

# CBR ve VBR Kodlanmış Ölçeklenir Videolar ile Uyarlanabilir İletimin Karşılaştırmalı Analizi

## Comparative Analysis of Adaptive Streaming with CBR and VBR Encoded Scalable Videos

Mehmet ÇALI

Elektrik – Elektronik Mühendisliği  
İzmir Yüksek Teknoloji Enstitüsü  
İzmir, Türkiye  
mehmetcali@iyte.edu.tr

Nükhet ÖZBEK

Elektrik – Elektronik Mühendisliği  
Ege Üniversitesi  
İzmir, Türkiye  
nukhet.ozbek@ege.edu.tr

**Özetçe**—Ağ kaynaklarının etkin şekilde kullanılmasında rol oynayan farklı ağ koşullarında optimum video akışının gerçekleşmesi konusu önem arz etmektedir. Bu nedenle son yıllarda özellikle DASH üzerindeki optimizasyon çalışmaları araştırmacıların dikkatini çekmektedir. Bu çalışmaların çoğu istemci kısmındaki uyarlama algoritmasının geliştirilmesine odaklanmış olsa da sunucu tarafındaki video kodlama sırasında alınan kararların video aktarımında algılanan kaliteye etkisi azımsanamaz. Bu çalışmada, ölçeklenir video kodlamada kodlama modu seçiminin algılanan kaliteye etkileri araştırılmıştır. CBR ve VBR modu ile kodlanan videoların aktarımı için üç farklı uyarlama algoritması uygulanmış ve geniş kapsamlı ağ koşullarında test edilmiştir. VBR modunun hem video donma süresi hem de video kalitesi değişim frekansı açısından önemli bir farkla üstün performansa sahip olduğu görülmüştür.

**Anahtar Kelimeler** — DASH; uyarlanabilir aktarım; SVC; CBR; VBR.

**Abstract**—Having an influence on the effective utilization of network resources, optimum video streaming in various network conditions have a significant role. Therefore, optimization studies on DASH receives considerable attention from the researchers. Although most of these studies focus on developing adaptation algorithm in the client side, the effect of decisions made during video encoding on the perceived quality cannot be underestimated. In this study, the effect of encoding mode selection on the perceived quality for scalable video coding is investigated. Streaming of videos which are encoded in CBR and VBR modes are tested with extensive network conditions by implementing three different adaptation algorithm. It is observed that VBR encoding has superior performance with a considerable margin in terms of both video stall duration and video quality switching frequency.

**Keywords** — DASH; adaptive streaming; SVC; CBR; VBR.

### I. GİRİŞ

Günümüzde internet trafiğinin önemli bir bölümünü video akışı uygulamaları oluşturmaktadır. HTTP Üzerinden Dinamik Uyarlanabilir Video Akışı (DASH), internet kalitesine göre video bit hızının zamana bağlı olarak ayarlanabilmesini

sağlayan bir standarttır. Bu standartta videolar sunucuda gösterim olarak adlandırılan çeşitli bit hızı seviyelerinde segment olarak adlandırılan zaman aralıklarına ayrılarak saklanır. İstemci tarafında ise optimizasyona izin veren ve karar alma mekanizmasına esneklik tanıyan bir yapısı olması sebebiyle yaygın olarak kullanılmaktadır. Bu bağlamda, HTTP Üzerinden Dinamik Uyarlanabilir Video Akışı (DASH) standardındaki uygulamaların optimizasyon çalışmaları mobil internet trafiğini iyileştirme açısından önemli yer tutar. DASH standardında uyarlama algoritması üzerinde ve videoların kodlama detayları hakkında herhangi bir kısıtlama bulunmadığı için [1] yapılan çalışmalar bu iki alana odaklanmıştır.

İstemci tarafındaki optimum uyarlama algoritması tasarımı çalışmaları öncelikle kullanıcının algıladığı kaliteyi ifade eden deneyim kalitesinin (QoE) tanımlanmasına ihtiyaç duyar. Uyarlanabilir video akışı sırasında QoE'yi ifade eden en önemli kriterler video donma süresi, ortalama video kalitesi, video kalitesindeki veya bit hızındaki değişim sıklığı ve video oynatımı başlangıç gecikmesi olarak sıralanabilir [2]. Tasarlanan uyarlama algoritmaları bit hızı seviyelerinin diğer bir deyişle gösterimlerin seçimi sürecinde bu kriterleri öncelik derecelerine göre göz önünde bulundurduğu gibi anlık video arabellek (buffer) boyutu, anlık tahmin edilen bant genişliği gibi unsurları da kullanır. Bu çalışmalara örnek olarak, [3] çalışması, en yüksek önceliği arabellek boyutuna verip, ortalama video kalitesi ve bit hızındaki değişim sıklığı arasındaki dengeyi bit hızındaki değişim kararını belirleyecek üç adet arabellek eşiği belirleyerek çözmeye çalışmıştır. [4] çalışması arabellek boyutu yerine sadece bant genişliği kullanım oranını kullanarak bit hızındaki değişim miktarını seçmeye çalışmıştır. Öne sürdükleri yöntemde sürekli bir değişken olan bant genişliği kullanım oranından ayırık bir değişken olan bit hızı değişim miktarına geçişte bulanık mantık denetleyicisi kullanılmışlardır.

Benzer çalışmaların yanı sıra daha yakın tarihlerde özellikle DASH ile kullanımı noktasında depolama ve bit hızı değişikliği sırasındaki avantajları [5] sayesinde ölçeklenir video kodlama (SVC) [6] esaslı DASH çalışmalarında önemli bir eğilim gözlenmektedir. Ölçeklenir video kodlamada alt seviyedeki

video gösterimleri üst seviyedeki video gösterimlerinin oluşturulmasında kullanılır. Bu nedenle DASH ile birlikte kullanılırken gösterim seviyelerindeki değişimlerde ve depolanma sırasında daha etkin kaynak kullanımı sağlar. SVC kullanan DASH çalışmaları örnek olarak bant genişliğinden bağımsız olarak sadece anlık arabellek boyutuyla bir arabellek eşliğinden faydalanarak bit hızı belirlemesi yapan [7] verilebilir. Çalışmalarında video akış sürecini üç aşamaya ayırıp, ilk aşamada belirlenmiş eşğin altında sadece arabellekteki en üst katman ve altındaki katmanlardan video akışı sağlamışlardır. Bu eşik aşıldığında kalite değişim sıklığını düşürmek için eşik değerinin belirli bir miktar arttıran ikinci aşamadan sonra video kalitesi artırılmaktadır. [8] çalışmasında uyarılma döngüsü iki bölüme ayrılmıştır. İstenen arabellek boyutuna gelene kadar en alt seviyeden video segmentleri indirilirken, bu aşamadan sonra arabellekte bulunup üst gösterime geçilecek segmentler SSIM [9] metriğine göre önceliklendirilir. Karar alma mekanizmalarında hem arabellek boyutu hem de tahmin edilen bant genişliğinden yararlanan SVC tabanlı DASH uygulamalarına örnek olarak [10] ve [11] verilebilir. Bit hızı değişikliği için [10] çalışmasında ancak arabellek kısıtı sağlandığında tahmin edilen bant genişliğine bakılırken, [11] çalışmasında bant genişliği eşğinin aşılması veya tahmin edilen bant genişliğinin ilgili video gösterimi bit hızının üzerinde bulunma şartlarından sadece birinin sağlanması durumunda da üst gösterime geçiş yapılabilmektedir.

Her ne kadar çalışmalar istemci tarafında odaklanmış olsa da, sunucu tarafında da DASH uygulamalarını farklı yönleriyle ele alan çalışmalar mevcuttur. Sunucu tarafındaki SVC tabanlı DASH çalışmalarından biri olan [12] de bu alandaki uygulamalarda kullanılabilmesi adına SVC DASH veri seti ve benzer veri setlerinin oluşturmasına izin veren bir sistem tasarlamışlardır. Çalışmada videolar değişken bit hızı (VBR) seçeneği ile kodlanmıştır. VBR, aynı gösterim seviyesi farklı segmentlerde farklı bit hızlarına sahip olabilecek şekilde gerçekleştirilen kodlama şeklidir. Bit hızının tek bir gösterim için segmentler arasında değişiklik göstermediği kodlama şekli ise sabit bit hızı (CBR) ile ifade edilir. Literatürde DASH'in sunucu tarafındaki uygulamaları olarak ifade edebileceğimiz farklı kodlama şekillerinde veri setleri öne süren başka çalışmalar da mevcuttur [13] [14] [15]. Fakat CBR ve VBR seçeneklerinde kodlanmış DASH veri setlerinin SVC tabanlı DASH uygulamalarındaki avantaj ve dezavantajlarını ortaya koyan kapsamlı bir çalışma mevcut değildir.

Bu çalışmada SVC tabanlı DASH ile uyumlu şekilde sunucu tarafında tutulan CBR ve VBR modunda kodlanmış videoların farklı uyarılma algoritmaları kullanılarak gönderilmesi ile elde edilen sonuçlar değerlendirilmiş olup, bu iki kodlama şeklinin istemci tarafındaki uygulamadan bağımsız performansları ortaya konmaya çalışılmıştır. Çalışmanın kalan kısımlarının organizasyonu şu şekildedir. Bölüm II'de istemci tarafında önceki çalışmaların yapılan analiz sistemine entegre edilmesine, bölüm III'te elde edilen bulguların değerlendirilmesine ve bölüm IV'te sonuçlara yer verilmiştir.

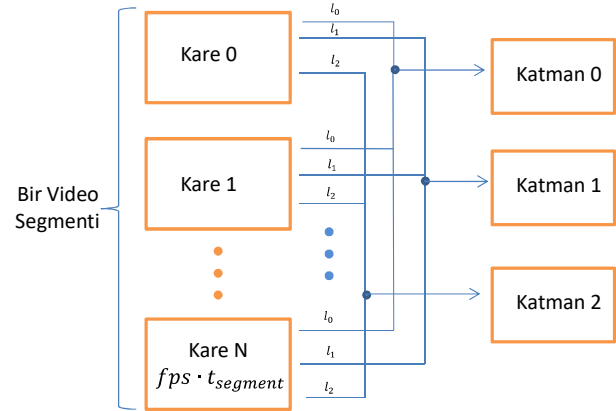
## II. YÖNTEM VE MATERYAL

### A. Veri Seti Hazırlığı

Veri seti sabit ve değişken bit hızlarında SVC standardında kodlanmış Big Buck Bunny [16], Tears of Steel [17] ve Sintel

[18] videolarından oluşmaktadır. Bu videolar Joint Scalable Video Model (JSVM) [19] ile kodlanmış olup CBR modunda 180 Kbit/s, 360 Kbit/s ve 720 Kbit/s bit hızında olmak üzere üç gösterim kullanılmıştır. VBR modu CBR'ye göre çok daha hızlı kodlanabildiği için pratik kullanımlarda daha fazla sayıda gösterim oluşturulmasına olanak tanır. Bu avantajını göz önünde bulundurarak VBR modu için 3 CGS ve 2 MGS katmanı olmak üzere ortalama 150 Kbit/s, 300 Kbit/s, 400 Kbit/s, 630 Kbit/s ve 850 Kbit/s bit hızlarına sahip 5 gösterim oluşturulmuştur.

SVC ile kodlanmış videolarda alt seviyedeki gösterimlerden üst seviyedeki gösterimler elde edilirken eklenen video bölümlerine katman adı verilir. SVC'deki katmanlar video bir bütün halindeyken video karelerine dağıtılmış durumda bulunur. Diğer bir yönden DASH ile gönderilecek video gösterimlerinin kareler yerine birkaç saniyelik segmentler bazında ayrı tutulması gerekmektedir. Bu nedenle SVC ile kodlanmış videolar için karelerdeki katman bilgilerini ayırma ve her bir segment için gruplama adına ihtiyaç duyulur. SVC katman çoklayıcı çözücü olarak ifade edebileceğimiz bu adımda katman bilgilerini ayırabilmek için katman başlangıçlarını gösteren Network Abstraction Unit (NALU) tanımlayıcı byte'ları ile video katman sayıları bilgilerini girdi olarak kullanan bir algoritma oluşturulmuştur.



Şekil 1 SVC Katman Çoklayıcı

Algoritma 1 içerisinde tüm video boyunca video karelerindeki katman bilgilerinin çıkartılıp  $V_{i,j}$  ile isimlendirilen video bölümlerinde gruplandırılması süreci, başlangıç video segmenti oluşturma gibi detaylara yer vermeden açıklanmaya çalışılmıştır. Daha sonradan tüm video bölümlerinin SSIM ölçümleri alınıp, video katman, segment ve bunların SSIM ve erişim bilgileri manifesto (MPD) dosyasına aktarılmıştır. Videoların DASH ile uyumlu hale getirilmesinden sonra istemci tarafındaki uygulamalara geçilmiştir.

### Algoritma 1 SVC katman çoklayıcı çözücü algoritması

#### Girdiler:

$N_i$ : Katman sayısı,  $N_{f,seg}$ : Segment içi frame sayısı,  
 $N_{gop}$ : görüntü grubu (GOP) boyutu  
 $N_i$ : NALU tanımlayıcı (0x00 0x00 0x00 0x01)

#### Geçici Değişkenler:

B: okunan byte dizisi,  $N_{f,toplam}$ : Toplam kare sayısı  
 $V_{2D}$ : 2 boyutlu NALU vektörü,  $V_{1D}$ : 1 boyutlu NALU vektörü  
 $\gamma$ : artik kare sayısını ifade eden değişken

#### Çıktılar:

$V_{i,j}$ : i katmanı ve j segmentine ait video dosyaları

#### while 1

b=okunan byte,  $b_4$ =okunan son 4 byte

if  $b_4=N_i$  then

extract  $N_i$  from B

append B to  $V_{1D}$

clear B

if  $(c_n) \bmod (N_i+1) = 0$  then

append  $V_{1D}$  to  $V_{2D}$

clear  $V_{1D}$  continue endif

elseif b=eof then

append  $V_{1D}$  to  $V_{2D}$

clear  $V_{1D}$  break endif

#### endwhile

$N_{f,toplam} \leftarrow V_{2D}$  satır sayısı

for i in  $N_i$

$V_{0,j}$ 'yi oluşturup anlık dosya olarak tanımla

for j in  $N_{f,toplam}$

if  $j \leq N_{f,seg}$  then

$N_{f,anlık} \leftarrow N_{f,seg} - N_{gop} + 1$

$\gamma \leftarrow 0$

else

$N_{f,anlık} \leftarrow N_{f,seg}$

$\gamma \leftarrow N_{f,anlık} - N_{gop} + 1$  endif

if  $(j) \bmod (N_{f,anlık}) = \gamma$  and  $j \geq N_{f,seg}$

$V_{i,j}$ 'yi oluşturup anlık dosya olarak tanımla endif

$N_i$ 'yi anlık dosyaya yaz

$V_{2D}[j][i]$ 'yi anlık dosyaya yaz

#### endfor

#### endfor

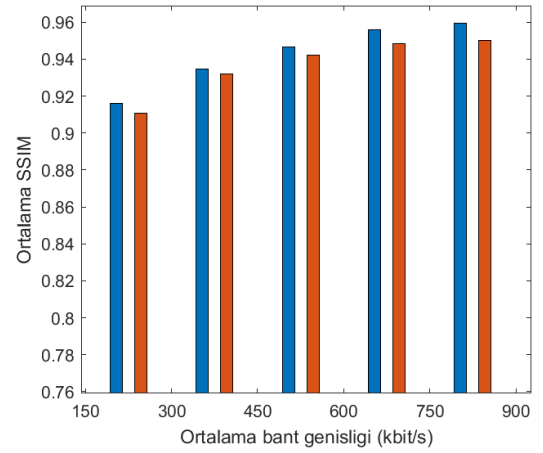
### B. İstemci Uyarlaması

Bu aşamada bölüm I'de bahsedilen istemci tarafıyla ilgili önerilmiş olan SVC tabanlı DASH algoritmalarından üçü [7] [8] [11] kurulan sisteme entegre edilmiştir. Öncelikle Linux işletim sistemli bir sunucuda tutulan CBR ve VBR ile kodlanmış olan videolar, yine Linux işletim sistemli bir istemciden istenerek gerçek zamanlı video akışı gerçekleştirilmiştir. Bu aşamada sunucu ve istemci arasındaki bant genişliğindeki değişimler planlanan zamanlarda çağırılan Linux tc komutlarından oluşan bir program yazılarak sağlanmıştır. Gerçek zamanlı video akışından sonra daha kapsamlı bir test gerçekleştirebilmek için hazırlanan video bölümleri bilgileri (SSIM, dosya boyutları, katman ve segment bilgileri) SVC DASH Simulatörüne [20] aktarılmıştır. Daha sonra, seçilen üç adet SVC tabanlı DASH çalışmalarında önerilen algoritmalar simülatöre entegre edilmiştir. Uyarlanan algoritmalarda en üst katman sayısının üstündeki dizi elemanlarına ulaşmaya çalışma koşulları veya belirli bir segment için önceki katmanlar indirilmeden üst katmana geçişe neden olabilecek özel koşullar gibi algoritmada değinilmeyen kısımlar ile simülatöre uyumluluk açısından gerekli bölümlerde algoritmaların genel yapısını bozmadan değişikliklere yer verildiğinin vurgulanması gerekmektedir. Uyarlamanın ardından üç algoritmanın CBR ve VBR modlarındaki veri seti

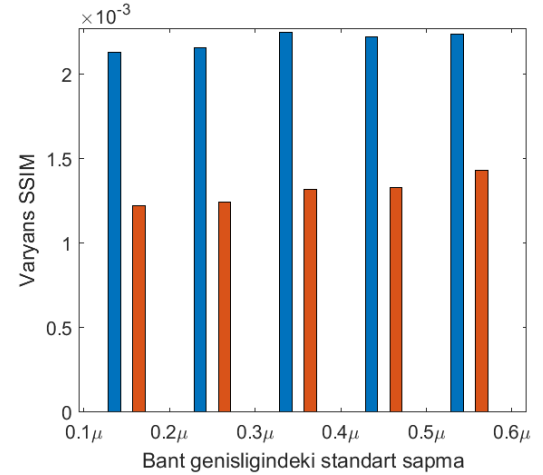
bölümlerinin ayrı ayrı performansları elde edilip tüm algoritmaların CBR ve VBR modlarındaki ortalama metrik değerleri hesaplanmıştır.

### III. BULGULAR

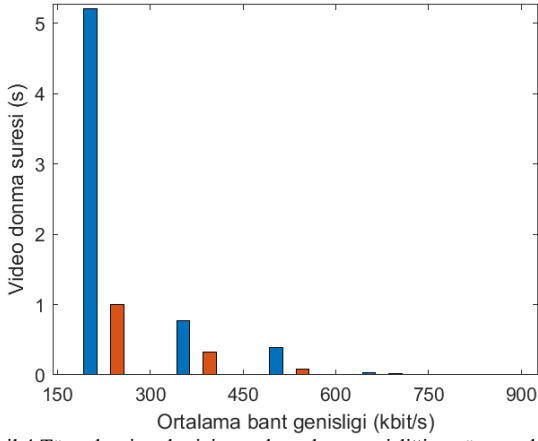
Bu bölümde VBR ve CBR modunda kodlanan videoların DASH ile uyumluluğu ve performans değerleri farklı yaklaşımlara sahip üç algoritmanın ortalama sonuçları ile birlikte sunulmuştur. Simülasyonlar 5 farklı ortalama, 5 farklı standart sapma ve 4 farklı durağanlık aralığından her biri için 10'ar adet bant genişliği dalga biçimi içeren veri seti ile gerçekleştirilmiştir. Simülasyonlarda video kalitesi SSIM metriği ile ifade edilip ortalama SSIM, varyans SSIM ve video donma sürelerinin yüksek bağımlılıkta buldukları bant genişliği karakterisiklerine göre grafikleri sırayla Şekil 1, Şekil 2, ve Şekil 3'te verilmiştir.



Şekil 2 Tüm algoritmalar için ortalama bant genişliğine göre ortalama SSIM sonuçları (mavi CBR modunu, kırmızı VBR modunu ifade eder)



Şekil 3 Tüm algoritmalar için bant genişliğindeki standart sapmaya göre SSIM varyansı sonuçları (mavi CBR modunu, kırmızı VBR modunu ifade eder)



Şekil 4 Tüm algoritmalar için ortalama bant genişliğine göre ortalama video donma süreleri sonuçları (mavi CBR modunu, kırmızı VBR modunu ifade eder)

Şekil 2’de görüldüğü gibi sabit bit hızında yapılan kodlama tüm ortalama bant genişlikleri için daha iyi performansa sahiptir. Bu duruma yol açan etmen olarak veri setindeki CBR katmanlarının benzer SSIM değerine karşılık gelen VBR katmanlarına göre daha düşük bit hızına sahip olması gösterilebilir. Şekil 3’te CBR’nin ortalama SSIM’deki avantajının aksine yüksek video kalite varyansı ile QoE’yi VBR’a kıyasla negatif etkilediği görülmektedir. Bu durumun sebebi ise hem düşük kodlama hızı nedeniyle az sayıda katmandan oluşması, hem de sabit bit hızını korumak için video içeriğindeki değişimlerden dolayı video kalitesinde aynı katman içinde yüksek değişimlere neden olması olarak düşünülebilir. Son olarak Şekil 4’te özellikle düşük ortalama bant genişliğine sahip dalga biçimlerinde CBR’nin çok daha yüksek video donma sürelerine neden olduğu görülmektedir. Yüksek donma süresi yine CBR’daki düşük katman sayısının bant genişliği düşmelerinde yeterli uyarılama esnekliğini sağlayamamasından kaynaklanmaktadır.

#### IV. SONUÇ

VBR ve CBR kodlama modlarının ölçeklenir videolar için DASH ile aktarımı sırasındaki performans değerlendirmeleri farklı uyarılama algoritmaları ve ağ koşulları için yapılmış olup VBR modunun video donma süreleri ve video kalite değişimi açısından önemli avantaja sahip olduğu görülmüştür. Her ne kadar CBR küçük bir farkla ortalama kalite açısından daha iyi sonuç verse de diğer iki QoE metriğindeki önemli fark ve video kodlama süresindeki avantajı VBR modunun ölçeklenir videolarda DASH ile kullanımını daha elverişli bulunmuştur.

#### KAYNAKLAR

[1] Stockhammer, T., “Dynamic adaptive streaming over HTTP-- standards and design principles,” in *Proceedings of the second annual ACM conference on Multimedia systems*, 2011, pp. 133-144.

[2] Kua, J., Armitage, G., and Branch, P., “A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1842-1866, 2017.

[3] Miller, K., Quacchio, E., Gennari, G., and Wolisz, A., “Adaptation algorithm for adaptive streaming over HTTP,” in *2012 19th international packet video workshop*, May. 2012, pp. 173-178.

[4] Xiong, P., Shen, J., Wang, Q., Jayasinghe, D., Li, J., and Pu, C., “NBS: A network-bandwidth-aware streaming version switcher for mobile streaming applications under fuzzy logic control,” in *2012 IEEE First International Conference on Mobile Services*, Jun. 2012, pp. 48-55.

[5] Kalva, H., Adzic, V., and Furht, B., “Comparing MPEG AVC and SVC for adaptive HTTP streaming,” in *2012 IEEE International Conference on Consumer Electronics (ICCE)*, Jan. 2012, pp. 158-159.

[6] Schwarz, H., Marpe, D., and Wiegand, T., “Overview of the scalable video coding extension of the H. 264/AVC standard,” *IEEE Transactions on circuits and systems for video technology*, vol. 17, no. 9, pp. 1103-1120, 2007.

[7] Sieber, C., Hoßfeld, T., Zinner, T., Tran-Gia, P., and Timmerer, C., “Implementation and user-centric comparison of a novel adaptation logic for DASH with SVC,” in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, May 2013, pp. 1318-1323.

[8] Çalı, M., and Özbek, N., “SSIM-based adaptation for DASH with SVC in mobile networks,” *Signal, Image and Video Processing*, pp. 1-8, 2020, doi: 10.1007/s11760-020-01646-y

[9] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600-612, 2004.

[10] Lekharu, A., Kumar, S., Sur, A., and Sarkar, A., “A qoe aware svc based client-side video adaptation algorithm for cellular networks,” in *Proceedings of the 19th International Conference on Distributed Computing and Networking*, Jan. 2018, pp. 1-4.

[11] Ozcan, S. G., Kivildim, T., Cetinkaya, C., and Sayit, M., “Rate adaptation algorithm with backward quality increasing property for SVC-DASH,” in *2017 IEEE 7th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*, Sep. 2017, pp. 24-28.

[12] Kreuzberger, C., Posch, D., and Hellwagner, H., “A scalable video coding dataset and toolchain for dynamic adaptive streaming over HTTP,” in *Proceedings of the 6th ACM Multimedia Systems Conference*, Mar. 2015, pp. 213-218.

[13] Zabrovskiy, A., Feldmann, C., & Timmerer, C., “Multi-codec DASH dataset,” in *Proceedings of the 9th ACM Multimedia Systems Conference*, Jun. 2018, pp. 438-443.

[14] Lederer, S., Mueller, C., Timmerer, C., Concolato, C., Le Feuvre, J., and Fliegel, K., “Distributed DASH dataset,” in *Proceedings of the 4th ACM Multimedia Systems Conference*, Feb. 2013, pp. 131-135.

[15] Le Feuvre, J., Thiesse, J. M., Parmentier, M., Raulet, M., and Daguet, C., “Ultra high definition HEVC DASH data set,” in *Proceedings of the 5th ACM Multimedia Systems Conference*, Mar. 2014, pp. 7-12.

[16] Blender Foundation, Big Buck Bunny. Accessed: Mar. 10, 2021, Available: <https://peach.blender.org/>

[17] Blender Foundation, Tears of Steel. Accessed: Mar. 10, 2021, Available: <https://mango.blender.org/>

[18] Blender Foundation, Sintel. Accessed: Mar. 10, 2021, Available: <https://durian.blender.org/>

[19] Reichel, J., Schwarz, H. and Wien, M., “Joint scalable video model 11 (JSVM 11)”. Joint Video Team, 2007

[20] Cali, M., and Ozbek, N.: Dash-Simulator. Accessed: April. 20, 2021, Available: <https://github.com/mehmet-cali/dash-simulator>