Low Delay Streaming of DASH Content with WebRTC Data Channel

Shuai Zhao, Zhu Li, Deep Medhi

Computer Science & Electrical Engineering Department
University of Missouri–Kansas City, USA
{szb53, lizhu, dmedhi}@mail.umkc.edu

***Abstract*— Instantaneous low delay on-demand video streaming with a very low start up and channel switching delay is highly desirable for users. The dominant over-the-top (OTT) solutions like DASH, which is based on HTTP and/or WebSocket, suffer from a slow start of the underlying TCP transport while the typical coding structure of the DASH content introduces additional delays. In this work, we address these issues with a new low delay transport based on a WebRTC data channel along with a new content side packetization scheme based on a new QoE metric driven DASH sub-representation to facilitate a fast start, with an agile and fine granular rate adaptation. Simulations on both NS-3 and the GENI testbed demonstrate the effectiveness of this approach. This can serve as a basis for a further peer assisted low delay over-the-top (OTT) solution for an instantaneous live video streaming solution.**

Keywords—DASH; WebRTC; OTT; Low Delay; WebSocket

I. Introduction

Video has become the dominating traffic over the Internet and with the recent development of the High Efficiency Video Coding (HEVC) standard [2] and maturation of the HTTP video streaming solution, such as DASH [1], that utilize the widely available HTTP infrastructure, over-the-top (OTT) video solution companies (like Netflix) are experiencing rapid expansion of service and dramatic growth in business. Compared with the managed network operators like COMCAST and TWC, which own the IP networks with better managed resource and QoS, one significant drawback of the OTT solution is that the stream start delay and channel switch delay are significantly longer than an IP network based solution.

Recently, IETF [5] developed a browser native solution for end-to-end video communication called WebRTC that abstracts the end point connectivity, fire wall traversal, congestion measure and control into a browser-based solution. Both Google and Firefox have implemented and deployed WebRTC support in Chrome and Firefox browsers, respectively. WebRTC is not TCP/HTTP based and therefore, has a much lower start up delay and potentially a much better link capacity utilization. The standard video segmentation in DASH has been explored in [7] and [8]. In this work, we investigate utilizing the data channel function from WebRTC connectivity, experiment a low delay DASH streaming solution, capitalize the built-in low delay and higher utilization of WebRTC link, with an Application layer video buffer and QoE metric driven self-pacing solution. This could lead to delivering a low delay DASH video at a much better end-to-end QoE. On the content side, the authors in [4] developed a new QoE metric for the MPEG file format to characterize frame loss induced distortion. This new QoE metric can be utilized to drive a DASH sub-representation scheme that offers a spatio-temporal QoE tradeoff at a much finer granular scale. This is ideally suited for the low delay streaming applications, as the "wiggle" room is very limited to prevent playback buffer underflow.

II. Fine Granular QoE/QoS Adaptation with DASH Sub-Representation

The innovation introduced on the content side is to introduce a temporal QoE metric [4] that can characterize the frame loss induced distortion to the playback of the video. Let the frame visual significance be the relative importance of a frame in a sequence, defined as the frame-by-frame difference computed as a scaled Eigen appearance distance

$$v_k = d(f_k, f_{k-1}) \quad (1)$$

where the distance is computed as scaled frame thumbnails in their Eigen appearance space

$$d(f_k, f_{k-1}) = \|AS(F_k) - AS(F_{k-1})\| \quad (2)$$

where S is the bi-linear scaling of the video frames to the h x w thumbnails and stacked into a vector in R^{hxw}, A is the PCA projection of the thumbnails in R^{hxw} to a low k-dimensional space. With this new frame significance function, we can define any frame loss from a sequence, as an exponentially decaying integration over the visual significance loss. Let the frame loss index be $L=\{l_1, l_2, \ldots, l_m\}$, and for lost frame l_k, the previous lost frame be p_k, then the loss for the sequence is given by

$$d(L) = \sum_{k=1}^{m} e^{-h(l_k - p_k)} v_k \quad (3)$$

Armed with this frame loss distortion metric, we can sort the I, B, and P frames in a video segment and characterize their loss consequences in the bits saved and introduced distortion. With decoding dependency considered, we can use a simple gradient search algorithm to organize the DASH segment into several sub-representations, that represent different temporal quality layers. Examples of such a temporal sub-representation scheme are illustrated in Fig. 1 where the operating points on rate reduction and temporal distortion are computed at the encoding time. The sample index associated with each temporal QoE and rate operating points are signaled.

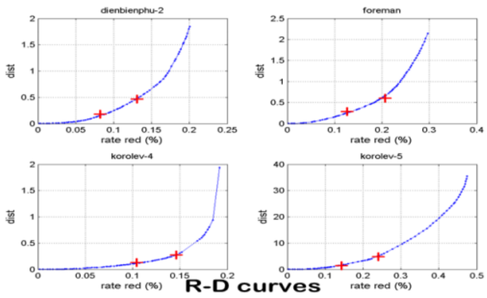

Fig 1. DASH Sub-Representation Packetization

The benefit of such a scheme is that instead of switching at the end of a DASH segment duration, streaming can be switched at the sub-representation level to better adapt to throughput changes. This may lead to a more flexible QoE adaptation, not only at the PSNR quality layers, that were pre-coded, but also to exploit the temporal QoE and rate tradeoffs, introduced by the sub-representation (see Fig. 2). For the 3 segments of content coded at 3 different rates, each segment consists of sub-representations L_k^j, where k is the PSNR quality layer index and j is the temporal layer index.

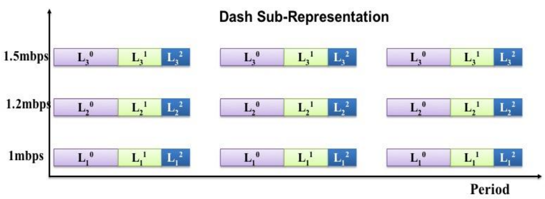

Fig 2. Sub-Representation Switching During Streaming Time.

This offers a much finer granular streaming time for QoE and QoS operating points and is essential for low delay streaming applications where the buffer level is very low. The example in Fig. 2 shows that for a 1.5 *Mbps* representation, if the channel throughput is reduced, then instead of switching all the way to a 1.2 *Mbps* representation, we can stay at a 1.5 *Mbps* PSNR layer by skipping certain temporal layers. This offers more flexibility and much finer granular rate operations that can maximize end-to-end received spatio-temporal QoE.

III. Low Delay DASH over WebRTC

WebRTC [5] is the new web browser based end-to-end connectivity solution that has a new transport and congestion control scheme to better utilize the link capacity and reduce delay. Data channel can be established in many secure ways, such as using Firebase [6]. We experimented with our sub-representation based DASH streaming over the WebRTC data channel as shown in Fig. 3.

A. Simulation Topology

We simulated our experiment using the NS-3 simulator (Fig. 3). In order to fully take control of existing WebRTC APIs, we created two Linux containers and connected them to the NS-3 network. Firebase is used for establishing secure data communication channels for two WebRTC applications. Once

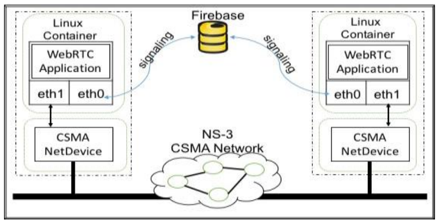

Fig 3. WebRTC Communication Topology

signaling is finished, both sides can send and receive WebRTC traffic simultaneously.

B. Preliminary Result

We simulated both the sender and receiver sides of the WebRTC data channel as shown in Fig.3. Link bandwidth was set to be 1 *Mbps*. We used the default 16*KB* chunk size and the "*bufferedamountlow*" event for the sender's side flow control. Fig.4 shows that the receiver maintains the maximum throughput. The initial receiving time difference between the sender and the receiver is 170*ms* (not shown in the graph).

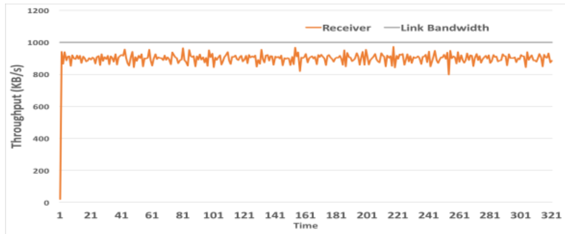

Fig.4 WebRTC Receiving Throughput

Our preliminary result shows that the WebRTC has a noticeably low start delay and high link bandwidth utilization (shown in Fig. 4). We believe that compared with a regular DASH streaming over HTTP, with an Apache server and DASH.js web based client, the new scheme would have a lower delay, and a better received QoE when throughput fluctuates between two PSNR quality layers, e.g, 1.0 *Mbps* and 1.2 *Mbps*, due to more operating points from sub-representation. A side-by-side demo is being planned for the final presentation.

References

[1] I. Sodagar: "The MPEG-DASH Standard for Multimedia Streaming Over the Internet". IEEE MultiMedia 18(4): 62-67 (2011).

[2] ISO/IEC 23008-1, Text of ISO/IEC 2nd CD 23008-1 MPEG Media Transport.

[3] J-R Ohm, G J. Sullivan: High Efficiency Video Coding: The Next Frontier in Video Compression [Standards in a Nutshell]. IEEE Signal Process. Mag. 30(1): 152-158 (2013)

[4] Z. Li, and I.Bouazizi, "FF: Temporal Quality Signalling in ISO Based Media File Format", ", ISO/IEC/JTC1/MPEG2014/m33239.

[5] RFC 7484, Web Real-Time Communication Use Cases and Requirements

[6] Firebase. [online] https://www.firebase.com

[7] P. Juluri,V.Tamarapalli, D. Medhi, "SARA: Segment aware rate adaptation algorithm for Dynamic Adaptive Streaming over HTTP," IEEE ICC workshop on QoE-FI, London, UK, June 2015.

[8] P. Juluri,V.Tamarapalli, D. Medhi, "QoE management in DASH systems using the segment aware rate adaptation algorithm," IEEE NOMS, April 2016