



T.R.
EGE UNIVERSITY
Graduate School of Applied and Natural Science



SDN Aware DASH Client Architecture

MSc Thesis

Simge Gizem ÖZCAN

International Computer Institute
Information Technology

İzmir
2019

T.R.
EGE UNIVERSITY
Graduate School of Applied and Natural Science

SDN Aware DASH Client Architecture



Simge Gizem ÖZCAN

Supervisor: Assoc. Prof. Dr. Müge SAYIT

International Computer Institute
Information Technology

İzmir
2019

The study titled as "SDN Aware DASH Client Architecture", which was submitted by Simge Gizem ÖZCAN as a master's thesis, was evaluated by defending it in accordance with the relevant provisions of EU Graduate Education Regulation and Institute of Natural and Applied Sciences Education and Instruction Directive. majority of unanimous votes. the defense was found worthy of defense and the dissertation on 28/08/2019 was found to be successful with the majority of unanimous votes.

Juri Members:

Juri President : Assoc. Prof. Dr. Müge SAYIT

Reporter Member : Prof. Dr. Bahar KARAOĞLAN

Member : Assist. Prof. Dr. Korhan KARABULUT

Signature



**EGE UNIVERSITY GRADUATE SCHOOL OF NATURAL AND
APPLIED SCIENCE**

DECLARATION OF CONFORMITY TO ETHICS

In accordance with the relevant provisions of the EU Regulations for Graduate Education, I submitted this master thesis entitled as "**SDN Aware DASH Client Architecture**", which I my own work, and I declare that all the results, documents, and information belongs to me, and documents that I have presented in the context and the information that I have referred to, are fully listed in the list of references, and that I have not infringed any patent or copyrights during the dissertation and writing, and that I have not submitted any part of this dissertation in another dissertation at this university or another university. I hereby declare that I have acted in accordance with the rules of scientific ethics at all stages from planning to the writing of this thesis and that I will accept any legal consequences in the event of the contrary.

28/08/2019

Simge Gizem ÖZCAN



ABSTRACT**SDN Aware DASH Client Architecture**

ÖZCAN, Simge Gizem

MSc in International Computer Institute

Supervisor: Assoc. Prof. Dr. Müge SAYIT

Aug 2019, 58 pages

Dynamic adaptive video streaming applications are one of the most popular applications of the last decade. In such applications, the basic architecture allows users to change quality as they desire. Evaluating Quality of Experience (QoE) effectively for Transmission Control Protocol (TCP) video streaming with network performance parameters is a challenging task. For this reason, the use of flexible systems such as Software Defined Network (SDN) to adapt to variable network conditions has increased, rather than traditional methods, in the development of these applications. Due to the varying conditions of the network, the use of multi-path and methods to improve downloaded segments provide great advantages for a stable and high-quality monitoring experience. Using multiple paths maintains the stability of the system where segment sizes differ or where short-term bandwidth decreases. Hence, using Scalable Video Codec (SVC) to improve the downloaded segments prevents bandwidth wastage because segment quality is provided by decoding layers.

In this thesis, we propose a QoE aware adaptive streaming over Hypertext Transfer Protocol (HTTP), named “mpBack²”, that requests segments of an SVC coded video content by using multiple paths by connecting an SDN Controller through OpenFlow protocol. Simulation results show that users achieve higher QoE by using our approach when compared to conventional rate adaptation algorithms.

Keywords: Video Streaming over HTTP, Scalable Video Coding, Multiple Path Rate Adaptation, Software Defined Networks, Quality of Experience

ÖZET**Yazılım Tanımlı Ağ Teknolojisi Farkındalıklı DASH İstemci Mimarisi**

ÖZCAN, Simge Gizem

Yüksek Lisans Tezi, Uluslararası Bilgisayar Enstitüsü Anabilim Dalı

Tez Danışmanı: Doç. Dr. Müge SAYIT

Ağustos 2019, 58 sayfa

Dinamik uyarlamalı video akışı uygulamaları, son on yılın en popüler uygulamaları arasındadır. Bu tür uygulamalarda, temel mimari kullanıcıların kaliteyi istedikleri gibi değiştirmelerine izin verir. Değişken ağ koşullarında Aktarım Kontrol Protokolü (TCP) video akışlandırma için Deneyim Kalitesini (QoE) arttırmak zorlu bir iştir. Bu nedenle bu uygulamaların geliştirilmesinde geleneksel yöntemlerden ziyade, Yazılım Tanımlı Ağ (SDN) gibi esnek sistemlerin kullanımı artmaktadır. İndirilen video segmentlerini iyileştirmek için birden fazla ağ yolu kullanılması, istikrarlı ve yüksek kaliteli bir izleme deneyimi için büyük avantajlar sağlar. Birden çok ağ yolu kullanılması, segment boyutlarının farklı olduğu veya kısa süreli bant genişliğinin azaldığı durumlarda sistemin kararlılığını korur. Bu nedenle, indirilen segmentleri iyileştirmek için Ölçeklenebilir Video Kodlama (SVC) kullanmak bant genişliği israfını önler, çünkü segment kalitesi, kalite katmanlarının birleştirilmesi ile sağlanır.

Bu çalışmada, bir SDN Kontrolcüsü olan OpenFlow protokolü aracılığıyla birden fazla ağ yolu kullanarak SVC ile kodlanmış video içeriğinin segmentlerini isteyen ve Köprü Metni Aktarım Protokolü (HTTP) üzerinden uyarlamalı akış sağlayan, “mpBack²” adını verdiğimiz QoE farkındalıklı bir hız adaptasyon algoritması, bulunmaktadır. Simülasyon sonuçları, bu algoritmanın geleneksel hız uyarlama algoritmalarına kıyasla daha yüksek QoE elde ettiğini göstermektedir.

Anahtar Sözcükler: Çoklu Ağ Yollu Hız Adaptasyon, Yazılım Tanımlı Ağlar, Kullanıcı Deneyim Kalitesi, HTTP Video Akışlandırma, Ölçeklenebilir Video Kodlama

FORWARD

This thesis is written as completion to the master degree International Computer Science, at the Ege University Izmir. During my study, I have dedicated a considerable amount of time and effort on multimedia and Software Defined Networks. The thesis is about dynamic adaptive video streaming in variable network conditions over the internet and the creation of a rate adaptation algorithm with the best user experience quality.

Each research and analysis conducted during my graduate studies has enabled the creation of meaningfully applicable scientific contributions in this thesis. The knowledge and experience I have gained in the process of preparing this study has contributed to me on this path in which I have pursued my goals and will continue to do so.

IZMIR

28/08/2019

Simge Gizem ÖZCAN

CONTENTS

	<u>Page</u>
ACCEPTENCE AND APPROVAL	iii
DECLARATION OF CONFORMITY TO ETHICS	v
ABSTRACT	vii
ÖZET	ix
FORWARD	xi
CONTENTS	xiii
CONTENTS (more)	xiv
FIGURES INDEX	xv
CHART INDEX	xvii
CHART INDEX (more)	xviii
TABLE INDEX	xix
INDEX OF ICONS AND ABBREVIATIONS	xxi
1. INTRODUCTION	1
2. BACKGROUND AND RELATED WORKS	3
2.1 HTTP DASH	3
2.2 SVC DASH	6
2.3 SDN	7
2.4 QoE	10
3. PROPOSED APPROACH	12
3.1 Rate adaptation algorithm with backward quality increasing	12
3.2 An MCDM method for the selection of layers	12
3.3 General Architecture	13
3.4 Improvement of the Dynamic Rate Adaptation Algorithm	15
3.4.1 Initialization	17
3.4.2 Candidate segment group selection	23
3.4.3 Segment selection and path assignments	26
3.4.4 Backup algorithm	28
4. PERFORMANCE EVALUATION	31
4.1 Simulation Setup	31
4.1.1 Mininet	31
4.1.2 Floodlight Controller	32

CONTENTS (more)

	<u>Page</u>
4.1.3 Network conditions	32
4.1.4 Video contents	32
4.1.5 Configurations	34
4.2 QoE Calculation	34
4.3 Performance Results	36
4.3.1 Single client simulations	36
4.3.2 Multiple client simulations	40
5. CONCLUSIONS	45
REFERENCES	48
REFERENCES (more).....	49
REFERENCES (more).....	50
REFERENCES (more).....	51
REFERENCES (more).....	52
REFERENCES (more).....	53
REFERENCES (more).....	54
REFERENCES (more).....	55
REFERENCES (more).....	56
ACKNOWLEDGE	57
RESUME	58

FIGURES INDEX

<u>Figure</u>	<u>Page</u>
2.1 MPD File content	3
2.2 HTTP Adaptive Streaming Architecture	4
2.3 Video Encoding/Decoding Process with SVC	6
2.4 The architectural difference between traditional and software defined networks.....	8
3.1 General Architecture of the Proposed System	13
3.2 The modules running on the client side	14
3.3 Rate Adaptation Flow Chart	15
3.4 FSM of rate adaptation algorithm	19
3.5 Phases and Zones Regarding Buffer Occupancy	20
3.6a Buffer during candidate segment selection for Steady Phase With same rep_{max}	24
3.6b Buffer during candidate segment selection for Steady Phase With decreased rep_{max}	24
3.7a Buffer during candidate segment selection for Greedy Phase With decreased rep_{max}	26
3.7b Buffer during candidate segment selection for Greedy Phase With increased rep_{max}	26
3.8 The relation between input/output rate and buffer occupancy	27
3.9 Buffer during segment selection for Backup Algorithm	29

CHART INDEX

<u>Chart</u>	<u>Page</u>
1.1 Cisco VNI Global IP Video Traffic Forecast, 2017 – 2020	1
4.1a Segment Bitrates by Video Content for Base Layer.....	33
4.1b Segment Bitrates by Video Content for Enhancement Layer 1.....	33
4.1c Segment Bitrates by Video Content for Enhancement Layer 2.....	33
4.1d Segment Bitrates by Video Content for Enhancement Layer 3	33
4.2a Average Initial Waiting Times of Single Client Simulations for Big Buck Bunny	36
4.2b Average Initial Waiting Times of Single Client Simulations for Elephants Dream	36
4.2c Average Initial Waiting Times of Single Client Simulations for Tears of Steel	36
4.3a Average Representation Bitrates of Single Client Simulations for Big Buck Bunny	37
4.3b Average Representation Bitrates of Single Client Simulations for Elephants Dream	37
4.3c Average Representation Bitrates of Single Client Simulations for Tears of Steel	37
4.4a Average Representation Changes of Single Client Simulations for Big Buck Bunny	37
4.4b Average Representation Changes of Single Client Simulations for Elephants Dream	37
4.4c Average Representation Changes of Single Client Simulations for Tears of Steel	37
4.5a Average Watched Segments of Single Client Simulations for Big Buck Bunny	38
4.5b Average Watched Segments of Single Client Simulations for Elephants Dream	38
4.5c Average Watched Segments of Single Client Simulations for Tears of Steel	38
4.6a Average QoE of Single Client Simulations for Big Buck Bunny	40
4.6b Average QoE of Single Client Simulations for Elephants Dream	40

CHART INDEX (more)

<u>Chart</u>	<u>Page</u>
4.6c Average QoE of Single Client Simulations for Tears of Steel	40
4.7a Average Initial Waiting Times of Multiple Client Simulations for Big Buck Bunny	40
4.7b Average Initial Waiting Times of Multiple Client Simulations for Elephants Dream	40
4.7c Average Initial Waiting Times of Multiple Client Simulations for Tears of Steel	40
4.8a Average Representation Bitrates of Multiple Client Simulations for Big Buck Bunny	41
4.8b Average Representation Bitrates of Multiple Client Simulations for Elephants Dream	41
4.8c Average Representation Bitrates of Multiple Client Simulations for Tears of Steel	41
4.9a Average Representation Changes of Multiple Client Simulations for Big Buck Bunny	41
4.9b Average Representation Changes of Multiple Client Simulations for Elephants Dream	41
4.9c Average Representation Changes of Multiple Client Simulations for Tears of Steel	41
4.10a Average Watched Segments of Multiple Client Simulations for Big Buck Bunny	42
4.10b Average Watched Segments of Multiple Client Simulations for Elephants Dream	42
4.10c Average Watched Segments of Multiple Client Simulations for Tears of Steel	42
4.11a Average QoE of Multiple Client Simulations for Big Buck Bunny	44
4.11b Average QoE of Multiple Client Simulations for Elephants Dream	44
4.11c Average QoE of Multiple Client Simulations for Elephants Dream	44

TABLE INDEX

<u>Table</u>	<u>Page</u>
3.1 Algorithm and FSM Variables	17
4.1 Min,Max And Average Bandwidth Values (Mbps) Of The End-To-End Paths with Max Available Bandwidth In Each Scenario	32
4.2 Specifications of Video Contents	33
4.3 Standard Deviations of Video Contents.....	34
4.4 Configuration Items	34
4.5 Weights of the QoE Metrics	35
4.6 Buffer Outage Counts and Durations of Single Client Simulations for Big Buck Bunny	39
4.7 Buffer Outage Counts and Durations of Single Client Simulations for Elephants Dream	39
4.8 Buffer Outage Counts and Durations of Single Client Simulations for Tears of Steel	39
4.9 Buffer Outage Counts and Durations of Multiple Client Simulations for Big Buck Bunny	43
4.10 Buffer Outage Counts and Durations of Multiple Client Simulations for Elephants Dream	43
4.11 Buffer Outage Counts and Durations of Multiple Client Simulations for Tears of Steel	43



INDEX OF ICONS AND ABBREVIATIONS

<u>Abbreviations</u>	<u>Explanation</u>
QoE	Quality of Experience
TCP	Transmission Control Protocol
SDN	Software Defined Network
SVC	Scalable Video Codec
HTTP	HyperText Transfer Protocol
DASH	Dynamic Adaptive Streaming over HTTP
MPEG	Motion Picture Experts Group
QoS	Quality of Service
MPD	Media Presentation Description
API	Application Program Interface
MCDM	Multi-Criteria Decision Making
COPRAS	Complex Proportional Assessment
AHP	Analytic Hierarchy Process
IP	Internet Protocol
Kbps	Kilobit per second
KBps	Kilobyte per second
Mbps	Megabit per second

1. INTRODUCTION

Internet videos constitute a major share of today's internet traffic. Cisco reports (Cisco Visual Networking Index, 2019) indicate that IP video traffic will rise to 82% by 2022 compared to 2017 (Figure 1.1). HTTP is one of the most preferred streaming protocols for presenting video content on the Web due to the reliability of TCP, the HTTP web-cache infrastructure, and its congestion avoidance algorithm. Youtube, Microsoft Silverlight, Apple HLS, Adobe OSMF, Netflix (Adhikari et al., 2012) are among the most popular clients that use this protocol and increase video traffic the most (Akhshabi, Begen, and Dovrolis, 2011; Jiang, Sekar, and Zhang, 2014). Such video providers must meet user-centric expectations, such as QoE, to avoid user losses.

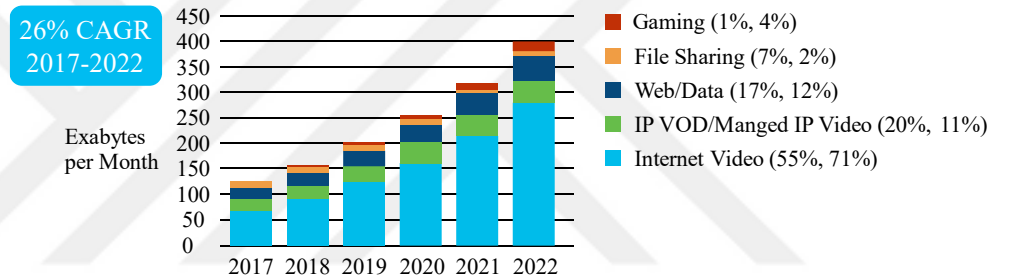


Chart 1.1 Cisco VNI Global IP Video Traffic Forecast, 2017 - 2020

Almost all suggested solutions to increase network capacity and improve user experience follow the same approach as chunk-based HTTP streaming. Media content is encoded according to quality and bitrate. Dynamic Adaptive Streaming over HTTP (DASH) (Stockhammer, 2011), an MPEG (Rainer et al., 2012) standard, provides uninterrupted video streaming with limited available bandwidth. In DASH applications, video is encoded at different quality levels. These qualities are divided into small parts called segments. This approach allows the client to make quality changes depending on the network conditions, dynamically determine the playback buffer size, and at the same time improve the performance. In SVC Based DASH, segments are also divided into layers. There is a base layer for each segment, and quality upgrades are provided by the enhancement layers.

SDN helps to create intelligent, flexible and programmable networks. Using SDN, it is possible to determine the quality of video segments that can be requested by utilizing the knowledge about network and user related information such as connection points, number of online users (Ramakrishna, Fernandes, and Karunakar, 2017). Using SVC enables rapid and flexible adaptation to variable network conditions (Politis et al., 2017). In (Cetinkaya, Ozveren, and Sayit, 2015) all video layers are treated as a different stream. According to the number of enhancement layers to be requested, the bitrate of these layers, and the existing path capacities; the server, client and SDN controller are communicated and the path is assigned by the controller. In (Xue et al., 2015), resources are used as much as possible and for this purpose, server and path redirections are made through Open Flow in the network where multiple clients and servers are located. Large companies like Google and Microsoft also use SDN for their data centers (Anon n.d., 2019).

Using the SDN technology and SVC, the proposed rate adaptation algorithm named “mpBack²”, which takes into account parameters such as network conditions, available bandwidth in available paths, client buffer occupancy, layer bitrates; and downloads appropriate segments from appropriate paths. It aims to optimize the advantages of SVC and SDN while minimizing video pauses and quality changes while maximizing user experience.

2. BACKGROUND AND RELATED WORKS

In this section, concepts and the studies in the literature related to HTTP DASH, SVC DASH, SDN, and QoE are given.

2.1 HTTP DASH

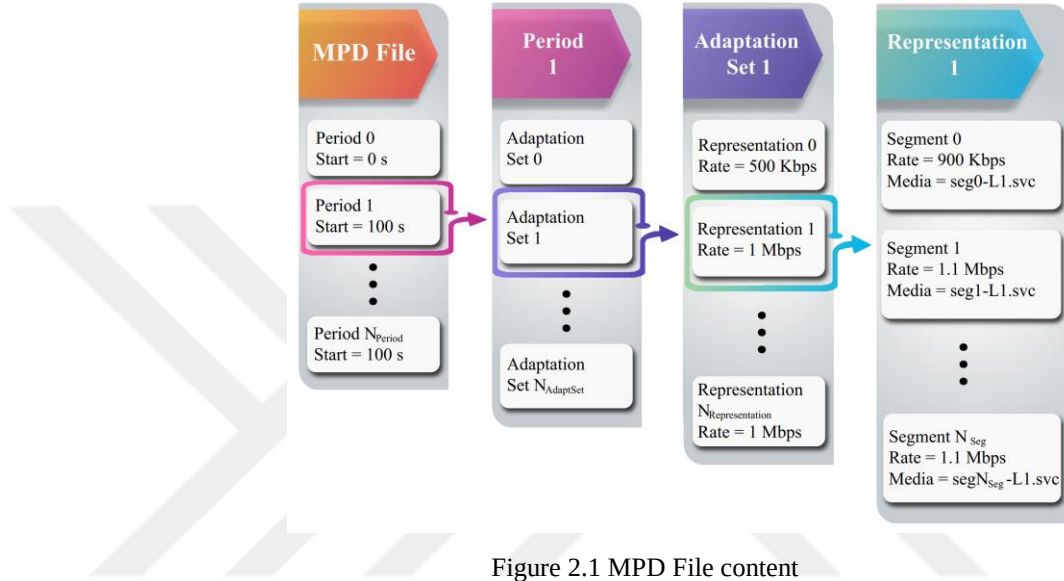


Figure 2.1 MPD File content

The main challenge with HTTP Adaptive Streaming systems is how to adjust video quality according to network changes for streaming video that provides a high-quality experience without interruption. The web server usually has little knowledge of the client/network status and these systems are client-driven, i.e. all control is on the client side. Therefore, a client must be prepared in advance or by the content/service providers. The Media Presentation Description file (MPD) contains multiple segment groups of different qualities held for each video on the server. Figure 2.1 shows the visualization of the MPD file. This file contains information about the video such as encoding type, segment duration and segment size, and average bitrates of representations. Segments are independent of each other because they start with a key frame. Unlike traditional video streaming, DASH selects the highest quality that can be achieved instead of directly controlling the video transmission rate. Different quality options are available for each segment. In this way, the client may require different segments in different bitrates according to varying network conditions such as bandwidth fluctuations and can

create a balance between high quality and buffer stability. If the size of the segments is small in terms of duration of the video, the system can quickly adapt to bandwidth changes.

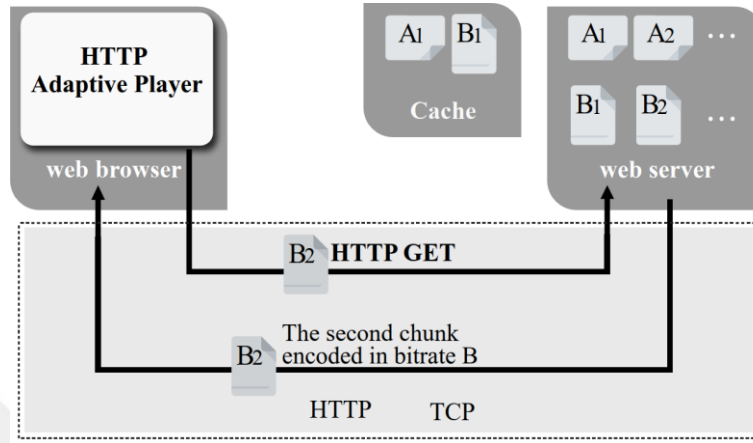


Figure 2.2 HTTP Adaptive Streaming Architecture

Video services using DASH pre-download segments and plays downloaded ones continuously. Because the content is pre-fetched using HTTP, there is no packet loss and therefore there is no need to provide error flexibility in the bit stream. After the TCP (Kuschnig et al., 2010, 2011) connection to the server is established, the client requests the MPD file in XML format and starts using the information in this file to request segments according to the rate adaptation algorithm and certain criteria. In Figure 2.2, the HTTP adaptive streaming architecture is given simply. In the scenario illustrated in the figure, the client sends a request to get second chunk, i.e. segment belonging the representation encoded in bitrate B. Similar to this scenario, the clients continue to get segments by sending requests, one after another. The segments are played sequentially and there is no video stalls during playback because they are sequential. The buffer occupancy rate, available bandwidth calculations and quality changes as the results in parallel with these calculations are made by the client. The client does not start video playback until a certain number of segments is downloaded at start-up. It fills the buffer at the initial waiting of a few seconds and requests segments as low as possible to shorten this time (Akhshabi et al., 2012). When playing video, this allows minimizing video pauses because future segments are available in the buffer (Seufert et al., 2015).

There are several rate adaptation algorithms in the literature for dynamically adaptive HTTP streaming systems. These algorithms are classified based on bandwidth, buffer, and hybrid bandwidth. Where bandwidth is taken into account, the client measures the bandwidth when downloading a segment and the quality appropriate for the next segment is calculated according to the bandwidth based rate adaptation algorithm. The client can select the next segment quality based on instant (Rainer et al., 2012) or pre-calculated average bandwidth (Cofano et al., 2017). In (Jiang, Sekar, and Zhang, 2014), the bandwidth estimation method which is stable against short-term bandwidth fluctuations and which reacts quickly to long-term fluctuations is studied. (Thang et al., 2012) checks the client and server side, then estimates the bandwidth and uses it to fix the system. In (Cicco and Mascolo, 2014), a bandwidth based algorithm focused on fairness, efficiency and stability were studied. Bandwidth based approaches can dynamically adapt to available bandwidth; however, they often do not achieve the quality allowed by the available bandwidth (Ozcan et al., 2017).

In addition to these studies, (Huang et al., 2014) shows that rate adaptation algorithms based on local buffer status are a viable solution. In buffer based algorithms, an interval is determined according to the buffer size and the buffer occupancy rate is kept in this range (Huang et al., 2014, 2013, 2018; Le et al., 2013). Hybrid approaches take into account both measured bandwidth and buffer occupancy rate (Petrangeli et al., 2015; Yin, 2015; Zhou 2014). In (Zou, 2015), the buffer is divided into three sections: safe, risky and temporary. To determine the next segment quality, these values were taken into account along with the estimated bandwidth.

Youtube has the ability to remove low-quality segments from the buffer according to bandwidth conditions and request high quality of the same segment numbers instead. When the necessary conditions are met, it is possible to download the same segment numbers in more than one quality. In (Ozcan et al., 2017; Mondal, 2017), this approach has been reported as unnecessarily waste of bandwidth. There are also some optimization efforts to minimize the amount of wasted bandwidth when assigning low quality segments (Mondal, 2017).

2.2 SVC DASH

SVC; allows users with different bandwidths, device capabilities, and user demands to watch the same video, even at different qualities, according to their own connection speed (Lu, 2013). (Sánchez, 2011; Famaey, 2013) provides the advantages of SVC in MPEG-DASH applications.

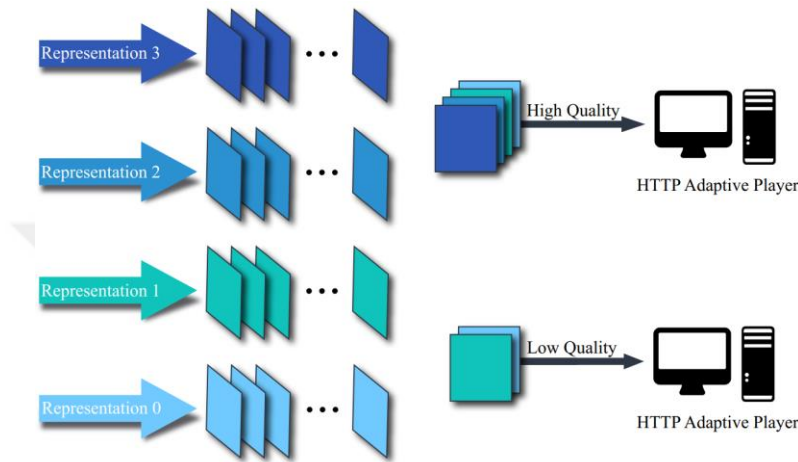


Figure 2.3 Video Encoding/Decoding Process with SVC

SVC standardizes the encoding of a particular video bitstream divided into small segment groups. It reduces the amount of content redundancy by converting each quality level into bytes by using the encoded video blocks from the lower quality of the segment. As shown in Figure 2.3, SVC has different segment groups for each quality level. It creates a desired number of enhancement layers in desired properties to compound with a base layer for each segment number. In order to be able to decode the x^{th} enhancement layer of a segment, the base layer of that segment and all layers up to the x^{th} layer must be downloaded. The requests for these layers are independent of each other. Even if the enhancement layers for a segment number have not been completely downloaded, video playback can continue from lower quality downloads. This situation causes the base layer to be given a higher priority than the enhancement layers and reduces the risk of outages (Seufert et al., 2015). All enhancement layers taken after the base layer are intended to improve observed video quality and there is no unnecessary repetitive information between layers (Schwarz et al., 2007). However, the fact that segments are structured increases the HTTP message load by requiring multiple HTTP

requests (Kalva et al., 2012). In (Zhao et al., 2015), this communication burden is mitigated and the cost is reduced by combining small sized layers.

Rate adaptation algorithms developed for non-scalable video encoding HAS systems can also be used for SVC-DASH. In addition, there are some studies for dynamically adaptable HTTP streaming systems based on scalable encoded video. In (Sánchez et al., 2011), the caching performance gain provided by SVC is examined.

In some studies, the next segment to be downloaded is selected based on the priorities of the base and enhancement layers. In (Andelin et al., 2012), the client chooses to download a enhancement layer of segment numbers with some layers downloaded, or a layer from a segment number that has not yet received any layers. If the number of segments in the buffer reaches a certain threshold, clients can also download the enhancement layers before the base layer is downloaded (Hosseini, 2016; Figueroa, 2016; Thenmozhi, 2017). In (Hwang, 2016), after the DASH client sends request messages to the server for base layers of some segment numbers, it also requests to download the enhancement layers for those segments. The fact that layers of a segment number can be retrieved separately also enables the quality of segments that have not been played back yet to be improved (Ozcan and Sayit, 2019). In (Ozcan and Sayit, 2019), the priorities of the candidate segments to be downloaded are determined by Multi-Criteria Decision Making (MCDM) methods and it is decided to take segments above a certain level of importance by using machine learning. Having a scaled structure allows SVC to be used in systems that host multiple paths or servers (Gohar et al., 2019).

2.3 SDN

In the traditional approach, routers forward packets using the embedded control unit which determines the routes. In other words, the control unit is embedded in forwarding devices in traditional networks. Figure 2.4 shows the difference between traditional and SDN approach, where the control is decoupled from switching entities and move to an external device called controller. SDN; is an approach that can be used to minimize the negative effects of variable network

conditions on QoE, and can be applied for flexible and rapid network adaptation. The SDN controller has network topology information such as the amount of traffic on the connections and the packet loss rate. It changes network behavior by making routing decisions based on flows and more attributes rather than traditional determination based decisions. It allows configurations to structure and manage network devices. It also provides solutions for many problems that cannot be solved by traditional network methods and architectures. With this architecture, network control can be programmed directly. Thus, different routing algorithms can be developed according to application needs by separating network control and routing functions for infrastructure, applications, and network services. Therefore, it can be used in many studies to improve service quality (QoS) (Tomovic et al., 2014; Karakus et al., 2017; Egilmez et al., 2013).

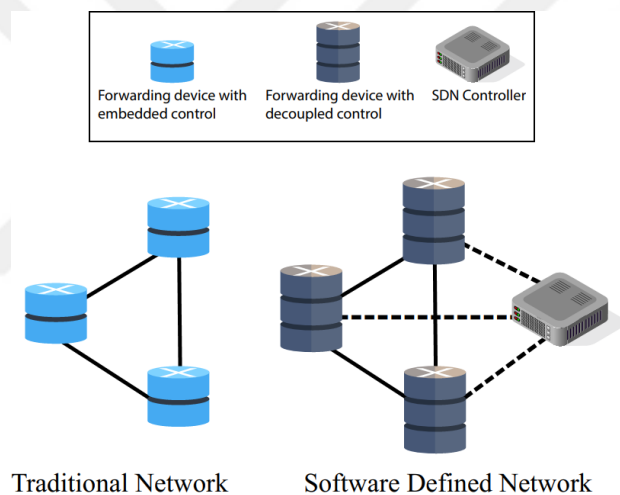


Figure 2.4 The architectural difference between traditional and software defined networks

In (Ramakrishnan et al., 2015), the bandwidth allocation algorithm is formulated over SDN, taking into account the client's buffer status and the complexity of the desired segments. (Xu et al., 2018) introduces a new hybrid design that combines traditional switching and SDN switching for both scalability and optimal performance.

SDN can assign one or more path (Elgabli et al., 2018) between servers and clients, by considering criteria such as traffic, bandwidth of the network where the servers and clients are located. It can redirect packets to a different path in case of communication failure on the default path (Thenmozhi et al., 2017). In (Laga et al.,

2014), it is recommended that the base and enhancement layers of SVC encoded video to be downloaded over the different network paths using SDN technology. In this study, the path with the smallest load is assigned to route the base layer packets. Like base layer path assignments, paths specified for enhancement layers are made according to layer priority. Enhancement layer packets with higher priority are sent over a smaller payload path. These paths can be determined based on the output of an optimization model, taking into account packet loss and delay variation (Egilmez et al., 2013).

In DASH systems, the user experience can be increased by using multiple servers (Zhang et al., 2015) or paths (Han et al., 2016), and existing resources can be used more effectively. In this way, it is possible to distribute more stable and fair resources to the users on the network. Where bandwidth is used to determine the quality of segment selections to be taken, bandwidth calculations are very important for a rate adaptation algorithm. If there is more than one path used for transferring the packets, a separate bandwidth can be calculated for each path by using the size of a segment, the download time and the path information in which the segment is downloaded. In (Kim and Chung, 2018), for a network where bandwidth fluctuations are high, the bandwidth of each path is calculated as aggregated and the fluctuations are reduced using a softening function that takes into account the calculated bandwidth differences of each path. In (Jmal and Chaari Fourati, 2019), the available bandwidth is measured by the SDN controller using a function that considers path capacities and downloaded segments.

The open network foundation forms the routing plane of switches and routers on the network (Shamim et al., 2018). It is the first protocol that enables communication between an SDN controller and switches. Depending on the rules loaded by the controller; it can act as a router, switch, firewall, or can perform the other roles. Switches replace the transmission tables with messages received by the SDN controller via this protocol. OpenFlow allows the controller to communicate with multiple devices and different hardware. It also allows checking the transmission logic of packets and can decide how to put packet rules into a hardware abstraction (Kreutz et al., 2015). Control platforms such as Floodlight (Floodlight,

2019), POX (Kleinrouweler et al., 2016), Beacon, OpenDayLight, Ryu which use the OpenFlow protocol, are used in studies on (Belkadi and Laaziz, 2017) SDN.

2.4 QoE

QoE refers to the quality perceived by the user based on subjective criteria. It creates a quality rating for the entire service. As a measure of end-to-end performance at the service level from the user's point of view, QoE is important for the design of systems and engineering processes. When designing systems, the expected QoE is often considered a system output metric and optimization objective.

The main purpose of a rate adaptation algorithm is to select and download segments to achieve the highest possible QoE value according to the network conditions in which it is located. In (Seufert et al., 2015; Bouten et al., 2012; Oyman and Singh, 2012; Garcia et al., 2014; Xue et al., 2014), the relationship between QoE Metrics and Adaptive HTTP streaming is examined. Initial waiting time, quantity of quality changes, outages, average monitoring quality are among the QoE metrics measured to determine overall quality (Sieber et al., 2013). Segment sizes, buffer size, and available bandwidth are highly effective in observing these metrics (Seufert et al., 2015). The factors affecting QoE have been identified in (Shi et al., 2015). It also examined how different metrics collected from real-time monitoring logs affect user experience in DASH sessions. In (Xiang et al., 2012), a flow strategy that can provide optimal results for QoE is solved using the Markov Decision Process.

The effect of QoE metrics on rate adaptation systems is detailed below.

Initial Waiting Time: It is the time taken until a certain number of segments are downloaded at the beginning of the video. This time varies according to factors such as segment size, available bandwidth, encoding settings. The initial waiting time less than 2.5 seconds does not affect QoE (Sieber et al., 2013). Therefore, this period should be kept as low as possible. This initial process also makes the system more stable against short-term bandwidth fluctuations (Seufert et al., 2015).

Average Bitrate: It gives the average value of the different qualities monitored in a video session. Generally, raising representation as much as possible is not equivalent to raising QoE (Garcia et al., 2014).

Representation Changes: In order to adapt to variable networks, it is inevitable that the monitored quality changes in a video session. It appears that users prefer a gradual quality decline during quality declines rather than shifting from one to the lowest quality (Mok et al., 2012).

Outage Count and Duration: Video outages occur because there are no segments left in the buffer to play. In HTTP DASH systems, this is often caused by instability in segment sizes or by downloading segments of higher quality than available bandwidth can provide. Research has shown that users prefer a long outage instead of multiple ones. Besides, they prefer quality changes and prolongation of initial waiting time instead of video outages (Seufert et al., 2015).

3. PROPOSED APPROACH

In this part of the thesis, previous version of the study, the general architecture of the proposed algorithm and the methodology of segment selection are explained.

3.1 Rate Adaptation Algorithm With Backward Quality Increasing

In the first version of our study, using a heuristic approach, the focus was on improving the already downloaded segments. Buffer based threshold was determined in the system and the quality allowed by bandwidth could be increased according to this threshold. In this study, segments are taken one by one, a single path is used, and segment selections are made only vertically. This algorithm has been shown to have a major effect on video pauses and smooth representation transitions.

3.2 An MCDM Method For The Segment Selection

In the second study, it is decided to request a group of segments at a time and the segment group to be downloaded is determined by Analytic Hierarchy Process (AHP) and COPRAS which is an MCDM method. Furthermore, SDN technology is also utilized for providing assistance to DASH clients. A priority ratio is calculated for all segments in the group regarding pre-defined criteria such as quality of the candidate segment and difference between segment number and playout index. The candidate segment group was kept wide, but only the segments with priority over a value were requested according to the priorities of the candidate segments. This priority limit value, which plays a major role in whether or not to request segments, is determined by the statistical confidence interval calculated using the priority values of the candidate group. The confidence ratios to be used were determined by a learning mechanism on SDN according to the characteristics of the scenarios. SDN controller communicates with the client at the beginning of the video session and gives criteria weights to calculate priority ratios and the limit ratio regarding current network conditions to the client.

As a result of the study, it's seen that this kind of approaches can be used to provide uninterrupted and flexible service where short-term connection failures are present.

3.3 General Architecture

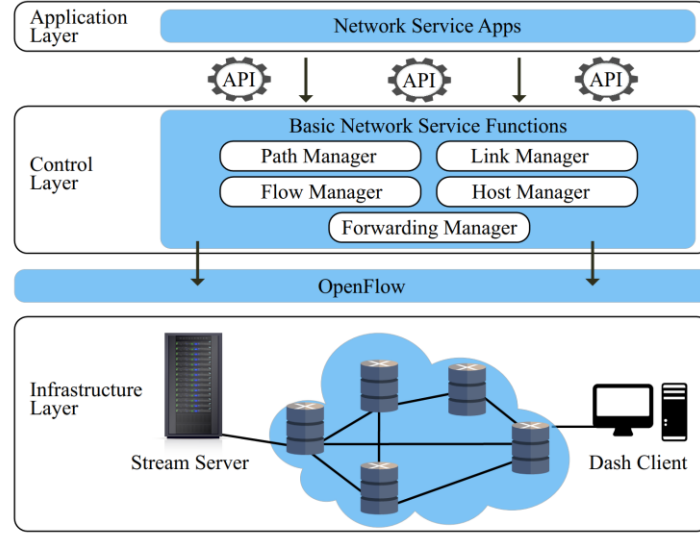


Figure 3.1 General Architecture of the Proposed System

In the proposed rate adaptation algorithm, the SDN controller has some basic network services that calculate the bandwidth of the paths and measure the traffic density. Figure 3.1 illustrates the general system with the controller modules. The host manager is responsible for monitoring the DASH clients entering the system. When a client enters the system, it sends the list of port numbers, which includes the set of ports that can be used by the client, to the SDN controller. Suppose np represents the maximum number of paths that can be used to transfer packets of a client. This number could be determined by network operator, or the video streaming company in case Internet Service Provider-Content Delivery Network (ISP-CDN) company cooperates. Based on its knowledge about network topology, the controller determines the disjoint paths in the network. It calculates the available bandwidth of these disjoint paths, maps the port numbers sent by the client with these path and sends this information to the newly joined client. In addition to that, the controller sends the rules related to selected paths, client's and server's (IP, port) tuples to the switches via the OpenFlow Protocol so that the switches forward the

packets of the clients over pre-determined paths. Note that, if the number of disjoint paths is greater than np , the controller selects np disjoint paths among them. In that case, Path Manager module of the controller selects the paths with the highest available bandwidth. Path changes may occur during video playback, the decision about the path change or about determining the optimal np are out of the scope of this study.

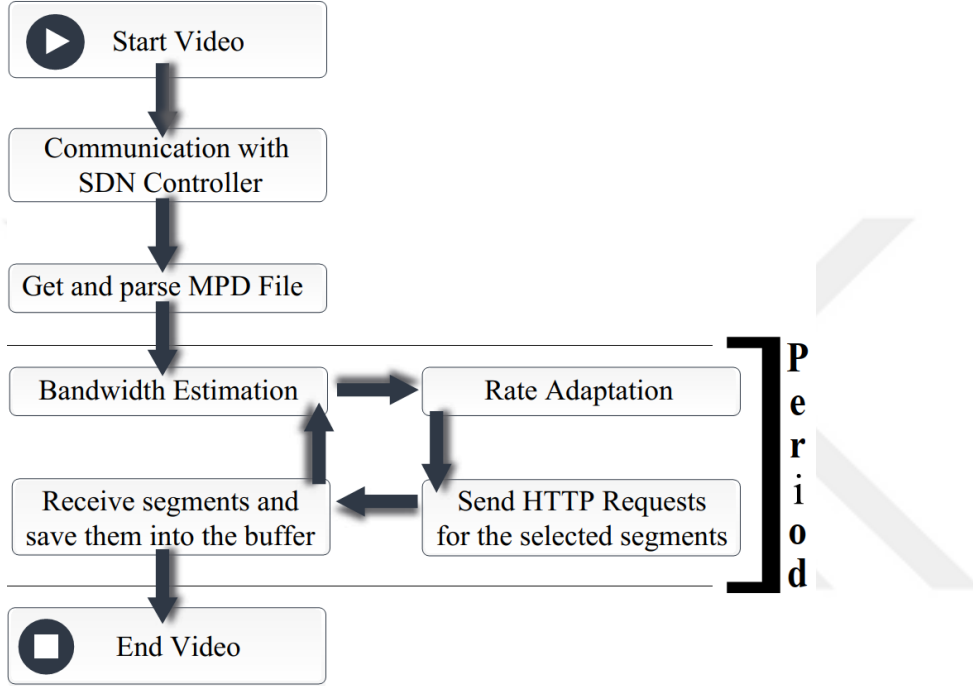


Figure 3.2 The modules running on the client side

The Figure 3.2 shows the modules running on the client side. In the proposed approach, the bandwidth information of each path is kept separately. At the beginning of the video session, the client retrieves the bandwidth values of np paths selected by SDN controller regarding to given clients port information. The client then requests the MPD file via the path with the highest bandwidth. After the MPD file is parsed, the bandwidth is re-estimated for the path from which the MPD file was retrieved. The rate adaptation algorithm determines the segments to be requested and the paths according to the estimated available bandwidth values and the current system state, which will be detailed in further sections. The requests are sent for the selected segments and then the segments downloaded are written to the buffer. After that, the bandwidth values of each path are re-estimated by including the last segment group (Rainer et al., 2012). In addition to the bandwidth values of

all paths, the overall bandwidth value is calculated by summing the bandwidth values of all paths to be used if appropriate conditions are met. The part in the flow which starts with the bandwidth calculation and ends with writing the received segments to the buffer is called a period. Initialization for a new period is not started until all desired segments are received for the previous period.

3.4 Improvement of the Dynamic Rate Adaptation Algorithm

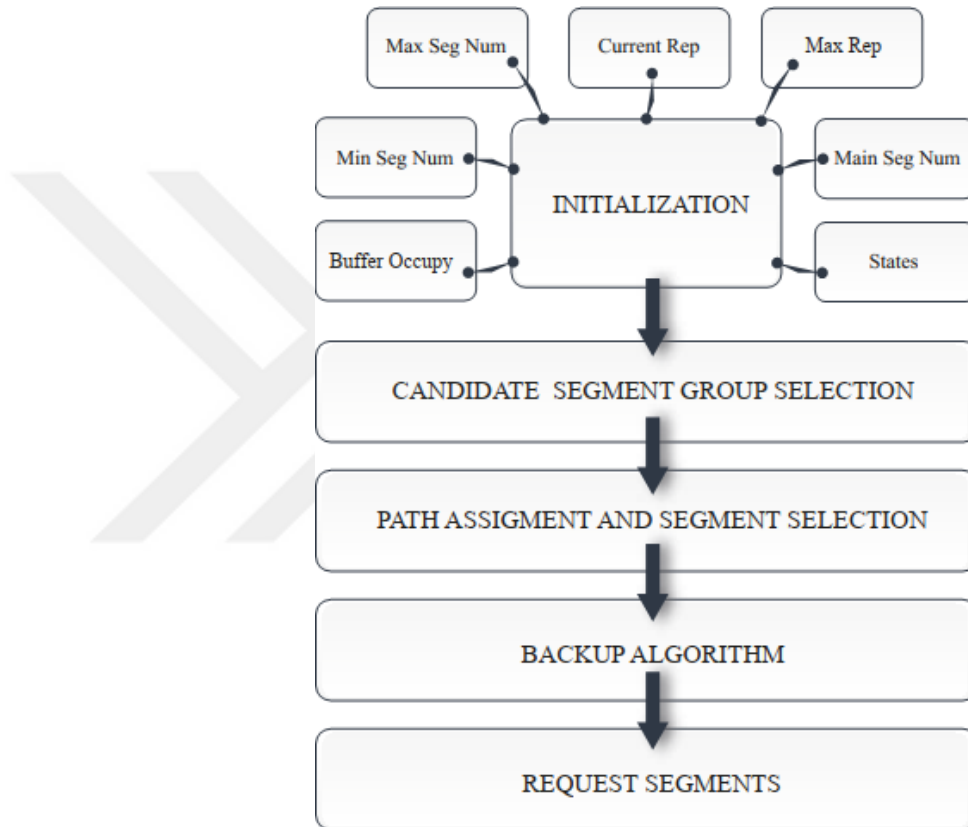


Figure 3.3 Rate Adaptation Flow Chart

Figure 3.3 shows the flow of the proposed rate adaptation algorithm. In order to better adapt to existing conditions, these conditions should be evaluated as fully as possible. For this reason, path information is obtained from the SDN controller and the segments to be requested are determined by also using this path information. In the proposed approach, these values, which will be used primarily in the path assignment and segment selection part, are calculated in the section called “Initialization”. This section also determines which state the system is in. The candidate segment group of the period is formed according to this specified state. The segments in the candidate segment group undergo some checks before path

assignment calculations are performed. Since there is no packet loss due to the use of TCP in case path changes and losses are ignored, if a segment in the candidate group has been previously requested, requesting this segment more than once will result in wasted bandwidth. Therefore, previously requested segments are not requested once more and they are removed from the candidate group. In the case of buffer filling during video start-up or outage, segments with the enhancement layer are also eliminated from the candidate group. This reduces the initial waiting and the outage durations. After these checks, if there is a suitable path for the segment that meets the other algorithm criteria, it will be requested. Where bandwidth is sufficient, improvements can be made from the minimum to the main segment number of the period. After the path assignment and selection is made within the candidate segment group, the backup algorithm is finally running. If the conditions are appropriate, new segments with the number from the main segment number to the maximum segment number may also be requested.

Table 3.1 shows the system variables with their symbols and descriptions which are used in the algorithms and FSM.

Symbol	Description
pl	Playout index
rep_{max}	Maximum representation estimated for current period
$rep_{max_{prev}}$	Maximum representation estimated for previous period
rep_{curr}	Representation estimated according to bandwidth of the path with the highest bandwidth
rep_{sum}	Representation estimated according to total bandwidth
sNo_{Min}	Minimum segment number of the period
sNo_{Main}	Main segment number of the period
sNo_{Max}	Maximum segment number of the period
bw_{pNo}	Bandwidth for path numbered pNo
occ_{pNo}	Path occupancy for the period
occ_{buffer}	Buffer fullness
$size_{sNo,lNo}$	Size of the segment with segment number sNo and layer number lNo
θ	Length of a segment
$sPath_{sNo,lNo}$	Selected path for the segment with segment number sNo and layer number lNo
$c_{sNo,lNo,pNo}$	Calculated download time of the segment with segment number sNo and layer number lNo from the path numbered pNo
$cTime_{max}$	Total calculated download time of a path with maximum downloading time

Table 3.1 Algorithm and FSM Variables

3.4.1 Initialization

In Initialization part, values such as main segment number, system phase and maximum available quality are determined at the beginning of each period.

Buffer Occupancy: Buffer occupancy is defined as 0 for the first period since no segment has been received yet. It is calculated by subtracting the playout index from the main segment of the previous period (Equation 1). Since the backup algorithm retrieves the segments irregularly from the main segment number to

maximum segment number, buffer occupancy calculations are limited to the part between the playout index and the main segment number of the period.

$$occ_{buffer} = sNo_{Main} - pl \quad (1)$$

Minimum Segment Number: When creating a new candidate segment group, it should be noted that the video is being watched, ie the playout index is constantly increasing. If the playout index approaches the main segment of the current period, this increases the risk of buffer outages. On the other hand, retrieving the layers of a segment number that is already monitored provides no benefit to QoE, while wasting resources. For these reasons, the first segment number of the candidate segment group (sNo_{Min}) is set to $pl + \alpha$ for improvement of already buffered segments and the client is given time to download.

Maximum Segment Number: For effective use of client resources, the buffer size is kept to a certain limit which is defined in configurations. If the number of segments in the buffer reaches $size_{buffer}$, no layer of a segment number can be requested, except those in the buffer. Therefore, after the downloaded segments are monitored, they are removed to free up the buffer for the new segments to be downloaded. The backup algorithm creates a candidate segment group between the main segment number and the maximum segment number for all layers, not to exceed the calculated maximum representation. The maximum downloadable segment number (sNo_{Max}) value used by the Backup algorithm is calculated as $pl + size_{buffer}$. The calculated sNo_{Max} does not exceed the quantity of the video segments.

Available Representation: The MPD file retrieved at the beginning of the video session contains the average bitrate information for all quality levels. Using average representation bitrates given in the MPD and the bandwidth estimations of all paths, rep_{curr} and rep_{sum} are calculated. rep_{curr} represents the quality level that the bandwidth of the path with highest bitrate allows. In addition to that, rep_{sum} is calculated using the sum of the bandwidth values of all paths determined by the SDN Controller.

States: Figure 3.4 shows the states transitions in the system. These states include information such as phase, safezone status and which bandwidth estimation to use. The first item on a state describes phase. If the second item is "Main" system calculates rep_{max} according to the path with the highest bandwidth only. If the item is "Overall", the total bandwidth of all available paths is used for the calculations. Zone status defined on the third item. "Safe" means the system is in safezone. In Figure 3.5 phases and zones based on buffer fullness are illustrated. As shown in the figure, zone and phase are defined independently of each other. In addition to buffer fullness, there are other conditions that affect the phase and zone status of the system.

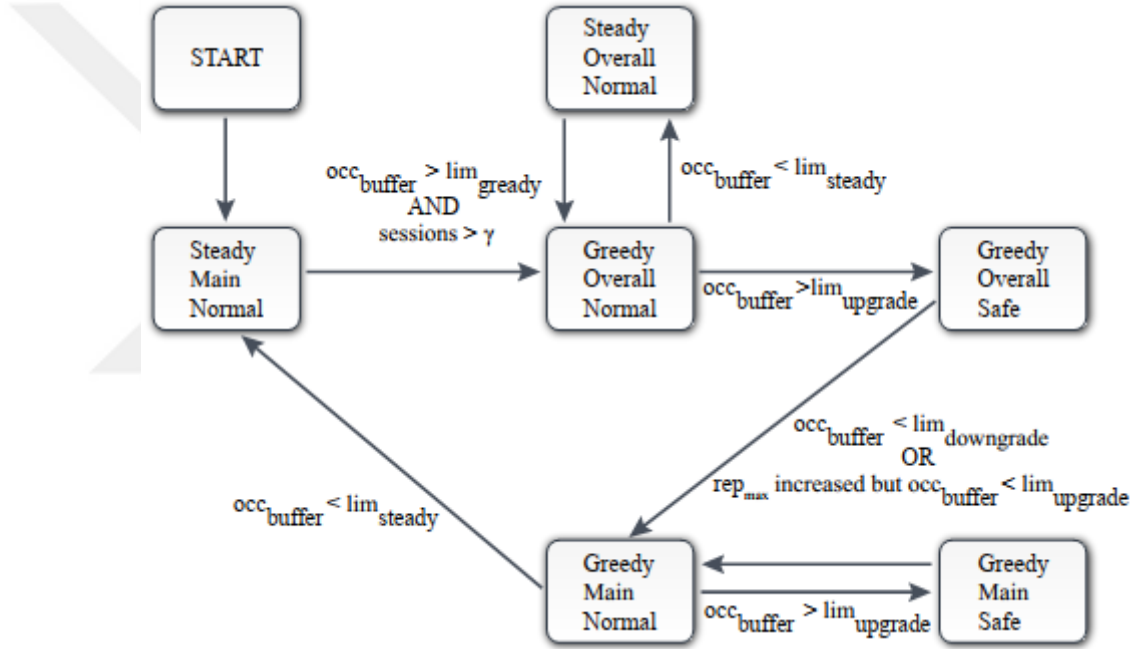


Figure 3.4 FSM of rate adaptation algorithm

In (Ozcan et al., 2017) only one top layer is requested after the topmost layer received of the segment numbers for the buffered segment optimization. Study shows that this vertical retraction of the already buffered segments provides a smooth quality change. In addition to this, in cases where buffer occupancy is saturated, the increase in the quantity of candidate segments makes the system more greedy and keeps buffer loading stable (Ozcan and Sayit, 2019). Based on these studies, there are two phases in the proposed rate adaptation algorithm; steady and greedy. The phase is set to steady at system start-up. At the beginning of the period, the lower (lim_{steady}) and the upper (lim_{greedy}) limits are set to determine which

phase the system is in for that period. If the buffer occupancy reaches lim_{greedy} , then the system is set to greedy phase after the γ of periods, where the buffer occupancy continuously exceeds this value. The system stays in the greedy phase until the buffer occupancy decreases to lim_{steady} . When the system goes into greedy phase, if rep_{sum} is greater than rep_{curr} , rep_{curr} is set to $rep_{curr} + 1$, and this value is used for calculations of the maximum quality that can be obtained rep_{max} . The reason that rep_{sum} is not used in this case is that the download rate of a segment to be received from one of the existing paths is lower than the download rate of the same segment to be taken from a path with the total bandwidth. However, when multiple paths are used, the number of the fluctuations in bandwidth increases due to the bandwidth fluctuations in each path, and therefore more frequently quality changes are observed. Additionally, reordering the segments downloaded from different paths takes time. If this happens, the download will not take place as planned because the system performs the representation calculation based on the total bandwidth.

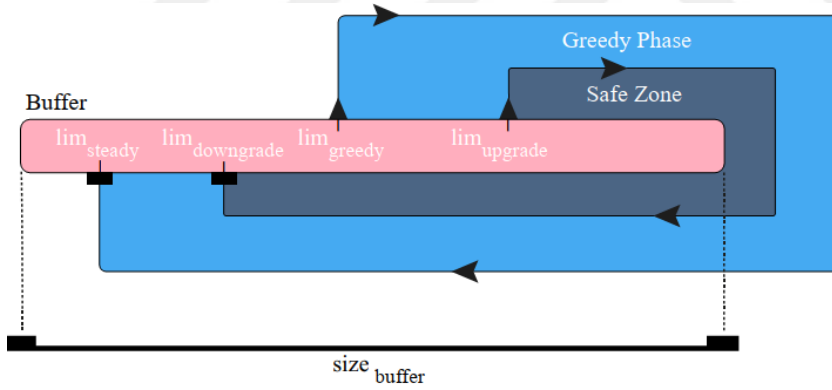


Figure 3.5 Phases and Zones Regarding Buffer Occupancy

Where available bandwidth is sufficient, the buffer may be fully saturated and all layers up to rep_{max} of all segment numbers in backward from sNo_{Min} to sNo_{Main} might be retrieved. This causes the bandwidth not to be used even if it is available. Therefore, in case of a safezone, segments of higher quality than rep_{curr} can be downloaded. To determine safezone lower ($lim_{downgrade}$) and upper ($lim_{upgrade}$) limits are assigned. The client enters the safezone, when buffer occupancy has reached $lim_{upgrade}$. If the system is still safe in the following periods, rep_{curr} , the representation that is decided to be used is increased to a higher value. The system

is removed from the safezone and the steady phase is determined if these two conditions happens at the same time: (i) rep_{curr} in a period is higher than or equal to maximum available representation of the previous period ($rep_{max_{prev}}$), ie there is an increase in available bandwidth as much as enough to increase available maximum representation when it is compared with the value of the previous period; (ii) the buffer occupancy is not greater than $lim_{upgrade}$. If the buffer fullness remains as expected after the system requests based on current values, the system is ready for further improvement and a new state is determined accordingly. Until the replenishment of the buffer exceeds $lim_{upgrade}$, the segments will continue to be requested with the quality established by using bandwidth values, phase status, and safezone. In this way, requesting high-quality segments which network can not transfer properly, which may cause unnecessary quality changes is prevented.

As a result of phase and safezone calculations, the value of rep_{curr} is determined to be used to calculate maximum available representation (rep_{max}) in a period.

Maximum Available Representation: The proposed approach aims to provide a gradual quality transition in bandwidth changes caused by the network. Therefore, if the difference between $rep_{max_{prev}}$ and the rep_{curr} of the current period is more than one, rep_{curr} is increased or decreased one by one (Algorithm 3.1 lines between 1-8). If the value of rep_{max} calculated for the current period is lower than $rep_{max_{prev}}$, and the system is in the safe zone, $rep_{max_{prev}}$ is used as rep_{max} for the current period (Algorithm 3.1 lines between 9-10). Thus, short-time bandwidth reductions do not increase the quantity of representation changes and affect QoE.

```

1   $rep_{max} \leftarrow rep_{max_{prev}}$ 
2  if  $ABS(rep_{curr} - rep_{max})$  then
3    if  $rep_{curr} > rep_{max}$  then
4       $rep_{curr} \leftarrow rep_{max} + 1$ 
5    else
6       $rep_{curr} \leftarrow rep_{max} - 1$ 
7    end
8  end
9  if  $NOT(rep_{curr} < rep_{max} \text{ AND } isSafeZone)$  then
10    $rep_{max} \leftarrow rep_{curr}$ 
11 end

```

Algorithm 3.1 Maximum Representation Settings

Main Segment Number: In cases where the video session is just started or maximum available representation indicates the base quality level, the period's main segment number is increased by β (Algorithm 3.2 lines between 1-2). Except in these cases, if the buffer is not fully loaded, the main segment number is increased by one per period (Algorithm 3.2 lines between 3-7). Otherwise, the main segment number is left the same as the previous period only in order to improve the quality of the segment numbers whose base layer is downloaded in the previous periods. The main segment number cannot be greater than the number of segments in the video (Algorithm 3.2 lines between 8-10).

```

1  if  $sNo_{Main} < \beta$  OR  $rep_{max} == 0$  then
2     $sNo_{Main} \leftarrow sNo_{Main} + \beta$ 
3  else
4    if  $occ_{buffer} < size_{buffer}$  then
5       $sNo_{Main} \leftarrow sNo_{Main} + 1$ 
6    end
7  end
8  while  $sNo_{Main} > videoLength$  do
9     $sNo_{Main} \leftarrow sNo_{Main} - 1$ 
10 end

```

Algorithm 3.2 Main Segment Number Settings

3.4.2 Candidate segment group selection

In the selection of the candidate segment group, all segment numbers from sNo_{Min} to sNo_{Main} are checked up to the rep_{max} permits (Algorithm 3.3 lines between 1-2). Paths are assigned to the segments that meet the algorithm criteria and requests are made for these segments (Algorithm 3.3 lines between 3-5).

```

1  foreach  $sNo$  from  $sNo_{Min}$  to  $sNo_{Main}$  do
2    foreach  $lNo$  from 0 to  $rep_{max}$  do
3       $sPath_{sNo,lNo} \leftarrow selectPath(sNo,lNo)$ 
4      if  $isFit(sNo,lNo)$  AND  $NOT(EMPTY(sPath_{sNo,lNo}))$  then
5         $addSegment(sNo, lNo, sPath_{sNo,lNo})$ 
6        if  $isSteadyPhase$  then
7           $BREAK$ 
8        end
9      end
10   end
11 end
12  $cTime_{max} \leftarrow MAX(ALL(occ_{pNo}))$ 

```

Algorithm 3.3 Pseudo-code for candidate group selection.

Steady Phase: In order to minimize initial waiting time and the outage duration, the lowest quality levels of the first β segment numbers after the main segment number of the previous period are taken in the case of video start-up and outages. The player will not start playing the video until the number of segments in the buffer reaches β .

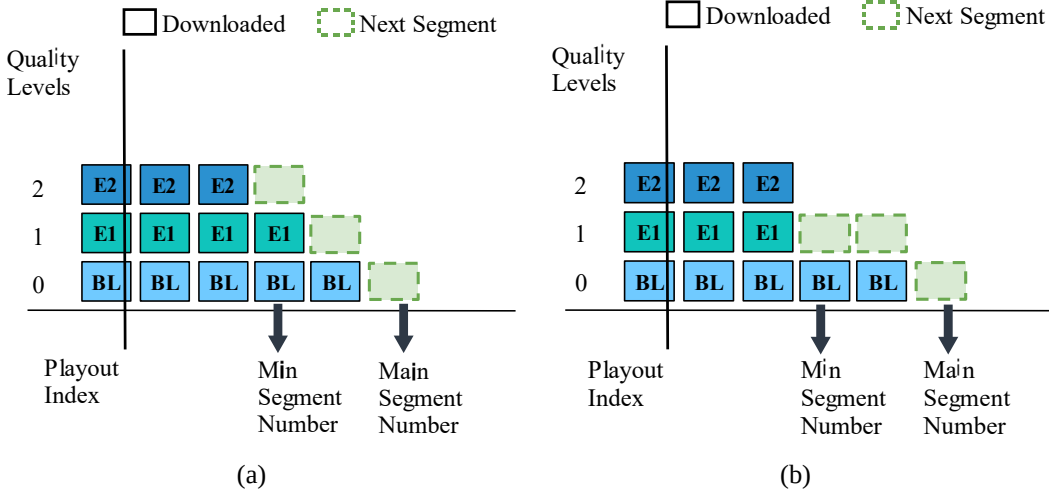


Figure 3.6 Buffer during candidate segment selection for Steady Phase

a) With same rep_{max} , b) With decreased rep_{max}

In this phase, the system can only request the smallest layer that has not yet been requested per segment number, if the conditions are appropriate. Because of this, if the system is in steady phase and a layer for a segment number is previously selected in the same period, the system exists from the loop of the layers for the segment number (Algorithm 3.3 lines between 6-8). Thus new segment numbers which have not been taken any layer yet have been given more chance to be downloaded. As a result, the risks of video pause have been reduced and at the same time, a gradual quality reduction has been achieved. Figure 3.6a shows the group of candidate segments to be selected when the rep_{max} value calculated in the period is the same as that of the previous period. If these segments are taken faster than the playout index progression, the buffer fill rate will be increased in the next period and the system will become more saturated for the next period. Figure 3.6b illustrates the case where rep_{max} falls when compared to the available maximum representation value of the period before the previous period. Since the group of segments to be received in the period is small, the period will take shorter time to be completed and the algorithm will be able to adapt itself to the new conditions more quickly if there are any positive/negative changes in the system. In such decreases in representation, the segment numbers received at low quality can be compensated in future periods as the periods complete their cycles in a short time. This also reduces the chance of using all open paths. The proposed approach aims to complete a period as soon as possible. Since the path with the highest bandwidth downloads a segment faster than the other available paths, downloading time of a

few of segments via this path might be less than downloading one of these segments via one of the other paths. In this case, all the segments are retrieved through the path with the highest bandwidth and the other paths are not in use until the period is completed. This waste problem is solved with backup algorithm that is explained in the next sections.

Greedy Phase: The system goes into this phase when the buffer is sufficiently full. Greedy phase states that rep_{max} and path assignments can meet the system needs to increase QoE. Therefore, in the case of appropriate conditions, all layers up to rep_{max} of the segments can be requested from the minimum segment number to the main segment number of the period (Figure 3.7). In this way, more segments are taken in a single period without having to make repeated initialization part. Since the quantity of segments in the candidate group is higher than that of the steady phase, the process of downloading all segments selected by the algorithm running in the greedy phase takes longer. This phase is more focused on improving the quality of segment numbers with one or more layers taken, rather than taking new segments to maintain or increase buffer fill. If there are a few of segment numbers to be improved in the backward, the period can be completed faster and the algorithm becomes better adapted to network fluctuations. Thus preventing the risk of buffer occupancy rate decrease and quality changes.

Figure 3.7a represents a scenario where rep_{max} increase to representation 2, after decreasing from representation 3 to 1. Thus, the quality of the segment numbers taken up to the first enhancement layers can be increased to 2. If there was nothing to improve for the backward segment numbers, only layers of the main segment number up to the rep_{max} could be requested.

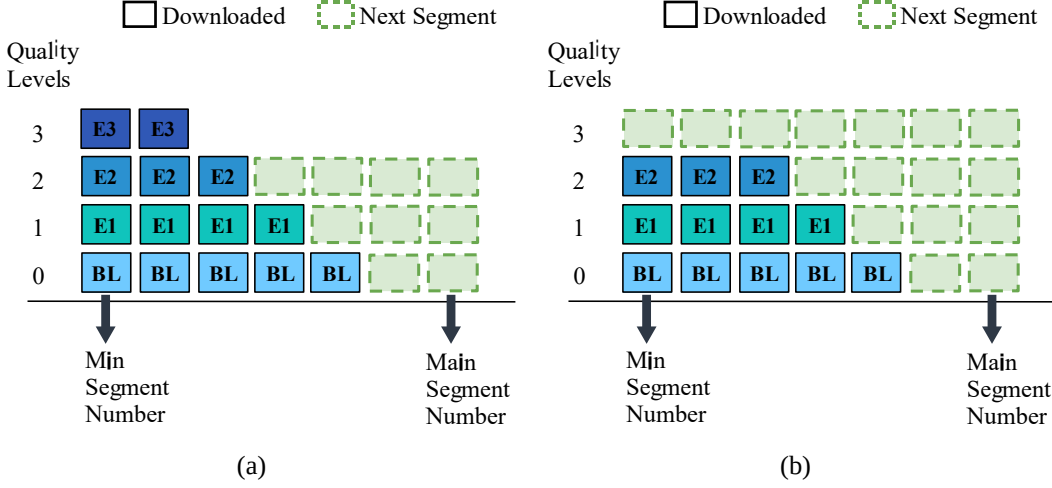


Figure 3.7 Buffer during candidate segment selection for Greedy Phase

a) With decreased rep_{max} , b) With increased rep_{max}

Figure 3.7b shows the selection of candidate segments if a period's rep_{max} value increases compared to previous periods. As it is seen, if rep_{max} value is increased during this phase, it causes an increase in the quantity of segments forming the candidate group and it can increase the completion time of that period.

3.4.3 Segment selection and path assignments

As previously mentioned, in cases such as video session start-up or outage, if a candidate segment has been requested before or the layer of the candidate segment is an enhancement layer, path assignment calculations are not performed for this segment and this segment is directly removed from the candidate segment group. Furthermore, for an enhancement layer, if rep_{max} is not greater than $rep_{max_{prev}}$, the system is in the steady phase, and the highest layer number requested of the previous segment number is less than the layer number; this segment is removed from the candidate segment group also in this case. (Algorithm 3.4 lines between 1-3). Thus, in case of an imbalance in segment sizes with sequential segment numbers, while the buffer is not saturated enough, representation changes are kept to a minimum. In addition, the system becomes more stable as bandwidth can be spent on new segment numbers that can saturate the buffer instead of segments removed from the candidate group.

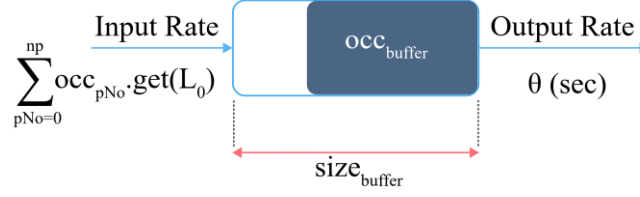


Figure 3.8 The relation between input/output rate and buffer occupancy

```

1  if  $lNo > 0$  AND  $isSteadyPhase$  AND  $NOT(isRequested(sNo-1, lNo)$ 
   AND  $rep_{max} \leq rep_{max_{prev}})$  then
2    return null
3  end
4   $minTime \leftarrow 0.0$ 
5   $sPath_{sNo, lNo} \leftarrow null$ 
6  foreach  $pNo$  from 0 to  $np$  do
7     $c_{sNo, lNo, pNo} \leftarrow occ_{pNo} + \frac{size_{sNo, lNo}}{bw_{pNo}}$ 
8    if ( $sPath_{sNo, lNo} == null$  OR  $minTime > c_{sNo, lNo, pNo}$ )
       AND  $sNo > pl + CEIL\left(\frac{c_{sNo, lNo, pNo}}{\theta}\right)$  then
9       $minTime \leftarrow c_{sNo, lNo, pNo}$ 
10      $sPath_{sNo, lNo} \leftarrow pNo$ 
11  end
12 end
13 if EMPTY ( $sPath_{sNo, lNo}$ ) AND  $lNo == 0$  then
14    $minTime \leftarrow c_{sNo, lNo, pNoWithHighestBandwidth}$ 
15    $sPath_{sNo, lNo} \leftarrow pNoWithHighestBandwidth$ 
16 end
17  $occ_{sPath_{sNo, lNo}} \leftarrow minTime$ 
18 return  $sPath_{sNo, lNo}$ 

```

Algorithm 3.4 Pseudo-code for path selection of a segment with segment number sNo and layer number lNo .

Since all requested segments from previous period have already been downloaded at the beginning of the new period, there is no data stream in the available and usable paths. Therefore, at the beginning of each period, the occupancy values of the all available paths (occ_{pNo}) are set to 0. The formula $\frac{size_{sNo, lNo}}{bw_{pNo}}$ shows how many seconds it will take to download segment with sNo segment number and lNo layer number through the path numbered pNo . In Figure

3.8, the relationship between input, output, and occupancy rates in the buffer are given. If we assume that the segment duration is θ , then the value of $\frac{size_{sNo,lNo}}{bw_{pNo}}/\theta$ of the segments with base layer indicates how many segments the playout index will go through during this process. The Equation (2) contains the steps of creating a generalized mathematical model that calculates the time it takes to download a segment through a path and estimates minimum segment number for the segment with that size to prevent retrieving already watched segment. If a segment can be downloaded in 4 seconds where the segment duration (θ) is 2 seconds, the difference must be at least 2 between pI and the segment number, ie the segment number to be requested must be greater than $pI + 2$. If it meets these conditions, the path that provides completing the download process in minimum time is selected for the relevant segment. In this matter, calculated download time of this segment is added to calculated download time of the previous segments of that period which assigned to the path (Algorithm 3.4 line 17). This value indicates occupancy of the path (occ_{pNo}). If no path is selected for a segment with base layer, the path with the highest bandwidth is assigned to this segment (Algorithm 3.4 lines between 13-15).

$$\forall size_{sNo,lNo}, bw_{pNo} \mid size_{sNo,lNo}, bw_{pNo} \text{ in } Q^+ \\ \forall \theta, pI \mid \theta, pI \text{ in } N^+$$

$$\begin{aligned} CEIL\left(\frac{size_{sNo,lNo}}{bw_{pNo}}/\theta\right) < 1 & \implies pI + 1 \\ CEIL\left(\frac{size_{sNo,lNo}}{bw_{pNo}}/\theta\right) < 2 & \implies pI + 2 \\ CEIL\left(\frac{size_{sNo,lNo}}{bw_{pNo}}/\theta\right) < 3 & \implies pI + 3 \\ & \vdots \\ CEIL\left(\frac{size_{sNo,lNo}}{bw_{pNo}}/\theta\right) < n & \implies pI + n \end{aligned} \quad (2)$$

3.4.4 Backup algorithm

The segments with segment number between sNo_{Min} and sNo_{Main} are selected and path assignments are made for them in the previous sections. The path with the maximum download time determines also the maximum download time for that

period ($cTime_{max}$). When retrieving segments from a path with the maximum download time, the other paths may be idle, which means that the bandwidths of them are not used. Besides, paths that have downloaded all segments assigned must wait for the path with the maximum download time. One of the main objectives of the proposed approach is to provide an uninterrupted data stream from all paths to be used between the server and the client. Therefore, the backup algorithm can assign new segments to be downloaded to paths with a total calculated download time which is less than $cTime_{max}$. This additional module aims to take all the layers that the period can take, not to exceed rep_{max} , from the sNo_{Main} to the sNo_{Max} (Algorithm 3.5 line 1-2), and to evaluate the time until all desired segments are downloaded from the path with the maximum download time. As shown in Figure 3.9, it assigns extra segments to all other paths except the path with the calculated maximum download time which also defines $cTime_{max}$. Since this backup algorithm can make decisions for segment numbers larger than the main segment number of the period, it reduces the burden of the basic decision algorithm in subsequent periods.

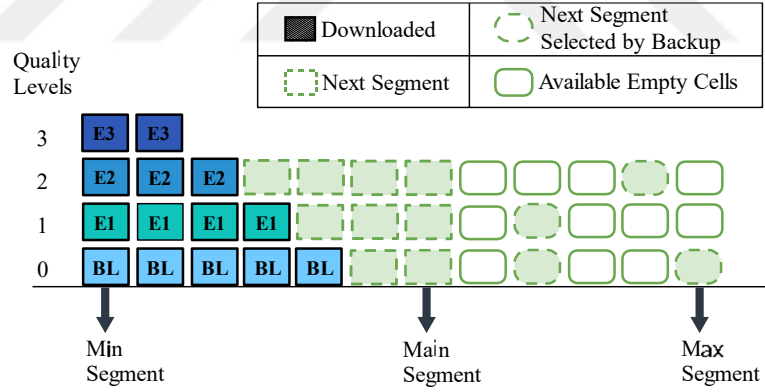


Figure 3.9 Buffer during segment selection for backup algorithm

```

1  foreach  $sNo$  from  $sNo_{Main} + 1$  to  $sNo_{Max}$  do
2    foreach  $lNo$  from 0 to  $rep_{max}$  do
3       $c_{sNo,lNo,pNo} \leftarrow occ_{pNo} + \frac{size_{sNo,lNo}}{bw_{pNo}}$ 
4      if  $isFit(sNo, lNo)$  AND  $c_{sNo,lNo,pNo} < cTime_{max}$  then
5         $occ_{sPath_{sNo,lNo}} \leftarrow c_{sNo,lNo,pNo}$ 
6         $addSegment(sNo, lNo, pNo)$ 
7      end
8    end
9  end

```

Algorithm 3.5 Pseudo-code for backup algorithm



4. PERFORMANCE RESULTS

As comparison approaches, we used conventional rate adaptation algorithms that request the next segment of an SVC encoded video. Both of the conventional approaches use the path with maximum available bandwidth as the main path. While Conventional Single-Path Client (convSinglePath) retrieves segments by using only this path, Conventional Multi-Path Client (convMultiPath), improves the quality by taking the segments with enhancement layers bigger than maximum available representation through the additional paths in addition to convSinglePath. convMultiPath downloads the upper layer of the maximum representation that bandwidth allows through the side path with the largest bandwidth. For the next layer, it uses the path among remaining side paths, which has the highest bandwidth. This selection goes until the client uses all paths assigned by the controller. All clients can select a group of segments at once, and then request them.

A total of 300 tests were performed for each video, 5 single client simulations and 5 multiple client simulations per scenario. The results are decomposed on the basis of scenario and video content, and the average values of these tests are reflected in the graphs.

4.1 Simulation Setup

The results are simulated using the Mininet software in SDN environment on Linux Operating System, including the OpenFlow switch used by SVC-DASH servers and clients. Floodlight, an OpenFlow controller, was implemented using the control platform.

4.1.1 Mininet

Simulations were performed by creating a virtual network over Mininet Platform. Mininet is an open source emulator that allows the design of simple or complex network architectures, the creation of software-defined networks, the realization and testing of streaming.

4.1.2 Floodlight Controller

Floodlight is an open source, Java-based network control platform that uses the OpenFlow protocol. In (Floodlight, 2019), Floodlight modules and architecture are given.

4.1.3 Network conditions

The performances of these clients are tested over 5 different scenarios which have different bandwidth values, oscillations and bitrate change counts. Table 4.1 shows average, maximum, minimum bandwidth values as well as oscillation levels of the main path to be used by the clients. The values are given according to simulations whose duration is the longest. The bandwidth of the additional paths remain the same during the simulations, however, since more than one clients transfer different layers over those paths, the clients are affected by the cross traffic produced by other clients, i.e. available bandwidth considerably changes during simulation.

Scenario	Minimum Bandwidth (Mbps)	Maximum Bandwidth (Mbps)	Average Bandwidth (Mbps)	Oscillations	Change Count
1	2	4	2,96	Low	6
2	1	6	4,16	High	7
3	3	9	6,05	High	5
4	5	7	6,07	Low	6
5	6	6	6	None	None

Table 4.1 Min,Max And Average Bandwidth Values (Mbps) Of The End-To-End Paths with Max Available Bandwidth In Each Scenario

4.1.4 Video contents

SVC-DASH encoded Big Buck Bunny, Elephants Dream and Tears of Steel videos were used in 720p quality. These videos have a total of 4 layers, one of which is the base layer, and the segment duration is 2 seconds for all of them. The

video lengths and cumulative average representation bitrates for each quality are given in Table 4.2, from the base layer to the top layer, respectively. These videos are selected for the application have different segment characteristics. Table 4.3 shows standard deviations of quality layers for each video contents. In general, the sizes of each segment varies, the bitrate values given in the MPD represents the average bitrate of whole segments. This allows us to obtain more realistic results from the application outputs since we can observe the behavior of the tested clients with segments having high variation in their sizes under similar network conditions.

Video Name	Video Length	Bitrates from low representation to high (Kbps)
Big Buck Bunny	299	1555, 2709, 4547, 6857
Elephants Dream	327	1661, 2821, 4886, 8196
Tears of Steel	368	1502, 2593, 4423, 7255

Table 4.2 Specifications of Video Contents

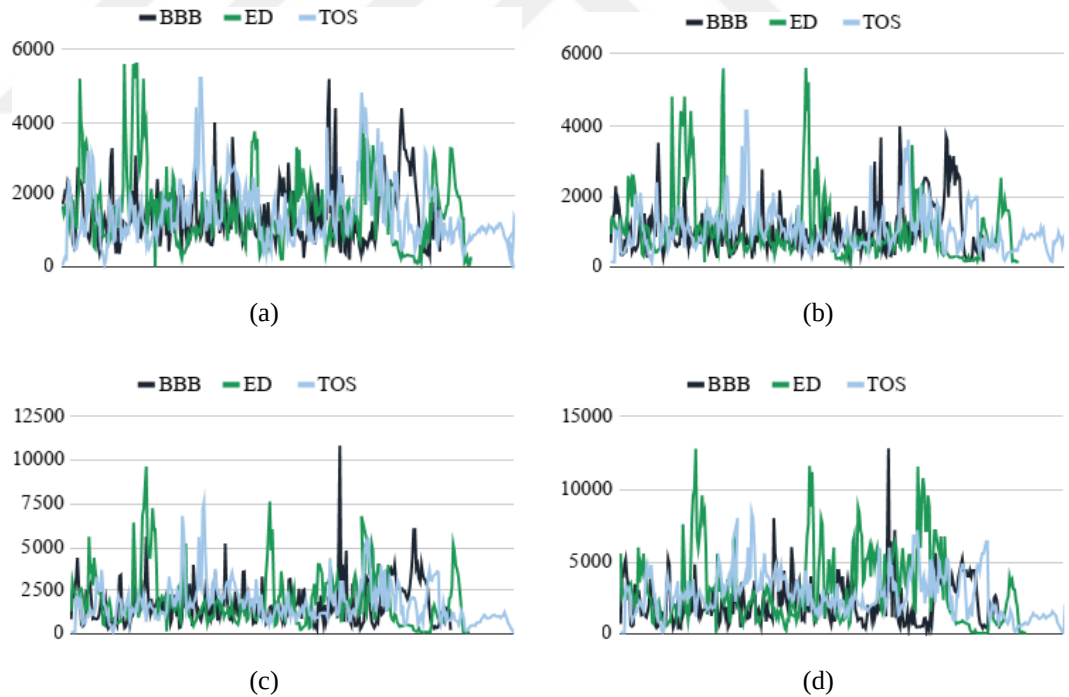


Chart 4.1 Segment Bitrates for each Quality Layer by Video Content a) Base Layer, b) Enhancement Layer 1, c) Enhancement Layer 2, d) Enhancement Layer 3

Video Name	Standard Deviations For Each Quality Layer (Kbps)			
	Base Layer	Enhancement Layer 1	Enhancement Layer 2	Enhancement Layer 3
Big Buck Bunny	836	699	1194	1499
Elephants Dream	1039	962	1569	2535
Tears of Steel	853	618	1091	1509

Table 4.3 Standard Deviations of Video Contents

4.1.5 Configurations

Configurations made during application are specified in Table 4.4.

Symbol	Value	Description
np	3	Path counts
α	2	Segments after playout
$lim_{downgrade}$	8	Lowest buffer occupation to quit safezone
$lim_{upgrade}$	15	Buffer occupation limit to enter safezone
lim_{steady}	8	Lowest buffer occupation for switching to steady phase
lim_{greedy}	6	Highest buffer occupation for switching to greedy phase
β	4	Initial segment count
γ	2	Number of periods after greedy limit is exceeded to set greedy phase
$size_{buffer}$	20	Buffer size

Table 4.4 Configuration Items

4.2 QoE Calculation

The Equation (3) contains modified version of the formulas used in QoE calculations (Yin et al., 2015). "rB" refers to the average bitrate value (KBps) monitored in a video session. The higher the bitrate received in a session, the more positive the user experience is. Initial waiting times (sec), representation changes

and video outages negatively affect the user experience. Initial waiting time is defined by "iW". In cases where this value is less than 2.5 seconds, the value "iW" is set to 0. "rC" indicates the quality changes count in the session. "uC" represents the number of outage while "uD" (sec) is the total duration of video outages in a session.

$$\begin{aligned}
 &\forall a_0, a_1, a_2, a_3, a_4 \mid a_0, a_1, a_2, a_3, a_4 \text{ in } Q^+ \\
 &\forall rB, iW, rC, uC, uD \mid rB, iW, rC, uC, uD \text{ in } \{0, Q^+\} \\
 &QoE^+ = rB \times a_0 \\
 &QoE^- = iW \times a_1 + rC \times a_2 + uC \times a_3 + uD \times a_4 \\
 &QoE = QoE^+ - QoE^- \tag{3}
 \end{aligned}$$

The weights used in the calculations are given in Table 4.5. These weights were calculated using the AHP, an MCDM method. AHP is one of the methods applied initially among MCDM methods. Because of its initial stage, simplicity and robustness, is used by all MCDM methods (Ozcan and Sayit, 2019). The main objective of AHP is to determine the coefficients of the variables in a linear equation. Therefore, as a first step, a decision matrix is formed by comparing the priority of all binary variables. In these binary comparisons, the values between 1-9 can be used from the lowest to the highest, respectively. The values used in these comparisons were determined based on the information about the priorities of the QoE parameters presented in the BACKGROUND AND RELATED WORKS/QoE section of this study.

	a_0	a_1	a_2	a_3	a_4
Value	0,236	0,049	0,092	0,436	0,187

Table 4.5 Weights of the QoE Metrics

For QoE calculation, each QoE parameter is multiplied by the weights which were determined as the output of AHP algorithm. The QoE value of a session is obtained by subtracting of weighted parameters that negatively affect the user experience from the weighted parameters which has positive effects on user experience.

4.3 Performance Results

The test results are divided into two parts as “single client simulations” and “multiple clients simulations” and the graphics are given separately for each video content.

4.3.1 Single client simulations

Chart 4.2 represents average initial waiting times for each clients per video content. Since mpBack² focuses on distributing the segments to paths for decreasing the period time, proposed approach is better than the other conventional approaches in terms of initial waiting time. As a result, the proposed approach achieved a maximum 59% and an average of approximately 35% decrease for initial waiting time in all tests compared to the two conventional approaches.

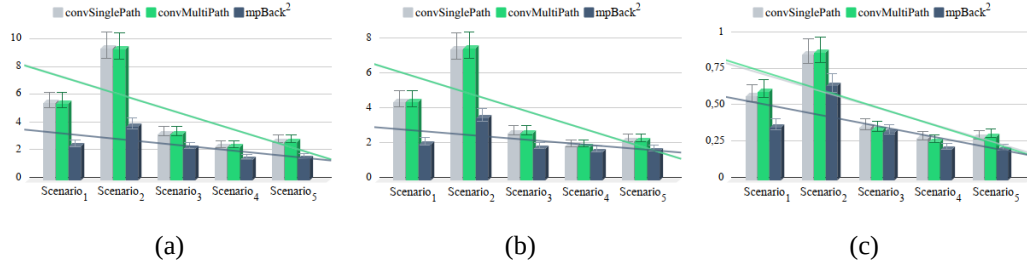


Chart 4.2 Average Initial Waiting Times of Single Client Simulations a) Big Buck Bunny, b) Elephants Dream, c) Tears of Steel

The average representation bitrates are shown in Chart 4.3. For all scenarios and video contents, average representation bitrates are higher than those in the other approaches. This shows that the proposed algorithm provides better visual quality under these circumstances. As a result, the proposed approach provided an increase in the average quality by a maximum of 175%, average of 67%, when compared to convSinglePath. When compared to the convMultiPath, this improvement was measured as a maximum of 69% and an average of 36%.

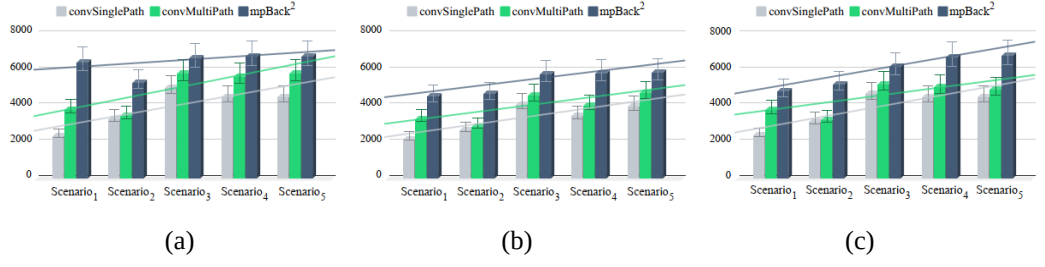


Chart 4.3 Average Representation Bitrates of Single Client Simulations a) Big Buck Bunny, b) Elephants Dream, c) Tears of Steel

In Chart 4.4, the averaged representation changes are comparatively given for all types of clients. The average representation changes count obtained in the proposed approach is less than that of the convMultiPath for most of the scenarios of all video contents. On the second scenarios of Chart 4.4b and Chart 4.4c, more representation exchanges took place than conventional clients because of a gradual transition is preferred by proposed approach rather than a sharp quality changes. For the other tests, convMultiPath has seen a lot of quality change in short-term bandwidth changes, while in the proposed approach these changes decrease by an average of 82%. According to the convSinglePath, the representation values are almost identical for all other simulations.

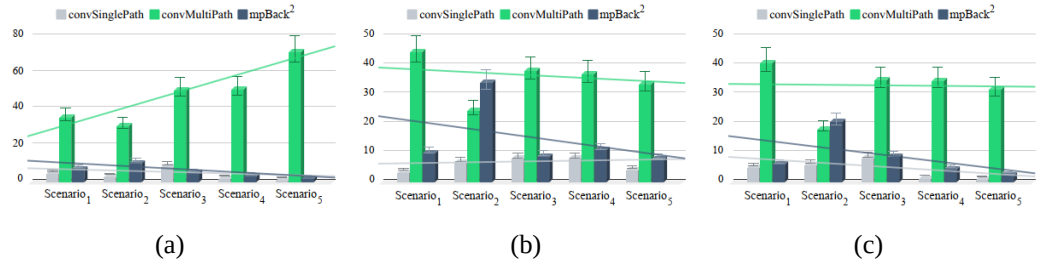


Chart 4.4 Average Representation Changes of Single Client Simulations a) Big Buck Bunny, b) Elephants Dream, c) Tears of Steel

The proportions of the qualities monitored in the video sessions are given as the stack graphs in Chart 4.5. It is seen that the clients running the proposed approach are able to watch the highest quality segments in all tests. In scenario 1 and 2 of all video contents, both conventional approaches can not watch any segment of the highest quality, while the highest quality video watching rate is a maximum 88% and average 45% for the proposed approach. When all scenarios are taking into account, it is seen that generally, the convSinglePath does not receive

any segments of the highest quality. However, when the convMultiPath can watch close to 25% of the highest quality segments, mpBack² watches almost all video segments at the highest quality.

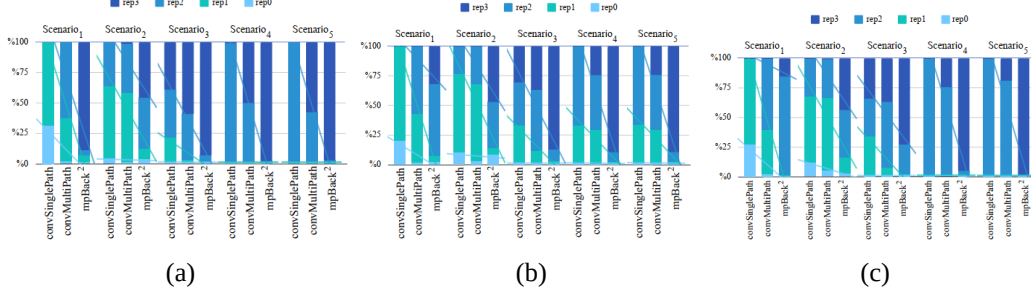


Chart 4.5 Average Watched Segments of Single Client Simulations a) Big Buck Bunny, b) Elephants Dream, c) Tears of Steel

In the proposed rate adaptation algorithm, the clients minimize the buffer outage by sending requests for segments lower layered where the measured throughput decreases by controlling system state. In Tables 4.6, 4.7, 4.8; outage statistics are given for all scenarios of the all video contents. As shown in the tables, buffer outages are observed only second scenario of Tears of Steel and first two scenarios of Elephants Dream with the proposed approach. On the other hand, buffer outages are observed most of the scenarios of all video contents where bandwidth is dynamically changing or segment sizes are not fit with average representation described in MPD files with the conventional approaches. Considering that viewers prefer initial waiting time or representation changes instead of video outages, the higher representation changes of mpBack² in the second scenarios of Elephants Dream and Tears of Steel are compensated for by the decrease in buffer outage times. When it is compared to both conventional approaches, the outage count is reduced by up to 18% for proposed approach. From the point of outage durations, mpBack² decreased the time duration at maximum of 99% in Table 4.7.

	convSinglePath		convMultiPath		<i>mpBack</i>²	
Scenario	#	Time	#	Time	#	Time
1	1	11,4	1	9,9	0	0
2	0,8	4	0,8	4,65	0	0
3	1	9,2	1	10,56	0	0
4	0	0	0,2	2,5	0	0
5	0	0	0	0	0	0

Table 4.6 Buffer Outage Counts and Durations of Single Client Simulations for Big Buck Bunny

	convSinglePath		convMultiPath		<i>mpBack</i>²	
Scenario	#	Time	#	Time	#	Time
1	2,2	40,95	2,2	40,96	1,8	0,92
2	2,2	34,27	2,2	36,01	0,2	0,07
3	0	0	0	0	0	0
4	1	11,23	1	11,85	0	0
5	2,8	29,08	3	30,88	0	0

Table 4.7 Buffer Outage Counts and Durations of Single Client Simulations for Elephants Dream

	convSinglePath		convMultiPath		<i>mpBack</i>²	
Scenario	#	Time	#	Time	#	Time
1	0	0	0	0	0	0
2	0,2	1,56	0	0	0,2	0,11
3	0	0	0,8	1,32	0	0
4	0	0	0,2	5,15	0	0
5	0	0	0	0	0	0

Table 4.8 Buffer Outage Counts and Durations of Single Client Simulations for Tears of Steel

Regarding to the overall QoE values in Chart 4.6, the proposed approach provides a clear improvement in all single client simulations compared to the other clients. Considering that these values evaluate all metrics fairly, it can be said that the proposed approach has achieved its purpose. *mpBack*² succeeded about QoE up to 188% compared to convSinglePath, and 80% compared to convMultiPath.

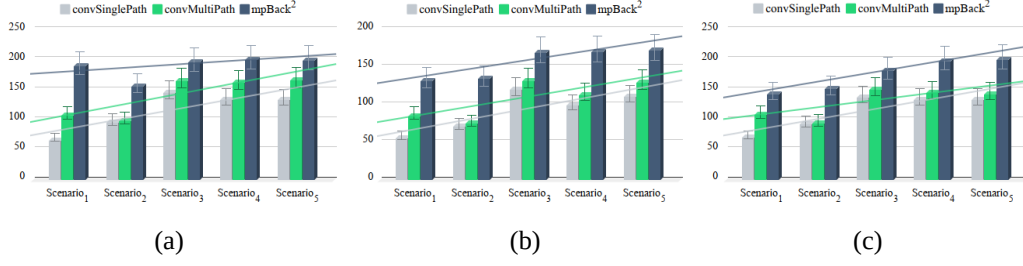


Chart 4.6 Average QoE of Single Client Simulations a) Big Buck Bunny, b) Elephants Dream, c) Tears of Steel

4.3.2 Multiple client simulations

In the simulations performed as multiple clients, 4 clients were used: convSinglePath, convMultiPath, and two *mpBack*² clients. Multiple client simulations were performed for comparison with conventional approaches and observing the situation where more than one proposed approaches share the same network.

Chart 4.7 shows that the proposed approach in multiple client simulations provided a general reduction in the initial waiting time.

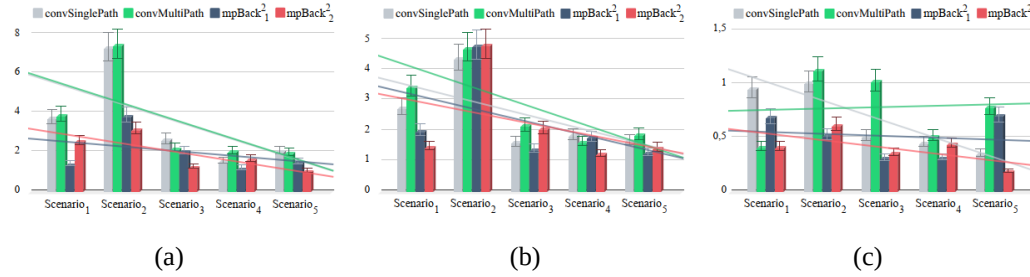


Chart 4.7 Average Initial Waiting Times of Multiple Client Simulations a) Big Buck Bunny, b) Elephants Dream, c) Tears of Steel

In Chart 4.8, the average representation bitrate values obtained in the video session are given. For multiple client tests, the bandwidths of all paths were increased in direct proportion to the number of clients. As a result, more usable bandwidth was provided in the common denominator, and each client had the privilege to use unused bandwidths. This situation provides the highest quality for all videos in scenarios 3,4 and 5. For this reason, the representation values of all clients for these scenarios are very close to each other and all approaches are able

to watch these videos mostly from the highest quality. Considering scenarios 1 and 2, both $mpBack^2$ clients experience similar performance in terms of the average representation bitrate being played, resulting in 42% improvement over convSinglePath and 14% over convMultiPath.

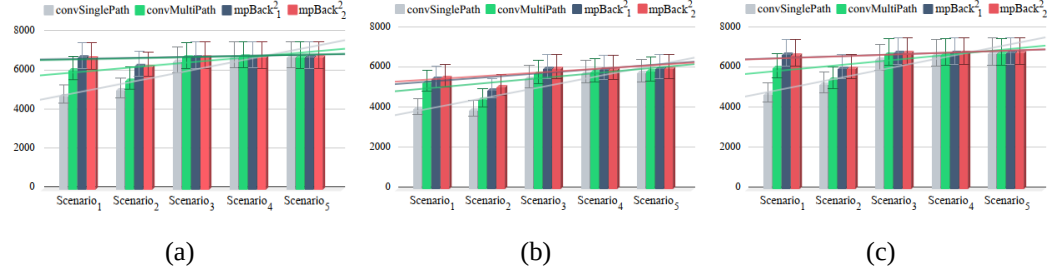


Chart 4.8 Average Representation Bitrates of Multiple Client Simulations a) Big Buck Bunny, b) Elephants Dream, c) Tears of Steel

Chart 4.9 shows representation changes for the video sessions. Except 4th and 5th scenarios of Big Buck Bunny, running clients on proposed approaches have completed these video sessions with a relatively low number of representation changes than other clients. The reason for the minimal variation in the number of representation changes in the Big Buck Bunny scenarios 4 and 5 is that the proposed approach provides gradually representation transitions rather than large representation jumps. In other scenarios, a maximum improvement of 95% was measured compared to two conventional clients. Average improvement rates are 79% according to convSinglePath and 75% according to convMultiPath.

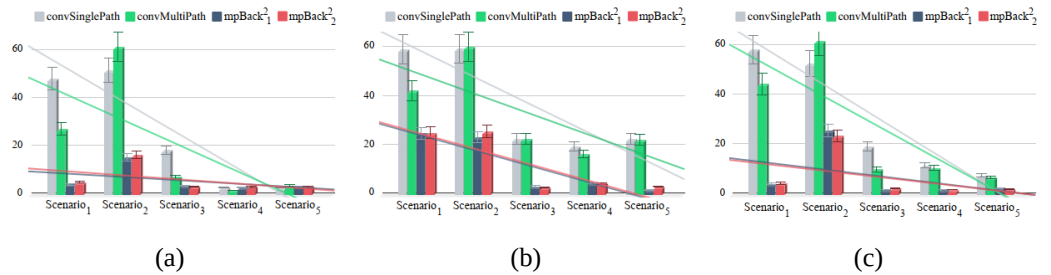


Chart 4.9 Average Representation Changes of Multiple Client Simulations a) Big Buck Bunny, b) Elephants Dream, c) Tears of Steel

Chart 4.10 represents the watched segments as stack graphs. As same as the representation bitrate, there is an improvement in the monitored segments due to the use of multi-path and total bandwidth. Clients who could not watch the top

segments in single client simulations were able to watch the video in high quality in multiple tests. When considering the overall average in the highest quality segment tracking, maximum 265% (Chart 4.10b Scenario 1) and the average 70% improvement has been resulted when compared to convSinglePath. According to the convMultiPath, a maximum 79% (Chart 4.10b Scenario 2) and average 20% compared improvement is achieved.

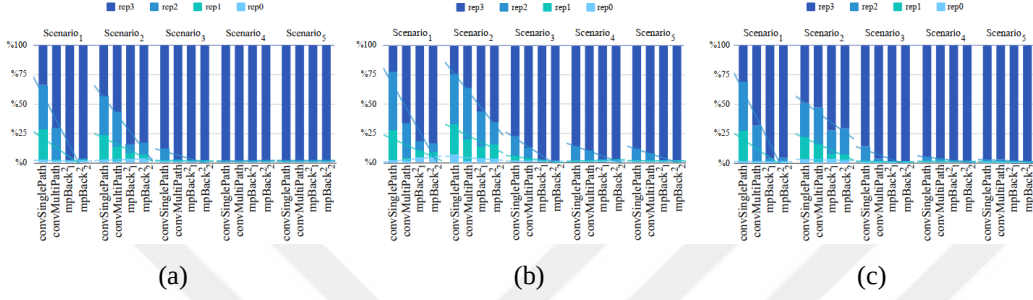


Chart 4.10 Average Watched Segments of Multiple Client Simulations a) Big Buck Bunny, b) Elephants Dream, c) Tears of Steel

Multiple client simulations showed a significant difference in the number and duration of buffer underruns in terms of mpBack² compared to single client simulations (Tables 4.9, 4.10, 4.11). Both conventional clients experience buffer underruns in all tests except 4th scenario of Big Buck Bunny and elongated video outages compared to the tests performed with single client. Even observing that the number of representation changes and initial waiting time in the proposed approach are longer than those in conventional approaches in some scenarios, overall QoE is still higher in mpBack² since buffer underruns are very limited with it in these scenarios, where conventional clients suffers from higher count and duration of underruns. In the proposed approach, the number of underruns yielded up to 83% better than the convSinglePath and up to 86% better than the convMultiPath. In terms of buffer underrun duration, these values are measured at a maximum of 95% lower for both conventional approaches.

	convSinglePath		convMultiPath		<i>mpBack</i>²		<i>mpBack</i>²₂	
Scenario	#	Time	#	Time	#	Time	#	Time
1	3,4	42,72	4	49,08	0	0	0	0
2	4	58,62	4,2	61,76	1,4	4,44	1,4	7,5
3	1,4	12,88	1,2	10,26	0,4	0,3	0,2	0,2
4	0	0	0	0	0	0	0	0
5	0,2	0,8	0	0	0	0	0	0

Table 4.9 Buffer Outage Counts and Durations of Single Client Simulations for Big Buck Bunny

	convSinglePath		convMultiPath		<i>mpBack</i>²		<i>mpBack</i>²₂	
Scenario	#	Time	#	Time	#	Time	#	Time
1	8,4	131,44	10,2	148	1,4	1,6	1,4	1
2	5,6	66,76	7,2	88,12	2,4	2	2,4	3,19
3	3	33,56	3,2	38,12	0	0	0	0
4	3	43,60	2,4	39,5	0	0	0	0
5	3,8	47,49	4,4	54	0	0	0	0

Table 4.10 Buffer Outage Counts and Durations of Single Client Simulations for Elephants Dream

	convSinglePath		convMultiPath		<i>mpBack</i>²		<i>mpBack</i>²₂	
Scenario	#	Time	#	Time	#	Time	#	Time
1	7,6	8,7	9,2	112,1	0	0	0	0
2	8,2	11,1	9,4	125,8	3,4	3,48	3,2	2,97
3	0,4	2,61	1	6,86	0	0	0	0
4	1,6	15,38	1,8	18,4	0	0	0	0
5	1	7,19	1,2	9,99	0	0	0	0

Table 4.11 Buffer Outage Counts and Durations of Single Client Simulations for Tears of Steel

Concerning QoE values, both clients running the proposed approach give similar values (Chart 4.11). Even in scenarios 3,4 and 5 of Elephants Dream and Tears Of Steel videos where total bandwidth is best suited for high-quality segment viewing, the QoE value of both conventional approaches is lower than that of the proposed approach. In scenarios 1 and 2, where bandwidth is not sufficient for all representation levels, the difference measured in QoE can be seen more clearly. As

result for multiple client simulations, the proposed approach increased QoE at maximum by 90% for convSinglePath and 38% for convMultiPath.

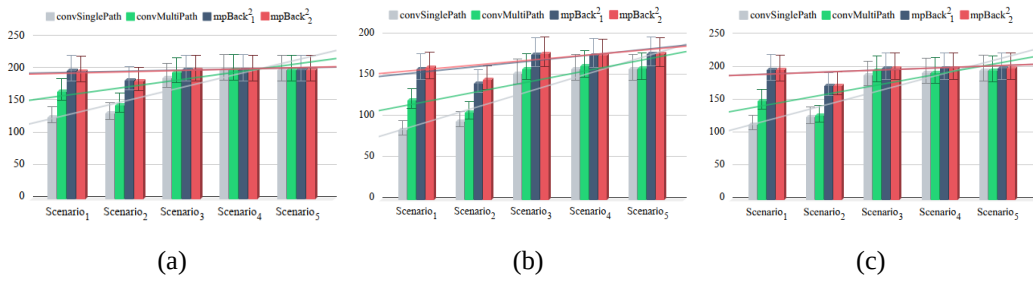


Chart 4.11 Average QoE of Multiple Client Simulations a) Big Buck Bunny, b) Elephants Dream, c) Tears of Steel

5. CONCLUSIONS

Using a scalable coded video is a promising alternative as to be used in today's most popular video streaming systems, DASH, since it is possible to adapt quality by extracting and adding video layers while utilizing web caches better than AVC coded video due to the need of only one file to obtain quality alternatives. Basically, SVC provides having different segment qualities by using layers, where base layer constitutes the lowest quality and each additional layers called enhancement layers provide quality improvements.

The structure of SVC brings advantage of increasing video quality by adding layers and this property of SVC can be used in increasing the quality of the on-fly video as well as already buffered video. When network conditions are getting better, increasing the quality of already buffered video by using SVC enhancement layers without wasting buffered low-quality segments is one of the contributions proposed in this thesis.

In cases where bandwidth is fluctuating or low, multiple paths can be used to overcome the problem of inadequate and unbalanced bandwidth from one path. SDN is defined as a new way of computer network architecture, where data and control plane are separated from each other, which provides to use alternative routing algorithms according to the application needs. By using SDN, it is possible to assign one or more paths between the clients and servers in the network with intelligent network controllers. These path assignments can be made according to the paths with the highest bandwidth, least traffic or other routing algorithms according to the needs of the system. SDN can authorize a rate adaptation algorithm for multi-path management and routing. In addition, it can redirect the information needed about the network to the client.

When SVC and SDN architecture are utilized in DASH systems by using more than one paths, they provide some advantages such as storage, bandwidth and increasing QoE. However, this system raises the questions of which layer of the which segment should be taken, in which order and from which path and makes segment selection more complicated. In this thesis, an SDN aware multi-path rate

adaptation algorithm that aims to improve the QoE as desired by video providers to prevent user losses is proposed.

Because of the contribution that SVC and SDN can make to a video streaming system, it is advantageous to use both together. To use these advantages, the heuristic approach and MCDM methodologies were used in previous versions of this study. While the first study focuses on the improvement of the backwardly SVC coded segments and downloads them vertically according to the heuristic method, the focal point of the second study is selecting the segments to be downloaded as much as possible for each decision period by using an MCDM method.

The proposed approach decides a group of segments to retrieve in a decision period and aims to minimize the time spend. To do so, segments to be downloaded should be allocated to keep all available paths busy during the video session. In order to adapt to dynamic network conditions, states are set for each period before segment selections. These states are defined based on bandwidth, buffer occupancy, and the current system states by also using the phase and zone.

When it is desired to determine the quality of a video surveillance session, metrics such as received bitrate, quality changes, video pauses should be evaluated. These metrics are referred to as QoE metrics in the literature. Studies show that some of these metrics are preferred by the audience over others. Therefore, these metrics should affect the overall quality calculated at different rates. AHP, an MCDM method, was used to determine the weights of metrics on overall QoE.

Mainly, the proposed system communicates with SDN controller to get information about network, estimates bandwidth values, sets its states, selects and retrieves segments by using available paths based on a mathematical modal. It targets to prevent wasting network resources such as bandwidth and increase received bitrates while decreasing outages, representation changes and initial waiting times.

Two traditional approaches using single and multi paths were used in the simulations and all of the videos used were coded with SVC. Simulation results

show that using only SVC or Multi-Path without a well-planned strategy is not enough to increase QoE. However, the proposed algorithm maximizes the positive QoE metrics while minimizing the negative ones when compared to both conventional clients. In single client simulations, the proposed approach improve the QoE up to to 188% compared to Conventional Single-Path Client, and 80% compared to Conventional Multi-Path Client. As result for multiple client simulations, proposed approach this improvement at maximum by 90% for Conventional Single-Path Client and 38% for Conventional Multi-Path Client.

In future work, investigating dynamic buffer size regarding to video properties and network conditions, machine learning based estimation for available maximum representation will be planned to be studied.

REFERENCES

- Adhikari, V. K. et al.**, 2012, Unreeling Netflix: Understanding and Improving Multi-CDN Movie Delivery., *IEEE INFOCOM*, pp. 1620–28.
- Akhshabi, S., Anantakrishnan, L., Begen, A.C., and Dovrolis, C.**, 2012, What Happens when HTTP Adaptive Streaming Players Compete for Bandwidth?, *In Proceedings of the 22Nd International Workshop on Network and Operating System Support for Digital Audio and Video*, (New York, NY, USA: ACM), pp. 9–14.
- Akhshabi, S., Begen, A.C., and Dovrolis C.**, 2011, An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP., *Second Annual ACM Conference on Multimedia Systems*, MMSys '11, New York, NY, USA: ACM, pp. 157–168.
- Anon. n.d.**, 2019, “Network Infrastructure.”, *Google AI*. (Retrieved 14 March 2019)
- Xue, J., Zhang, D., Yu, H., Chen, C.W.**, 2014, Assessing quality of experience for adaptive HTTP video streaming, *In 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pp. 1–6.
- Shi, W., Wang, Z., Zhou, C., Sun, L.**, 2015, Characterizing viewing engagement patterns in DASH, *In 2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pp. 1–6.
- Andelin, T., Chetty, V., Harbaugh, D., Warnick, S., and Zappala, D.**, 2012, Quality selection for Dynamic Adaptive Streaming over HTTP with Scalable Video Coding, *In Proceedings of the 3rd Multimedia Systems Conference on - MMSys '12*, (Chapel Hill, North Carolina: ACM Press), p. 149.
- Belkadi, O., and Laaziz, Y.**, 2017, A Systematic and Generic Method for Choosing A SDN Controller., *International Journal of Computer Networks and Communications Security*, 5, 11, pp. 239–247.

REFERENCES (more)

- Bouten, N., Famaey, J., Latré, S., Huysegems, R., Vleeschauwer, B.D., Leekwijck, W.V., and Turck, F.D.,** 2012, QoE optimization through in-network quality adaptation for HTTP Adaptive Streaming, *In 2012 8th International Conference on Network and Service Management (Cnsm) and 2012 Workshop on Systems Virtualization Management (Svm)*, pp. 336–342.
- Cetinkaya, C., Ozveren, Y.i and Sayit, M.,** 2015, An SDN-Assisted System Design for Improving Performance of SVC-DASH., *Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 819–26.
- Cicco, L.D., and Mascolo, S.,** 2014, An Adaptive Video Streaming Control System: Modeling, Validation, and Performance Evaluation, *IEEE/ACM Transactions on Networking* 22, pp. 526–539.
- Cisco Visual Networking Index,** Forecast and Trends, 2017–2022 *White Paper*-Cisco. (Retrieved 12 July 2019).
- Cofano, G., Cicco, L.D., Zinner, T., Nguyen-Ngoc, A., Tran-Gia, P., and Mascolo, S.,** 2017, Design and Performance Evaluation of Network-assisted Control Strategies for HTTP Adaptive Streaming, *ACM Trans. Multimedia Comput. Commun. Appl.* 13, pp. 42:1–42:24.
- Egilmez, H.E., Civanlar, S., and Tekalp, A.M.,**2013, An Optimization Framework for QoS-Enabled Adaptive Video Streaming Over OpenFlow Networks, *IEEE Transactions on Multimedia* 15, pp. 710–715.
- Elgabli, A., Liu, K., and Aggarwal, V.,**2018, Optimized Preference-Aware Multipath Video Streaming with Scalable Video Coding, *IEEE Transactions on Mobile Computing* 1–1.
- Famaey, J., Latré, S., Bouten, N., Meerssche, W.V. de, Vleeschauwer, B.D., Leekwijck, W.V., and Turck, F.D.,** 2013, On the merits of SVC-based HTTP Adaptive Streaming, *In 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pp. 419–426.

REFERENCES (more)

- Figuerola, A.A.M., and Favalli, L.**, 2016, Buffer Management for Scalable Video Streaming, *In Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications*, (ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)), pp. 18–23.
- Floodlight**, <http://www.projectfloodlight.org/floodlight/>. (Retrieved 12 July 2019)
- Garcia, M.-, Simone, F.D., Tavakoli, S., Staelens, N., Egger, S., Brunnström, K., and Raake, A.**, 2014, Quality of experience and HTTP adaptive streaming: A review of subjective studies, *In 2014 Sixth International Workshop on Quality of Multimedia Experience (QoMEX)*, pp. 141–146.
- Gohar, A., Raza, A., and Lee, S.**, 2019, Dynamic Adaptive Streaming Over HTTP Using Scalable Video Coding with Multipath TCP in SDN, *In 2019 International Conference on Information Networking (ICOIN)*, pp. 480–484.
- Han, B., Qian, F., Ji, L., and Gopalakrishnan, V.**, 2016, MP-DASH: Adaptive Video Streaming Over Preference-Aware Multipath, *In Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, (New York, NY, USA: ACM), pp. 129–143.
- Hosseini, S.A., Lu, Z., Veciana, G. de, and Panwar, S.S.**, 2016, SVC-Based Multi-User Streamloading for Wireless Networks, *IEEE Journal on Selected Areas in Communications* 34, pp. 2185–2197.
- Huang, T.-Y., Johari, R., McKeown, N., Trunnell, M., and Watson, M.**, 2014, A Buffer-based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service, *In Proceedings of the 2014 ACM Conference on SIGCOMM*, (New York, NY, USA: ACM), pp. 187–198.
- Huang, T.-Y., Johari, R., and McKeown, N.**, 2013, Downton Abbey Without the Hiccups: Buffer-Based Rate Adaptation for HTTP Video Streaming.

REFERENCES (more)

- Huang, W., Zhou, Y., Xie, X., Wu, D., Chen, M., and Ngai, E.,** 2018, Buffer State is Enough: Simplifying the Design of QoE-Aware HTTP Adaptive Video Streaming, *IEEE Transactions on Broadcasting* 64, pp. 590–601.
- Hwang, J., Lee, J., and Yoo, C.,** 2016, Eliminating Bandwidth Estimation from Adaptive Video Streaming in Wireless Networks, *Image Commun*, 47, pp. 242–251.
- Jiang, J., Sekar, V. and Zhang, H.,** 2014, Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With Festive., *IEEE/ACM Trans. Netw.*, 22(1), pp. 326–340.
- Jmal, R., and Chaari Fourati, L.,** 2019, Assisted DASH-aware networking over SDN–CCN architecture, *Photon Netw Commun*.
- Kalva, H., Adzic, V., and Furht, B.,** 2012, Comparing MPEG AVC and SVC for adaptive HTTP streaming, *In 2012 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 158–159.
- Karakus, M., and Durresi, A.,** 2017, Quality of Service (QoS) in Software Defined Networking (SDN): A survey, *Journal of Network and Computer Applications* 80, pp. 200–218.
- Kim, Y., and Chung, K.,** 2018, Multipath-based transmission scheme for improving the QoE of HTTP adaptive streaming, *Journal of Visual Communication and Image Representation* 55, pp. 12–20.
- Kleinrouweler, J.W., Cabrero, S., and Cesar, P.,** 2016, Delivering Stable High-quality Video: An SDN Architecture with DASH Assisting Network Elements, *In Proceedings of the 7th International Conference on Multimedia Systems*, (New York, NY, USA: ACM), pp. 4:1–4:10.
- Kreutz, D., Ramos, F.M.V., Veríssimo, P.E., Rothenberg, C.E., Azodolmolky, S., and Uhlig, S.,** 2015, Software-Defined Networking: A Comprehensive Survey, *Proceedings of the IEEE* 103, pp. 14–76.

REFERENCES (more)

- Kuschnig, R., Kofler, I., and Hellwagner, H.**, 2011, Evaluation of HTTP-based Request-response Streams for Internet Video Streaming, *In Proceedings of the Second Annual ACM Conference on Multimedia Systems*, (New York, NY, USA: ACM), pp. 245–256.
- Kuschnig, R., Kofler, I., and Hellwagner, H.**, 2010, An Evaluation of TCP-based Rate-control Algorithms for Adaptive Internet Streaming of H.264/SVC, *In Proceedings of the First Annual ACM SIGMM Conference on Multimedia Systems*, (New York, NY, USA: ACM), pp. 157–168.
- Laga, S., Cleemput, T.V., Raemdonck, F.V., Vanhoutte, F., Bouten, N., Claeys, M., and Turck, F.D.**, 2014, Optimizing scalable video delivery through OpenFlow layer-based routing, *In 2014 IEEE Network Operations and Management Symposium (NOMS)*, pp. 1–4.
- Le, H.T., Nguyen, D.V., Ngoc, N.P., Pham, A.T., and Thang, T.C.**, 2013, Buffer-based bitrate adaptation for adaptive HTTP streaming, *In 2013 International Conference on Advanced Technologies for Communications (ATC 2013)*, pp. 33–38.
- Liu, C., Bouazizi, I., and Gabbouj, M.**, 2011, Rate Adaptation for Adaptive HTTP Streaming, *In Proceedings of the Second Annual ACM Conference on Multimedia Systems*, (New York, NY, USA: ACM), pp. 169–174.
- Lu, X., and Martin, G.R.**, 2013, Improved Rate Control Algorithm for Scalable Video Coding, *Schloss Dagstuhl- Leibniz-Zentrum Fuer Informatik GmbH, Wadern/Saarbruecken, Germany*.
- Mok, R.K.P., Luo, X., Chan, E.W.W., and Chang, R.K.C.**, 2012, QDASH: a QoE-aware DASH system, *MMSys '12 Proceedings of the 3rd Multimedia Systems Conference*, pp. 11–22.

REFERENCES (more)

- Mondal, A., Sengupta, S., Reddy, B.R., Koundinya, M.J.V., Govindarajan, C., De, P., Ganguly, N., and Chakraborty, S.,** 2017, Candid with YouTube: Adaptive Streaming Behavior and Implications on Data Consumption, *In Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video*, (New York, NY, USA: ACM), pp. 19–24.
- Oyman, O., and Singh, S.,** 2012, Quality of experience for HTTP adaptive streaming services, *IEEE Communications Magazine* 50, pp. 20–27.
- Ozcan, S.G., and Sayit, M.,** 2019, Improving the QoE of DASH over SDN: A MCDM Method with an Intelligent Approach, *In 2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pp. 100–105.
- Ozcan, S.G., Kivilcim, T., Cetinkaya, C., and Sayit, M.,** 2017, Rate adaptation algorithm with backward quality increasing property for SVC-DASH, *In 2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, pp. 24–28.
- Petrangeli, S., Famaey, J., Claeys, M., Latré, S., and De Turck, F.,** 2015, QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming, *ACM Trans. Multimedia Comput. Commun. Appl.* 12, pp. 28:1–28:24.
- Politis, I., Tselios, C., Lykourgiotis, A., and Kotsopoulos, S.,** 2017, On Optimizing Scalable Video Delivery over Media Aware Mobile Clouds., *IEEE International Conference on Communications (ICC)*, pp. 1–6.
- Rainer, B., Lederer, S., Mueller, C., and Timmerer, C.,** 2012, A Seamless Web Integration of Adaptive HTTP Streaming., pp. 1519–23.
- Ramakrishna, M., Fernandes, R.C., and Karunakar, A.K.,** 2017, Estimation of Adaptation Parameters for Scalable Video Streaming over Software Defined Networks., *Procedia Computer Science*, 115, pp. 715–22.

REFERENCES (more)

- Ramakrishnan, S., Zhu, X., Chan, F., and Kambhatla, K.**, 2015, SDN Based QoE Optimization for HTTP-Based Adaptive Video Streaming, *In 2015 IEEE International Symposium on Multimedia (ISM)*, pp. 120–123.
- Sánchez, Y., Hellge, C., Schierl, T., Leekwijck, W.V., and Louédec, Y.L.**, 2011, Scalable Video Coding based DASH for efficient usage of network resources, *A Position Paper for the Third W3C Web and TV workshop*.
- Sánchez de la Fuente, Y., Schierl, T., Hellge, C., Wiegand, T., Hong, D., De Vleeschauwer, D., Van Leekwijck, W., and Le Louédec, Y.**, 2011, iDASH: Improved Dynamic Adaptive Streaming over HTTP Using Scalable Video Coding, *In Proceedings of the Second Annual ACM Conference on Multimedia Systems*, (New York, NY, USA: ACM), pp. 257–264.
- Schwarz, H., Marpe, D., and Wiegand, T.**, 2007, Overview of the Scalable Video Coding Extension of the H.264/AVC Standard, *IEEE Transactions on Circuits and Systems for Video Technology* 17, pp. 1103–1120.
- Seufert, M., Egger, S., Slanina, M., Zinner, T., Hoßfeld, T., and Tran-Gia, P.**, 2015, A Survey on Quality of Experience of HTTP Adaptive Streaming, *IEEE Communications Surveys Tutorials* 17, pp. 469–492.
- Shamim, S.M., Shisir, S., Hasan, A., Hasan, M., and Hossain, A.**, 2018, Performance Analysis of Different Open Flow Based Controller Over Software Defined Networking, *Global Journal of Computer Science and Technology*.
- Sieber, C., Hoßfeld, T., Zinner, T., Tran-Gia, P., and Timmerer, C.**, 2013, 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)*, pp. 1318–1323.
- Stockhammer, T.**, 2011, Dynamic Adaptive Streaming over HTTP –: Standards and Design Principles., *Second Annual ACM Conference on Multimedia Systems, MMSys '11*, New York, NY, USA: ACM, pp. 133–144.

REFERENCES (more)

- Thang, T.C., Ho, Q., Kang, J.W., and Pham, A.T.**, 2012, Adaptive streaming of audiovisual content using MPEG DASH, *IEEE Transactions on Consumer Electronics* 58, pp. 78–85.
- Thenmozhi, R., Preethi, T., Sadhana, S., Shruthi, V., and Yuvarani, B.**, 2017, INTEGRATING MULTIMEDIA SERVICES OVER SOFTWARE DEFINED NETWORKING, *International Research Journal of Engineering and Technology (IRJET)*, 04, 5, pp. 622-626.
- Tomovic, S., Prasad, N., and Radusinovic, I.**, 2014, SDN control framework for QoS provisioning, *In 2014 22nd Telecommunications Forum Telfor (TELFOR)*, pp. 111–114.
- Xiang, S., Cai, L., and Pan, J.**, 2012, Adaptive Scalable Video Streaming in Wireless Networks, *In Proceedings of the 3rd Multimedia Systems Conference*, (New York, NY, USA: ACM), pp. 167–172.
- Xu, H., Huang, H., Chen, S., Zhao, G., and Huang, L.**, 2018, Achieving High Scalability Through Hybrid Switching in Software-Defined Networking, *IEEE/ACM Transactions on Networking* 26, pp. 618–632.
- Xue, N., Chen, X., Gong, L., Li, S., Hu, D., and Zhu, Z.**, 2015, Demonstration of OpenFlow-Controlled Network Orchestration for Adaptive SVC Video Multicast.”, *IEEE Transactions on Multimedia*, 17(9), pp. 1617–29.
- Yin, X., Jindal, A., Sekar, V., and Sinopoli, B.**, 2015, A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over http, *In Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, (New York, NY, USA: ACM), pp. 325–338.
- Zhang, S., Li, B., and Li, B.**, 2015, Presto: Towards fair and efficient HTTP adaptive streaming from multiple servers, *In 2015 IEEE International Conference on Communications (ICC)*, pp. 6849–6854.

REFERENCES (more)

- Zhao, M., Gong, X., Liang, J., Wang, W., Que, X., and Cheng, S.**, 2015, QoE-Driven Cross-Layer Optimization for Wireless Dynamic Adaptive Streaming of Scalable Videos Over http, *IEEE Transactions on Circuits and Systems for Video Technology* 25, pp. 451–465.
- Zhou, C., Lin, C., Zhang, X., and Guo, Z.**, 2014, A Control-Theoretic Approach to Rate Adaption for DASH Over Multiple Content Distribution Servers, *IEEE Transactions on Circuits and Systems for Video Technology* 24, pp. 681–694.
- Zou, X.K., Erman, J., Gopalakrishnan, V., Halepovic, E., Jana, R., Jin, X., Rexford, J., and Sinha, R.K.**, 2015, Can Accurate Predictions Improve Video Streaming in Cellular Networks?, *In Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications - HotMobile '15*, (Santa Fe, New Mexico, USA: ACM Press), pp. 57–62.

ACKNOWLEDGE

First of all, I would like to express my special thanks to my advisor Assoc.Dr. Müge SAYIT for her continuous support, attention, and knowledge. Her guidance helped me a lot during my contributions to the literature and researches.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Dr. Bahar KARAOĞLAN, and Assist. Prof. Dr. Korhan KARABULUT, for their encouragement, suggestions, and insightful comments.

My sincere thanks also goes to academic and administrative staff of International Computer Institute, Cemal Orhan ÇETİNKALP, Alper Tunga GÜLBAHAR, Hülya KARSLI, and Yasin KILIÇDAĞI for all kind of supports.

I thank my friend Yiğit AKDOKUR, for endless moral support, all figures and charts in this study. Also I thank my friends Tuğçe KIVILCIM and Seçkin EŞER for the stimulating discussions and being with me whenever I needed.

Last but not the least, I would like to thank my family, for supporting me spiritually throughout my life.

28/08/2019



Simge Gizem ÖZCAN

RESUME

Simge Gizem ÖZCAN

Email: ozcan.simgemizem@gmail.com

Linkedin: <https://www.linkedin.com/in/simge-gizem-%C3%B6-a5187766/>

Academic Background:

- M.Sc in Information Technology at Ege University, International Computer Institute, Izmir, Turkey. 2013 - 2019
- B.Sc in Statistics at Ege University, Izmir, Turkey. 2009 - 2013

Publications:

- Improving the QoE of DASH over SDN: A MCDM Method with an Intelligent Approach, 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN). 2019
- Rate adaptation algorithm with backward quality increasing property for SVC-DASH. IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin). 2017

Work Experience:

- Salesforce Developer @ Emakina.TR 2017 - Present
- Salesforce Developer @ Avandel 2015 - 2017