An Objective Approach to Measuring Video Playback Quality in Lossy Networks using TCP

Timothy Porter and Xiao-Hong Peng

Abstract—In this work, we investigate a new objective measurement for assessing the video playback quality for services delivered in networks that use TCP as a transport layer protocol. We define the new metric as pause intensity to characterize the quality of playback in terms of its continuity since, in the case of TCP, data packets are protected from losses but not from delays. Using packet traces generated from real TCP connections in a lossy environment, we are able to simulate the playback of a video and monitor buffer behaviors in order to calculate pause intensity values. We also run subjective tests to verify the effectiveness of the metric introduced and show that the results of pause intensity and the subjective scores made over the same real video clips are closely correlated.

Index Terms—Pause intensity, video playback quality, TCP, buffer behavior.

I. INTRODUCTION

THE development of the Real-time Streaming Protocol (RTSP) revolutionized the way multimedia content was delivered over the Internet and was used in combination with the Real-time Transport Protocol (RTP) to stream video on-demand [1]. Since RTP uses the connectionless User Datagram Protocol (UDP) as its underlying transport mechanism, packet loss, delay and re-ordering incidents were expected within congested networks. Whilst a local buffer allowed for jitter, and RTP sequencing for re-ordering, the quality of playback would almost certainly be degraded due to packet loss, regardless of the buffer size.

Modern Internet video playback applications, however, are predominantly Flash based [2], such as Youtube and BBC iPlayer [3] (in the UK), but the more conventional standalone media players such as Windows Media Player and VideoLAN Client (VLC) are still widely in use and under constant development. In either case, the Transmission Control Protocol (TCP) is used as the transport protocol which now becomes the preferred method of transport rather than a RTP/UDP combination [4]. This is because the majority of video content delivery over the Internet is not live and user’s bandwidth is often many times greater than the video bitrate, so buffering in conjunction with retransmission through TCP allows mitigating the effect of packet loss. As a result, streams remain TCP-friendly [5] which is important for scalability. The loss recovery mechanism provided by TCP also allows the use of newer codecs, such as H.264, with high compression rates which are less resilient to data loss [6].

Existing objective measures of perceived video quality, such as the Peak Signal-to-Noise Ratio (PSNR), are more suited to objectifying the performance of compression algorithms [7]. Moreover, these measurements can only be used when comparing two streams of identical synchronicity, and when one stream is degraded in comparison to the other. This type of measurement is particularly useful when dealing with a UDP video stream [8], since lost packets are not retransmitted. When using TCP, however, packets are always recovered if loss occurs, and so momentary quality degradation is rarely seen. This indicates the unsuitability of PSNR as a metric for quality assessment in this case, and alternative metrics for this type of applications are not yet available [9]. For this reason, we turn our focus on buffer behavior because video pausing due to buffer under-runs has become the main source of quality concerns in video streaming using TCP.

In this letter, we propose a novel method to characterize the perceived quality of a video stream by measuring the degree of interruption caused by pauses during playback. We show how packet loss affects the number of pauses, the average duration per pause and, ultimately, the pause intensity to be defined, given a set of playback criteria such as the playback rate, available bandwidth and buffer size. We also validate the proposed quality metric by showing its correlation with the subjective measurements conducted in the same streaming environment.

II. PAUSE INTENSITY

In a TCP network, the service delivery time varies as lost packets are retransmitted, causing delays and variable incoming data rates at the receiver buffer. If the playback rate is greater than the incoming data rate, the buffer will be utilized and eventually become exhausted, causing the playback to pause. To measure the effect of buffer exhaustion on the perceived playback quality, we introduce a metric termed pause intensity which is contributed jointly by both the number and the duration of pauses during playback. Given the number of pauses per unit time \( u \) and the average duration per pause \( v \), the pause intensity \( I_p \) is defined as

\[
I_p = uv
\]  

(1)

If \( V \) is a discrete random variable representing the pause duration, which attains values \( v_1, v_2, \ldots \) with occurrence frequency \( f(v_1), f(v_2), \ldots \), the pause intensity can also be obtained by

\[
I_p = \sum_{i \in \Phi} [v_i f(v_i)]/T
\]  

(2)
where $T$ is the total playback time and $\Phi$ is an index set for all the pause durations that have been recorded within $T$.

Using this metric, we are able to quantify the effect of pauses that directly hamper the perceived quality in terms of the continuity of playback. As the continuity is closely related to the loss performance of data delivery in a TCP network, we will show, through the experiments described below, the relationship between pause intensity and packet loss rates.

### III. EXPERIMENT SETUP

The experimental testbed is based on a TCP connection established between two Linux-based machines (a server and a client), using the `iperf` network utility to generate packets and the Traffic Control (TC) utility to limit the transmission rate to 1 Mbit/s. This closely mimics the traffic pattern of a video being streamed, for example over the Hyper Text Transfer Protocol (HTTP) which is typical. A video bit rate of 900Kbit/s is selected for the video playback simulator, allowing noticeable observation in changes to the buffer when packet loss is introduced. The network monitoring tool, `tcpdump`, is used to capture and store the timestamp and length of every packet arriving at the `iperf` client. The standard TCP Cubic [10] is used.

A third machine is configured using the TC utility, bridging the server and client, to drop packets at rates between 0%-10%, incrementing at a 0.5% interval for each capture. We use a randomly distributed loss algorithm to ensure each test is comparable; the average packet loss is calculated based on the entire test duration.

Capture files provided by `tcpdump` are parsed by the simulator to control the buffer of a video playback program growing and shrinking accordingly. With a pre-defined buffer size of 5Mbit and a video playback rate close to that of the available bandwidth, we are able to monitor the utilization of the buffer, the moments when it becomes exhausted and when it recovers. Two thresholds are set in the testbed for playback to start and stop at certain buffer usage levels which indicate the remaining amount of video data cached in the buffer in Mbits.

### IV. RESULTS AND ANALYSIS

Firstly, Fig. 1 demonstrates buffer behaviors during playback when there is no loss (a), and at a loss rate of 1.5% (b), respectively. The two thresholds marked show when playback will be paused (the upper line) and subsequently resumed should the buffer recover (the lower line). The duration of each video clip is 1600 seconds if uninterrupted.

As shown by Fig. 1(a), during the pre-buffering stage before playback begins, the buffer fills rapidly since there is no outgoing data at this time. When the upper threshold is met, the buffer fill rate reduces due to playback of video commencing. The buffer continues to fill at this reduced rate until the maximum capacity of the buffer is reached. Since there is no loss, and the bandwidth available is sufficiently higher than the playback rate, the buffer remains full for the duration of the video clip. When the average packet loss rate increases to 1.5% in the case of Fig. 1(b), losses cause the input data rate to decrease below the playback rate. As expected, the buffer begins to empty until the lower threshold is met and playback is paused. In this example, we see the playback pausing on several occasions whilst the buffer recovers.

Collectively, over an extended range of packet loss rates, we determine the number of pauses experienced during playback, and the duration of the pause in each case. In Fig. 2, the number of pauses per unit time (second), $u$, and the average duration in seconds of each pause, $v$, are shown for the range of packet loss rates tested by Fig. 2(a) and Fig. 2(b), respectively. As we can see, when the packet loss rate is below 1% the buffer is able to maintain playback for the entire duration of video playback with little or no interruption. As the loss rate exceeds 1%, the buffer is seen to deplete on one or more occasions, leading to pauses in playback. For moderate levels of packet loss, e.g., 1%-6%, the duration of each pause is predominantly between 1.5 and 2 seconds, despite the pauses themselves becoming more frequent.

The distributions of pause durations in terms of their occurrence frequency are shown in Fig. 3 at different packet loss rates. It is observed that, when network conditions become far poorer, the number of pauses begins to plateau, while the duration of each pause becomes increasingly prolonged.
Based on the results given in Figs. 2 or 3, pause intensity for individual loss rates can be obtained using either Eq. 1 or Eq. 2. In particular, at a specific loss rate, the occurrence frequency $f(v_i)$ for each duration value $v_i$ ($i = 1, 2, \ldots$) can be found from the corresponding distribution graph, e.g. in Fig. 3, and then Eq. 2 is applied to calculate the pause intensity value. As a result, the relationship between pause intensity and packet loss performance is shown in Fig. 4. It can be seen that pause intensity scales proportionally with the packet loss rate across a range of realistic network conditions, whilst it is untrue if just looking at the performance of the number of pauses or pause duration alone as shown in Fig. 2.

To further establish the effectiveness of the metric defined, a subjective survey was conducted using a series of videos that were paused at the rates and durations with respect to the packet loss rates in our tests. Forty users participated in the survey and gave scores whilst watching the videos. A Mean Opinion Score (MOS) scale from 5 down to 1: Imperceptible, Perceptible but not annoying, Slightly annoying, Annoying and Very annoying, was used. The subjective result against the objective measurement (pause intensity) based on the same set of videos is shown in Fig. 5. Clearly, pause intensity is seen to be closely correlated with the subjective measurement, and hence can be used to objectify the quality of playback for video streaming over TCP.

V. CONCLUSION

We propose and experimentally evaluate a new objective metric, namely the pause intensity, for estimating the video playback quality. This metric is especially useful for video streaming over TCP/IP networks as in this case the conventional objective measurement such as PSNR does not apply. In future work, we intend to use pause intensity as a target parameter for selecting appropriate technologies towards performance optimization in video streaming applications.

REFERENCES