ARTICLE IN PRESS
JID: COMCOM [m5G;September 12, 2016;6:18]

Computer Communications 000 (2016) 1–11

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Network-based video freeze detection and prediction in HTTP adaptive streaming

Tingyao Wu[a,*], Stefano Petrangeli[b], Rafael Huysegems[a], Tom Bostoen[a], Filip De Turck[b]

[a] Bell Labs, Nokia, Copernicuslaan 50, B-2018 Antwerpen, Belgium
[b] Department of Information Technology (INTEC), Ghent University- iMinds, Technologiepark-Zwijnaarde 15, 9052 Ghent, Belgium

ARTICLE INFO

Article history:
Received 6 April 2016
Revised 5 August 2016
Accepted 8 August 2016
Available online xxx

Keywords:
HTTP adaptive streaming
Decision tree C4.5
Freeze detection
Freeze prediction
Quality of experience
Apple live streaming
Microsoft smooth streaming
DASH

ABSTRACT

Given the popularity of HTTP adaptive streaming (HAS) technology for media delivery over mobile and fixed networks, the clients Quality of Experience (QoE) for HAS video sessions is of particular interest for network providers and Content Delivery Network (CDN) operators. Despite that, network providers are not able to directly obtain QoE-relevant metrics such as video freezes, initial buffering time, and the frequency of quality switches from the client. This paper proposes a scalable machine learning (ML) based scheme that offline detects and online predicts video freezes using a few features extracted from the network-based monitoring data, i.e., a sequence of HTTP GET requests sent from the video client. We determine the discriminative features for detecting video freezes based on multi-scale windows using the criterion of information gain (IG). Four traditional classifiers are investigated and the C4.5 decision tree is selected because of its simplicity, scalability, accuracy, and interpretability. Our approach for session-based offline freeze detection is evaluated on the Apple HTTP Live Streaming video sessions obtained from a number of operational CDN nodes and on the traces of Microsoft Smooth Streaming video sessions acquired in a controlled lab environment. Experimental results show that, even with the disturbance of user interactivity, an accuracy of about 91% can be obtained for the detection of a video freeze, a long video freeze, and multiple video freezes. The experiments for the online freeze prediction show that more than 30% of the video freezes can be foreseen one segment in advance.

1. Introduction

Video streaming accounts for more than half of the traffic over the Internet. For video delivery, HTTP Adaptive Streaming (HAS) has become a key technology [1]. HTTP adaptive streaming technologies include MPEG-DASH, Apple's HTTP Live Streaming, Microsoft's Smooth Streaming, Adobe's Dynamic Streaming, etc. In an HAS architecture, video content, which is hosted on an HTTP Web server, is encoded in different quality levels (bit rates), and chunked into independent segments. An HAS client sequentially requests segments with potentially different qualities using HTTP GET requests. The retrieved segments can be played as a seamless video, possibly combining different quality levels. The key feature of HAS is that the client is responsible for determining which quality level to download according to its available resources. One of the main advantages of HAS as a delivery method is that, the client can adapt the video quality based on the perceived bandwidth. This allows the client to smoothly play the video with a low quality even when the perceived bandwidth is limited, while when a higher bandwidth is available, it allows the client to request higher quality levels.

With the pervasion of HAS deployments, network and Content Delivery Network (CDN) operators are increasingly interested in knowing the end user's quality of experience (QoE) of HAS sessions, because the HAS QoE reveals the satisfaction of the end users and the engagement with their services. Despite that, obtaining the subjective QoE perception is costly and not scalable. Alternatively, researchers on video QoE try to link the objective QoE metrics, including the statistics of bit rates, initial buffering delay, video freezes etc., to the actual perceived QoE (for example [2]). The network and CDN operators could benefit from these models: by obtaining or predicting the objective QoE metrics, they can infer the QoE of the end-users. For instance, aggregated HAS QoE metrics over different video sessions provide a good view on the usage of the network; a low average viewing quality or frequent freezs could indicate that the capacity of certain network links is under-dimensioned and should be increased. Another example is that, the network provider can prioritize the

* Corresponding author.
E-mail addresses: tingyao.wu@nokia.com, tingyao.wu@gmail.com (T. Wu).

http://dx.doi.org/10.1016/j.comcom.2016.08.005
0140-3664/© 2016 Published by Elsevier B.V.

delivery of the segment whose client is approaching a freeze, actively preventing that the HAS client experiences a true re-buffering event.

However, the full QoE metrics are typically not available to network providers. While an HAS client may report QoE metrics to the Web server (for instance, the "advanced logging" feature in Microsoft Smooth Streaming [3]), this information is typically not passed to network providers. During the video delivery, the network nodes only act as receiving and forwarding points for the communication between the HAS clients and HAS server or CDN. This means that the network provider can access the flow of HTTP GET requests sent by the video client when requesting a new segment to download. Consequently, the network operator can mine the sequence of HTTP GET requests to retrieve the QoE metrics of the analyzed HAS session. Indeed, in our previous study [4], we demonstrated that from the sequence of GET requests collected at intermediate network elements, the HAS session can be reconstructed to derive all QoE-related metrics. These metrics include the average playout quality, changes in the play-out quality, re-buffering due to buffer starvation, re-buffering caused by interactivity, etc.

Among all HAS impairments, video freezes are undoubtedly the most deleterious factor with respect to end-users' satisfaction and engagement. The 2015 annual report of Conviva [5] shows that just 1% increase of the re-buffering ratio could reduce the video engagement with 14 min. With respect to QoE metrics mining, identifying or predicting re-buffering events is thus the most important task. In [6], we propose a scalable tree-based in-network freeze detection technique under the assumption of no user interactivity. The results demonstrated that the machine learning (ML)-based method can quickly identify the existence of freeze in an off-line video session with a substantial accuracy.

This paper extends our previous work [6] in two folds. First, we examine the performance of the session-level freeze detection with the additional complexity of user interactivity, including pause and re-positioning. We discover that pause events make it more complex to discriminate between sessions with/without freezes. Second, we propose a method to online predict whether the delivery of a video segment could lead to a freeze at the client side in the near future. [7] shows that if the client could report to the network nodes the buffer-filling level on a regular basis, the network provider could prioritize the delivery of particular segments to effectively reduce the occurrence of video freezes. While obtaining the report from the client is typically unrealistic, our freeze prediction method actually provides an in-network solution to know the likelihood of a freeze at the client in the near future. Nevertheless, coupling the result of online freeze prediction and segment prioritization is out of scope for this paper.

The remainder of this paper is organized as follows. Section 2 gives an overview of the research on HAS QoE. Section 3 presents the proposed freeze detection and prediction framework. In Section 4, we detail the machine learning-based approach for the session-level freeze detection and the segment-level freeze prediction. In Section 5 we describe the experiments and the corresponding results. The conclusions and future work are given in Section 6.

2. Related work

2.1. QoE in HTTP adaptive streaming

Lately, the HAS-based QoE research has received a lot of attention. HAS was inherently designed to adapt to volatile network environments by increasing and decreasing quality levels using a buffer to reduce the occurrence of video freezes. Thus HAS is considered to be an effective technique to enhance the end-user's QoE. By contrast, classical HTTP video streaming (e.g., progressive downloading) is only protected to fluctuating network conditions by maintaining a sufficiently high buffer at the client. In [8], Liu et al. study the QoE of DASH in comparison with fixed-bit-rate streaming and provide evidence that adaptive streaming improves end-users' QoE greatly. This advantage is also verified by Yao et al. [9]. They compare adaptive and fixed-rate streaming under vehicular mobility circumstances. Their work reveals that HAS can effectively reduce the freeze duration by 80%.

Comparing to the traditional video delivery techniques, HAS expands a single quality level into multiple quality levels. This extension results in a new metric relevant for the QoE, namely the frequency of video quality changes. In fact, traditional quality metrics (Peak Signal to Noise Ratio, Structural Similarity and Video Quality Metric, etc.) fail to predict user-perceived video quality due to the rate adaptation inherent to HAS systems [10]. Instead, the initial buffering time, number of freezes, duration of freezes, (average) playback quality, bit-rate switching frequency, etc., are the objective metrics that determine the user's QoE in HAS system [11]. In [12], the authors overview the recent studies targeting the optimization of HAS adaptation strategy, and also highlight the difficulties for accessing HAS QoE. Seufert et al. [13] provide a comprehensive survey of the relationship of HAS and subjectively perceived quality. In the survey, the main influence factors of the QoE of HTTP video streaming are discussed, and the influence of temporal resolution, spatial resolution and video quality on QoE are also presented.

2.2. Freeze reduction in HAS client

To reduce the occurrence of freezes, many techniques rely on improving the client rate-determination algorithm (RDA). For instance, Claeys et al. [14] and Wu et al. [15] use reinforcement learning in the RDA. The RDA learns to request the most suitable quality level given the client's instant status by progressively optimizing the QoE-related reward. For this technique, the punishment for a freeze is about two orders of magnitude larger than a low playback quality and quality switching. In [16], an HAS RDA was proposed that detects bandwidth changes using a smoothed HTTP throughput measure based on the segment fetch time. This algorithm increases the video quality stepwise and aggressively decreases the quality level when a sudden bandwidth drops occurs to avoid a video freeze. In [17] and [18], Huang et al. argue that only observing and controlling the playback buffer, without taking the available network bandwidth into account, is already sufficient to avoid unnecessary re-buffering and to control the delivered video quality.

2.3. Network-based freeze detection and reduction

In-network HAS QoE monitoring provides a solution for network and CDN providers to determine whether the end-user is satisfied with the video service. Huysegems et al. [4] propose a session reconstruction technique applied on an intermediate network node. The technique is able to fully recover how the media content was consumed by the client and all QoE metrics can be retrieved using the sequence of intercepted HTTP GET messages. However, this technique requires manual effort to create a suited set of reconstruction rules for each new HAS client.

Researchers not only study on monitoring the HAS QoE from the network, but also try to mitigate video freeze and optimize the QoE for multiple clients from the network. In [19], Bouten et al. present merits of deploying intelligent network elements that manage the QoE in an HAS delivery network. The authors argue that a combination of HAS-enabled clients with an operator-centric or user-centric policy enforced at intelligent proxies leads