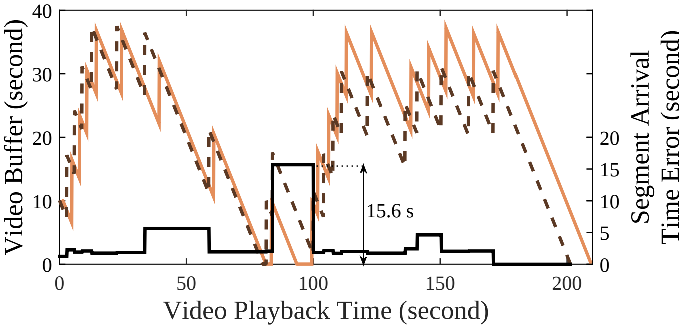


FIGURE 12 Segment Arrival Time Error Between the US-PoP and the Client

instability of the mobile access network adding a high end to end latency is an important obstacle that negatively impacts the accuracy of QoE estimation. Thus, to achieve a high accuracy in estimating QoE for video streaming, it is highly recommended to place the monitoring function at the client in high mobility environments or at the edge network.

6.3 | The Behavior of Video Buffer Monitoring VNF in Real Scenario

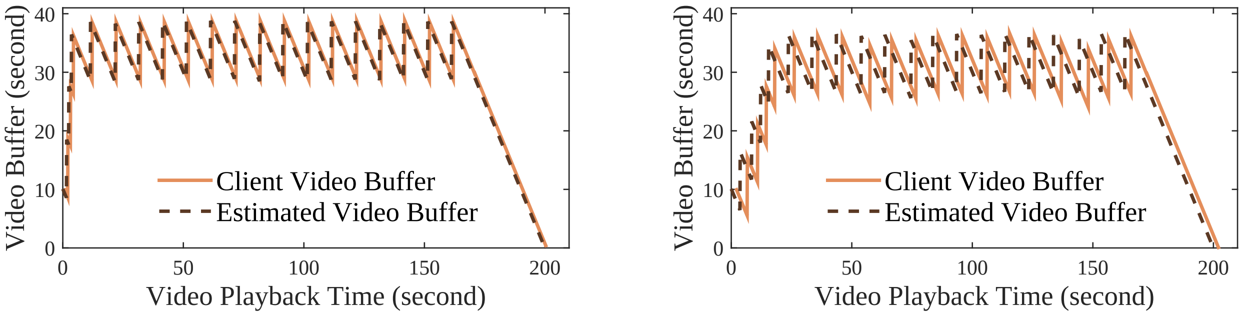
To validate our insights in a real scenario, we have implemented several experiments where the client browses video while moving in a vehicle. In this scenario, the client connects to the Internet by using the mobile network as described in Section 5.4. First, we measure the mobile network condition in between the client and the AWS cloud. Table 3 shows the average round trip time (RTT) and downstream throughput of different PoPs.

TABLE 3 Mean RTT and Throughput in Mobility Scenario

PoP	RTT	STD	Throughput	STD
EU-PoP	59 ms	15.4 ms	17.9 Mbit/s	8.9 Mbit/s
US-PoP	218 ms	22.5 ms	35.5 Mbit/s	20.5 Mbit/s

The table shows, that the network between the client and the EU-PoP has lower delay. However, both network measurements have high standard deviation values. It is due to our limited number of replications and the instability of the high mobility environment. Figure 13 shows the behavior of estimated and actual video buffer obtained from the VNF and the client browser.

In fact, we have implemented measurements for the real scenario in both city and countryside. We, however only present the experiment in country side scenario, since the difference in result between these two areas is negligible. Regarding the



(a) EU-PoP

(b) US-PoP

FIGURE 13 Video Buffer Monitoring at AWS with Mobility Client in Country Side

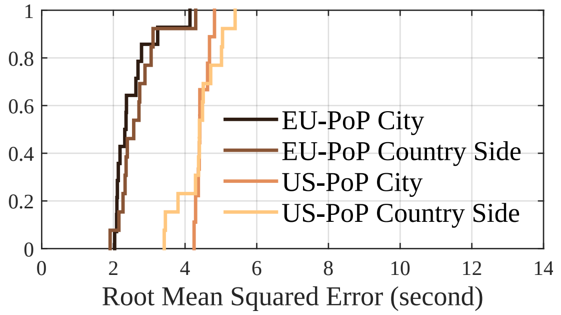


FIGURE 14 RMSE Between Estimated and Actual Video Buffer

accuracy of video buffer estimation in this scenario, the results measured at the EU-PoP show a high video buffer level and high accuracy estimation as depicted in Figure 13 a. The buffer level keeps stable in between about 39 s and 29 s, in which the higher level is the maximum amount of video buffered in the memory. The lower level is the threshold at which a new segment is requested. Conversely, the measurement results at the US-PoP show a time shifting in estimated and actual video buffer as shown in Figure 13 b. To evaluate the accuracy of the VNF monitoring, we calculate the RMSE between estimated and actual video buffer from Eq. (1). Figure 14 shows the CDF of the RMSE in all mobility scenarios.

At first, the figure shows that all errors of estimated and actual video buffer are larger than 2 s. Meanwhile, this error in the testbed scenario at the EU-PoP is generally smaller. This is due to the fact that the instability of the wireless environment influences the accuracy of the VNF monitoring. Furthermore, with additional high delay and congestion in the mobile network, the estimation error is doubled in case of the US-PoP scenario where most of the discrepancy between estimated and actual video buffer is larger than 4 s. This is reasonable since video buffer estimation relies on segment arrival time as described in previous sections. To conclude, we believe that the video buffer estimation error is highly dependent on the network delay between the client and monitoring point. Based on our analysis at packet level in Section 6.1.3, it can be seen that the estimation error trends to steadily grow up with an increasing distance between the client and monitoring point.

7 | CONCLUSION

Although the video streaming market offers the video providers a potential increase in revenue, the quality of video delivery must be ensured to meet the QoE perceived by end users. This challenges the network operators and requires a mechanism to monitor the video quality as well as QoE in the network. A possible solution is the implementation of a Virtual Network Function for video monitoring on an existing cloud management system.

In this work, we analyze and evaluate the accuracy of a VNF for video buffer monitoring in the Amazon Web Service cloud environment. Through multiple testbed and real scenarios experiments, we observe that the VNF monitoring for video buffer gains a high accuracy if it is placed at the EU-PoP nearby the client. Conversely, when the VNF is placed at the US-PoP far away from the client, it estimates the video buffer with higher error of approximately 4 s in the real scenario compared to the value measured on the client browser. In the testbed scenario, we observe that the VNF gains highest accuracy in estimating QoE for video streaming when it is placed at the client device. Besides, placing the VNF at the edge of the network also shows acceptable results for QoE monitoring. The results also indicate that the VNF overestimates QoE when it is deployed at the US-PoP faraway from the client. For future work, we plan to extend the measurements of the real scenario to different cities and regions with significant number of samples to validate the accuracy of the QoE estimation in the cloud.

References

1. Cisco Systems. *Cisco visual networking index: forecast and methodology, 2016–2021.* https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf. White Paper, Accessed April, 2018; 2017.

2. Chios M, Clarke D, Willis P, et al. *Network functions virtualisation: an introduction, benefits, enablers, challenges and call for action.* https://portal.etsi.org/nfv/nfv_white_paper.pdf, White Paper, Accessed March, 2017; 2012.

3. Brunnström K, Beker SA, De Moor K, et al. Qualinet white paper on definitions of quality of experience. *European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003).* 2012;.

4. ITUT . Subjective video quality assessment methods for multimedia applications. *Recommendation P. 910.* 2008;.

5. ITUT . Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment. *Recommendation P.913.* 2016;.

6. Pinson MH, Wolf S. Comparing subjective video quality testing methodologies. In: :573–582International Society for Optics and Photonics; 2003.

7. ITUT . Vocabulary for performance and quality of service. *Recommendation P.10/G.100.* 2006;.

8. Hoßfeld T, Keimel C, Hirth M, et al. Best practices for QoE crowdtesting: QoE assessment with crowdsourcing. *IEEE Transactions on Multimedia.* 2014;16(2):541–558.

9. Howe J. The rise of crowdsourcing. *Wired magazine.* 2006;14(6):1–4.

10. ITUT . Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference. *Recommendation J.144.* 2004;.

11. Dinh-Xuan L, Seufert M, Wamser F, Tran-Gia P. Qoe aware placement of content in edge networks on the example of a photo album cloud service. In: ; 2016.

12. Fiedler M, Hossfeld T, Tran-Gia P. A generic quantitative relationship between quality of experience and quality of service. *IEEE Network.* 2010;24(2).

13. Egger S, Reichl P, Hoßfeld T, Schatz R. "Time is bandwidth"? Narrowing the gap between subjective time perception and Quality of Experience. In: :1325–1330IEEE; 2012; Ottawa, Canada.

14. Egger S, Hoßfeld T, Schatz R, Fiedler M. Waiting times in quality of experience for web based services. In: :86–96IEEE; 2012; Melbourne, Australia.

15. Hoßfeld T, Seufert M, Hirth M, Zinner T, Tran-Gia P, Schatz R. Quantification of YouTube QoE via crowdsourcing. In: ; 2011; California, USA.

16. Takahashi A, Hands D, Barriac V. Standardization activities in the ITU for a QoE assessment of IPTV. *IEEE Communications Magazine.* 2008;46(2).

17. De Vera D, Rodríguez-Bocca P, Rubino G. QoE monitoring platform for video delivery networks. In: :131–142Springer; 2007; San JosÃľ, USA.

18. Piamrat K, Viho C, Bonnin JM, Ksentini A. Quality of experience measurements for video streaming over wireless networks. In: :1184–1189IEEE; 2009; Las Vegas, NV, USA.

19. Orsolic I, Pevec D, Suznjevic M, Skorin-Kapov L. Youtube QoE estimation based on the analysis of encrypted network traffic using machine learning. In: :1–6IEEE; 2016; Washington, DC, USA.

20. Staehle B, Hirth M, Pries R, Wamser F, Staehle D. YoMo: A YouTube application comfort monitoring tool. 2010;.

21. Wamser F, Seufert M, Casas P, Irmer R, Tran-Gia P, Schatz R. YoMoApp: A tool for analyzing QoE of YouTube HTTP adaptive streaming in mobile networks. In: ; 2015; Paris, France.

22. Schatz R, Hoßfeld T, Casas P. Passive youtube QoE monitoring for ISPs. In: IEEE; 2012; Palermo, Italy.

23. Casas P, Schatz R, Hoßfeld T. Monitoring YouTube QoE: Is your mobile network delivering the right experience to your customers?. In: IEEE; 2013; Shanghai, China.

24. Wamser F, Casas P, Seufert M, Moldovan C, Tran-Gia P, Hoßfeld T. Modeling the YouTube stack: from packets to quality of experience. *Computer Networks.* 2016;.

25. Casas P, Seufert M, Schatz R. YOUQMON: A system for on-line monitoring of YouTube QoE in operational 3G networks. *ACM SIGMETRICS Performance Evaluation Review.* 2013;41:44-46.

26. Yao J, Kanhere SS, Hossain I, Hassan M. Empirical evaluation of HTTP adaptive streaming under vehicular mobility. In: :92–105Springer; 2011; Valencia, Spain.

27. Müller Christopher, Lederer Stefan, Timmerer Christian. An evaluation of dynamic adaptive streaming over HTTP in vehicular environments. In: :37–42ACM; 2012; Chapel Hill, NC, USA.

28. Seufert M, Wamser F, Casas P, Irmer R, Tran-Gia P, Schatz R. YouTube QoE on mobile devices: Subjective analysis of classical vs. adaptive video streaming. In: ; 2015; Dubrovnik, Croatia.

29. Eittenberger PM, Hamatschek M, Großmann Ml, Krieger UR. Monitoring mobile video delivery to android devices. In: :119–124ACM; 2013; Oslo, Norway.

30. Dinh-Xuan L, Seufert M, Wamser F, Tran-Gia P. Study on the accuracy of QoE monitoring for HTTP adaptive video streaming using VNF. In: ; 2017; Lisbon, Portugal.

31. Oechsner S, Ripke A. Flexible support of VNF placement functions in OpenStack. In: :1–6IEEE; 2015; London, UK.

32. Carella G, Corici M, Crosta P, et al. Cloudified IP multimedia subsystem (IMS) for network function virtualization (NFV)-based architectures. In: :1–6IEEE; 2014; Madeira, Portugal.

33. Soares J, Gonçalves C, Parreira B, et al. Toward a telco cloud environment for service functions. *IEEE Communications Magazine.* 2015;53(2):98–106.

34. Yu R, Xue G, Kilari VT, Zhang X. Network function virtualization in the multi-tenant cloud. *IEEE Network.* 2015;29(3):42–47.

35. ETSI . Network functions virtualisation (nfv): Architectural framework. *ETSI GS NFV.* 2013;2(2):V1.

36. Merkel Dirk. Docker: Lightweight linux containers for consistent development and deployment. *Linux Journal.* 2014;2014(239):2.

37. Jinesh V, Sajee M. *Overview of amazon web services.* https://media.amazonwebservices.com/AWS_Overview.pdf, White Paper. Accessed January, 2017; 2014.

38. Seufert M, Egger S, Slanina M, Zinner T, Hoßfeld T, Tran-Gia P. A survey on quality of experience of HTTP adaptive streaming. *IEEE Communications Surveys & Tutorials.* 2015;17(1):469–492.

39. Müller C, Timmerer C. A VLC media player plugin enabling dynamic adaptive streaming over HTTP. In: MM '11:723–726ACM; 2011; New York, USA.

40. Stockhammer T. Dynamic adaptive streaming over HTTP: Standards and design principles. In: ACM; 2011; Santa Clara, CA, USA.

41. Hemminger S. Network emulation With NetEm. In: ; 2005; Canberra, Australia.

42. Hooft J, Petrangeli S, Wauters T, et al. HTTP/2-Based adaptive streaming of hevc video over 4G/LTE networks. *IEEE Communications Letters.* 2016;20(11):2177-2180.

43. Allman M, Paxson V, Blanton E. *TCP congestion control.* 5681: RFC Editor; 2009.

44. Hoßfeld T, Schatz R, Biersack E, Plissonneau L. Internet video delivery in youtube: From traffic measurements to quality of experience. In: Springer, Berlin, Heidelberg 2013 (pp. 264–301).

How to cite this article: L. Dinh-Xuan, C. Popp, V. Burger, F. Wamser, and T. Hoßfeld (2018), Impact of VNF Placements on QoE Monitoring in the Cloud, *IJNM Special Issue on QoE-centric Analysis and Management of Communications Networks*, *2018*.