Towards QoE-Aware Dynamic Adaptive Streaming Over HTTP

by

Ashkan Sobhani

Thesis submitted in partial fulfillment of the requirements for the the degree of

PhD in Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

uOttawa

L'Université canadienne
Canada's university

Abstract

HTTP Adaptive Streaming (HAS) has now become ubiquitous, and it accounts for a large proportion of multimedia delivery over the Internet. Consequently, it poses new challenges for content providers and network operators. In this study, we aim to improve the user's Quality of Experience (QoE) for HAS using from two main approaches including client centric approach and network assisted approach.

In the client centric approach, we address the issue of enhancing the client's QoE by proposing a fuzzy logic–based video bitrate adaptation and prediction mechanism for Dynamic Adaptive Streaming over HTTP (DASH) players. This adaptation mechanism allows HAS players to take appropriate actions sooner than existing methods to prevent playback interruptions caused by buffer underrun and reduce the ON-OFF traffic phenomena, which causes instability and unfairness among competing players. Our results show that compared to other studied methods, our proposed method has two advantages: better fairness among multiple competing players by almost 50% on average and as much as 80% as indicated by Jain's fairness index, and better perceived quality of video by almost 8% on average and as much as 17%, according to the eMOS model.

In the network assisted approach, we propose a novel mechanism for HAS stream adaptation in the context of wireless mobile networks. The proposed mechanism leverages recent advances in the 3GPP DASH specification, including the optional feature of QoE measurement and reporting for DASH clients. As part of the proposed mechanism, we formulate a utility-maximization problem that incorporates factors influencing QoE to specify the optimum value of Quality of Service (QoS)-related parameters for HAS streams within a wireless mobile network. The results of our simulations demonstrate that our proposed system results in better perceived quality of video, measured by Mean Opinion Score (MOS), by almost 7% on average, while lowering the freezing period by almost 20% on average across HAS users when compared to other approaches where HAS users only rely on local adaptation logics.

Acknowledgements

First I would like to express my deepest gratitude to my supervisor, Professor Shervin Shirmohammadi, for his unrelenting support, constructive guidance and encouragement throughout my graduate studies.

Also, I would like to thank my colleague Dr. Abdulsalam Yassine who worked closely with me on the better part of my research. He has continuously put his solid technical qualifications at my disposal throughout my academic endeavor.

Moreover, I would like to thank all members of the Distributed and Collaborative Virtual Environment (DISCOVER) Laboratory, for their cooperation, support and simply being wonderful friends.

Last, but not least, I want to express my infinite appreciation and love to my wife Maryam whose consistent encouragement, support and unconditional love have taken me through all hardships incurred during my academic journey.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Definitions

3GPP	3rd Generation Partnership Project
AF	Application Function
AGO	Accumulated generation operation
AGS	Accumulation generated sequence
AIMD	Additive Increase and Multiplicative Decrease
AMBR	Aggregate Maximum Bit Rate
AMC	Modulation and Coding
APN	Access Point Name
AR	Auto Regression
ARP	Allocation and Retention Priority
AVC	Advanced Video Coding
BB	Buffer Based
CDF	Cumulative Distribution Function
CDN	Content Delivery Network
CDR	Charging Data Record
CQI	Channel Quality Indicator
CTB	Conservative throughput based
CV	Critical Value
DANE	DASH Aware Network Element
DASH	Dynamic Adaptive Streaming Over HTTP
DiffServ	Differentiated services
DOF	Degree of fulfillment
DP	Dynamic Programming
DSCP	Differentiated Services Code Point
ECDF	Empirical Cumulative Distribution Function
eNB	Evolved Node B
EPS	Evolved Packet System
ER	Efficiency ratio
EWMA	Exponential Weighted Moving Average
FLC	Fuzzy logic controller
FTP	File Transfer Protocol
GBR	Guaranteed Bit Rate
GM	Gray model

GPS	Global Positioning System
HAS	HTTP Adaptive Streaming
HTTP	Hyper Text Transfer Protocol
IFOM	IP flow mobility
IMS	IP Multimedia Subsystem
ITB	Instant throughput based
JND	Just Noticeable Difference
KAMA	Kaufman's Adaptive Moving Average
KKT	Karush–Kuhn–Tucker
LTE	Long-Term Evolution
MANE	Media-Aware Network Element
MAPCON	Multi-access PDN connectivity
MBR	Maximum Bitrate
MCS	Modulation and Coding Scheme
MLP	Multi-Layer Perceptron
MOM	Mean of maxima
MPD	Media Presentation Description
MPEG	Moving Picture Experts Group
MRE	Mean Relative Error
MVA	Mean Value Across
NAT	Network Address Translators
NBI	Normal Boundary Intersection
NLP	Nonlinear Programming
NSWO	Non-seamless WLAN offloading
OTT	Over-The-Top
P2P	Point to Point
PCC	Policy and Charging Control
PCRF	Policy and Charging Rules Function
P-CSCF	Proxy Call Session Control Function
PDN	Packet Data Network
PSNR	Peak Signal to Noise Ratio
QCI	QoS Class Identifier
QoE	Quality of Experience
QoS	Quality of Service
RAT	Radio Access Technology

RTP	Real-time Transport Protocol
RTSP	Real-Time Streaming Protocol
RTT	Round-Trip Time
SAND	Server and Network-assisted DASH
SDN	Software Defined Networking
SFT	Segment fetch time
SLA	Service Level Agreement
SP	Sub Problem
SSC	Average smoothing constant
SSIM	Structural Similarity Index
STB	Smoothed through based
SVR	Support Vector Regression
TB	Throughput Based
TCP	Transmission Control Protocol
TFT	Traffic Flow Template
TVSQ	time-varying subjective quality
UDP	User Datagram Protocol
UE	User Equipment
URL	Uniform Resource Locator
VoD	Video on Demand
VQM	Video Quality Metric

List of Symbols

$\widetilde{r_{ijl}}$,	The rate allocated to each HAS UE
$\widehat{B_i}$	The normalized predicted buffer level
B_{ij}	Buffer level
K_{max}	The maximum number of Iterations
O_1	First objective function
r_{ijl}	Level of video bitrate
λ_0	Initial value for Lagrangian multipliers
ω_1	Weighting coefficient
B_{th}	Buffer threshold
Dl_i	The amount of delay
$Fast_{SC}$	The shortest length of look-back samples
N_{max}	Maximum capacity of resource blocks
O^u	Utopia Point
P_i	The indication flag
SP_i	The selected video bitrate
$Slow_{SC}$	The longest length of look-back samples
V_i	The video bitrate
$X^{(0)}$	A sequence of buffer levels
$Z^{(1)}$	The background sequence
f_s	A function to estimate estimated throughput
x_{ij}	Binary variable
$\bar{\Delta}$	The average of errors
ν_0	Initial value for Lagrangian multipliers
$D(r_i, r_j)$	Function of perceived quality difference between two different qualities
Di	Selected video bitrate
FD	A fixed distance
i	The number of eNBs
j	The number of UEs
K	The number of Iterations
L	The total number of existing eNBs
L(x)	A set of linguistic values
m	The available video bitrate levels

n	Number of periods
T	Duration of each segment
U(r)	Sigmoid function to model the objective quality
UE_i	The set of UEs served by the eNB
z^{th}	Anchor point
α	Step size
έ	Error Index
λ and ν	Lagrangean Penalty Variables
Φβ	Utopia line
$ER(i)$	Efficiency Ratio
$SSC(i)$	Average smoothing constant
$ST(i)$	Smoothed throughput
$T(i)$	The computed throughput of the i^{th} segment
l	The number of samples
α	Smoothing factor

Chapter 1.Introduction

1.1 Motivation

Internet video traffic is massively growing at unprecedented rates so that it is estimated that Internet video traffic including TV, Video on Demand (VoD), and P2P video streaming will account for more than 80% of all Internet traffic by the end of 2019, according to a study by Cisco [1]. Hence, video traffic is driving next-generation service provider network designs. More importantly, the study [1] also concludes that overall video on demand (VoD) traffic will be fueling more than 70% of the total video traffic crossing IP networks, and it is expected to more than double over the next five years.

In addition, there is an increasing trend for mobile video streaming, greatly attributable to the rising adoption of smartphone technologies along with recent evolutions in wireless communications [2]. Considering the stringent constraints related to bandwidth and latency requirements in video streaming and the unpredictable nature of the wireless radio channel, streaming videos over mobile wireless networks is very challenging. As a result, video traffic is also considered as a driving force for designing innovative and reliable solutions in next-generation mobile networks.

Given the global trend of increasing VoD traffic, issues pertaining to the underlying VoD network architectures, as well as data transfer within their boundaries, Internet video streaming has garnered considerable attention from the industry and academia. In general, VoD service is provided using two streaming methods, IPTV and IP Over-The-Top (OTT) Video (IP video). The former utilizes a managed delivery network and Quality of Service (QoS) is guaranteed based on a Service Level Agreement (SLA) such as IP based TV service provided by At&T (U-

verse TV), while the latter uses best-effort delivery, which is usually via the Internet and quality is not guaranteed in this type of streaming [3].

In OTT video streaming, because both the delivery network and the end user devices are unmanaged and heterogeneous, and there is no relationship or agreement between service providers and the involved network operators, improving the viewer Quality of Experience (QoE) is more challenging compared to that in IPTV. Hence, the use of adaptation in this type of service has become a common approach. Moreover, OTT services can use different transport and application protocols, e.g. RTSP, RTP, UDP, FTP, and HTTP. Because the HTTP protocol is used widely on the Internet and HTTP packets can simply travel across firewalls and Network Address Translators (NAT) without restrictions, adaptive streaming over HTTP is an cost-efficient communication technique for conveying media and video content over the Internet, so much so that most popular video hosting service providers such as Microsoft Smooth Streaming [4], Apple HTTP Live Streaming [5], Adobe HTTP Dynamic Streaming [6] , and Akamai [7] prefer using HTTP that runs over TCP. Among different HTTP streaming techniques, adaptive streaming over HTTP (Imitation of Streaming via Short Downloads), considered as a pull-based method, has become dominant recently because it uses flexible rate adaptation to deliver the highest video quality possible while ensuring better utilization of content and network resources. In pull-based methods, a player residing at a client site is in charge of downloading a video file from the server. Therefore, opposite to the traditional RTP/UDP-based streaming, the rate adaptation mechanism is located at the client side, and the player adapts the video stream in terms of bitrate and quality based on the current network context, including available bandwidth, amount of buffered video, and local conditions such as screen resolution.

The ultimate goal of all adaptation algorithms is to maximize network utilization while maintaining the highest perceived quality of streaming video, which are contradictory elements and not easy to achieve simultaneously. Furthermore, the design of an efficient adaptation algorithm can be even more complicated if the

interaction among multiple players sharing a bottleneck link is considered [5]–[7] instead of simply considering a single player.

A variety of end-user devices operating under the multiple constraints of heterogeneous mobile networks are present in the process of content delivery, such as clients' own capabilities. For instance, end-users' devices can be limited by display resolution, maximum video bitrate, or supported media formats.

1.2 Challenges and research problem

In this section, we present the challenges addressed in this study. First, owing to the nature of video streaming over HTTP, the limited number of representations with different bitrates are stored on servers. Therefore, a mismatch between real network throughput and the selected video bitrate is inevitable. Such a mismatch is the result of the innate variation in encoded video bitrate, throughput measurement uncertainties, and limited number of available video representations, and it leads to the challenge of automatized adaptation between producers and consumers to deliver the best possible content quality. Furthermore, the player might encounter playback interruptions due to sudden increases in congestion or equivalently abrupt declines in incoming throughput. To cope with potential video freezing, the playback buffer must be held at a reliable level at all times, and to do so, the adaptation logic must swiftly tune the video bitrate to a reliable level.

As pointed out in [8], a higher QoE cannot always result from simply playing a video at a higher average bitrate. In other words, allocating instantaneous Quality of Service (QoS) does not necessarily lead to better QoE, especially in the long term. Performing up and down switching between different representations of the video to keep up with network bandwidth fluctuations may cause a significant reduction in QoE [9]. Hence, an adaptation algorithm should consider the effect of these fluctuations in quality levels to minimize their negative impact.

Another issue related to dynamic adaptive streaming over HTTP (DASH) is the variety of end-user device capabilities operating under heterogeneous networks. In such a case, the adaptation algorithm uses device capability as one of the decision

factors for selecting an appropriate video representation. The design of such algorithms is very challenging considering the constraints imposed by the capabilities of various devices. In addition, measuring bandwidth-related metrics on an end-to-end path that uses different network technologies is important, especially for applications such as DASH because they use the measured available bandwidth in their network paths as a key decision factor for adaptation. To tackle such a challenge, the adaptation algorithm requires a mechanism to not only continuously measure the available bandwidth but also to adapt its measuring process based on bandwidth dynamics to yield reliable measurements.

Another major challenge in HAS is related to unfairness in terms of bandwidth utilization among multiple players interacting on a shared bottleneck link. As properly explained in [10], after the start-up phase when the playback buffer has been built-up, in order to keep the buffer at the fixed size, the HAS player downloads a new segment only if the buffer content equivalent to the content of a video segment has been consumed. Now, if the download duration is less than the segment duration, the HAS player stays idle until the segment duration expires. This ON-OFF behaviour of HAS player creates the ON-OFF traffic pattern, which is the root cause of the unfairness among competing players. This unfairness results in unnecessary oscillation and unstable behavior in competing players. An adaptation algorithm should be designed in a way that minimizes the negative effect of the ON-OFF traffic pattern, which allows competing players to converge to an equilibrium point where they equally share the bandwidth at the bottleneck link. Consequently, an adaptation algorithm should maintain the balance between stability and adaptability while dealing with the uncertainties of the transport infrastructure involved.

Lastly, the dramatic increase in video traffic consumed by mobile devices has motivated wireless network operators to consider the characteristics of multimedia services for optimizing their networks to enhance users' QoE. However, there are two major challenges that hinder the efficiency of HAS in the context of a mobile wireless network:

- HAS adaptation methods use the average network throughput measured for previously downloaded video segments as an indication of network condition and the available bandwidth. In a mobile wireless network, the radio channel conditions can significantly change over a short period of time, and due to device mobility, the measured TCP throughput does not reflect the real network condition. This leads to an inaccurate estimation of the network bandwidth.
- In mobile wireless networks, the available bandwidth for each user is proportional to the amount of radio resource allocated to that user by a scheduler operating at a base station; e.g., eNodeB in Long-Term Evolution (LTE) network. Since the scheduler does not have any knowledge regarding the streamed video, it does not take into account the content characteristics in the process of scheduling. As a result, it potentially jeopardizes the video stream's perceived quality, which is the ultimate goal of all adaptation algorithms in HAS.

Moreover, the current architecture of wireless networks only provides mechanisms to handle QoS delivery, which is not efficient for the new adaptive paradigm of HAS. Hence, those QoE-related parameters, for example, buffer level, average bitrate, and switching rate, can be used by wireless network operators to optimize their networks with aim of increasing the overall QoE. To this end, a new system architecture is required to realize the mechanisms of acquiring QoE feedback and controlling DASH clients in the application layer. Defining such a system is also very challenging because network operators are required to dynamically decide different QoS parameters according to existing resource limitations (especially radio resources) to maximize the overall QoE.

1.3 Research Goal and Objectives

Based on the aforementioned challenges, it is concluded that a research initiative is needed to explore opportunities to improve the quality of VoD services. In this regard, this thesis focuses on addressing the following technical questions:

1. How to predict the available bandwidth for HAS players? In HAS, which uses OTT service delivery, the player does not have direct access to the network information. Owing to the lack of such information, HAS clients need a reliable estimation method to predict the available bandwidth.
2. How to maintain balance between being responsive and stable? The main goal of adaptation logic, i.e. maximizing QoE, is not always aligned with other goals such as maximizing network utilization.
3. How to control the process of selecting appropriate quality levels for a HAS player to increase QoE? Operating HAS is a complex process, so its controller needs to deal with uncertain inputs and select from a limited number of representations.
4. How to provide fairness and stability for competing HAS players? As mentioned before, the ON-OFF traffic pattern can cause unfairness among competing players, and effective adaptation logic should consider the negative impact of such a pattern.
5. How can wireless mobile network operators benefit from QoE-monitoring features to optimize their networks? The new QoE-monitoring features in HAS open up a great opportunity for wireless network operators to optimize their networks with the goal of increasing the clients' satisfaction without imposing extra expenses.

1.4 Methodology approach

The approaches used for addressing the aforementioned technical challenges are as follow:

Although there are several methods for estimating the end-to-end available bandwidth in best-effort networks [11]–[13], we are interested in methods that work passively and do not impose any further communication burden on the network. To this end, we take advantage of Kaufman's Adaptive Moving Average (KAMA) [14], used mostly in financial markets to extract more understandable trends. Moreover, dynamic calculation of the smoothing factor enables KAMA to adapt itself to

different extents of bandwidth variation. For designing the rate adaptation logic, we employ a fuzzy-based controller. As opposed to conventional controllers, a fuzzy-based controller is useful for running a complex process such as HAS, which can be comprehended better using imprecise qualitative knowledge of experts rather than using precise quantitative models. In addition, the proposed controller considers both the estimated bandwidth and the buffer status to select the most appropriate video bitrate. Considering the abovementioned inputs allows the controller to maintain a balance between being responsive and stable more effectively based on the amount of buffered content. In addition, we develop a modular emulator to evaluate different adaptation algorithms under controlled and uncontrolled network conditions. This emulator is composed of a web server and a bandwidth limiter. Several studies have attempted to maximize QoE [15]–[21]. Because switching up and down between different representations can lead to QoE degradation, an adaptation approach called Additive Increase and Multiplicative Decrease (AIMD) is commonly used to improve the player's QoE smoothly instead of rough increases [8]. Accordingly, the proposed fuzzy-based controller preserves the minimum buffer length to avoid playback interruption and to minimize quality changes during the playback. Regarding the evaluation of QoE, even though metrics based on Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) are still widely used in the literature, in many cases for HAS streams; these metrics are not well correlated with the perceived visual quality of the human visual system, which is non-linear [22]. To carry out a better assessment of QoE, we also present the evaluation of different adaptation methods compared to our proposed method using an eMOS model considering the most important factors influencing the viewers' perception.

To extend our work to design a novel mechanism for the delivery of HAS services over wireless mobile networks, we take advantage of recent advances in the 3GPP DASH specification [24] as a means of QoE measurement and reporting for HAS clients. This built-in feature creates new opportunities for network operators and content/service providers to leverage QoE-related information to ensure QoE-

aware service provisioning [26]. We will formulate a utility maximization problem so that the different QoE metrics given by UE are modeled by different utilization functions, indicating the extent of the impact of allocated QoS parameters on the QoE. This is an example of a constrained optimization problem, and the estimated QoE is the objective function while the resource limitation imposed by UEs' subscription information, limited number of video representations, limitation on overall outgoing bandwidth of gateway, and radio resource limitations are constraints.

All result data provided in this thesis were captured from fully functional prototypes used in real-world field trials. For a specific HAS session, the controlling logic in terms of choosing the quality level to be delivered from the server resides on the client side. This logic is not subject to standardization and would, in this regard, represent one of the sources for differences between commercial HAS solutions. As the choice of quality level leads to different traffic bitrates toward the client and these changes occur on a semi-continuous basis, the client either directly or indirectly relates to the bandwidth available for a session. An indirect approach to this would involve applying algorithms that use information from the client's receive buffer such as filling degree and arrival rate. The direct approach to available bandwidth estimation would involve performing either active or passive measurements.

1.5 Contributions

The contributions of this thesis are as follows:

- We propose an FLC mechanism for HAS that dynamically adapts the rate of incoming video and makes intelligent decisions for downloading subsequent video segments. This is rather unique because current mechanisms either take into account the measured available bandwidth, which make them be too sensitive to network bandwidth fluctuations, or they take into account the buffer size of the playback and minimize the fluctuations, but the player remains sluggish to network changes. Our method tackles the challenge of defining buffer

thresholds by considering the fuzzy aspects of buffer thresholds.

- Our proposed mechanism progressively downloads segments as long as there is a representation with a video bitrate higher than the estimated available bandwidth. This is important because it eliminates the ON-OFF traffic pattern, which causes instability and unfairness behavior of competing players and, negatively affects QoE. Our mechanism allows the congestion controller at the transport layer to stay active and, consequently, as shown in our experimental results, the players using the proposed method share the available bandwidth fairly compared to the sharing in other existing methods.
- We propose, for the first time, the use of Kaufman's Adaptive Moving Average (KAMA) to enable HAS players to be responsive to steady bandwidth variation while not reacting to unnecessary short-term bandwidth fluctuations. In other words, players maximize network utilization without continuously changing video bitrate. This further reduces up and down switching between different representations and leads to a significant increase in QoE.
- We propose a Grey-model predictor, which allows the FLC to make decisions based on the predicted buffer level. This means the FLC can proactively respond to the current conditions using the predicted buffer level and take an appropriate action sooner than existing methods can, especially when the chance of buffer underrun or buffer overflow is high. Buffer underrun is the main cause of playback interruptions. Our Grey-model predictor helps the controller maintain the buffer at a reliable level at all times.
- We propose a new network assisted mechanism for delivering QoE-aware HAS services over wireless mobile networks. The proposed method enables network operators to adjust the Guaranteed Bit Rate (GBR) QoS-related parameters (e.g. guaranteed bitrate and maximum bitrate) of users availing HAS services to maximize the overall QoE. To do so, the proposed mechanism benefits from a recently standardized reporting framework to collect QoE metrics. Moreover, instead of enforcing the specified QoS parameters using either the core network at the network level or eNBs at the bearer level, they are dictated by periodically

updating the MPD file sent to users during presentation. This optimized delivery of HAS services enhances service capacity within wireless networks, as well as the quality of users' perception.

- We formulate the problem of specifying the proper values of the aforementioned QoS-related parameters as a utility maximization problem that incorporates two objectives, including maximizing the overall average video bitrate and minimizing the quality level switches, which are contradicting in some scenarios. The objective functions and the corresponding constraints are defined using parameters influencing QoE, for example, buffer level, average video bitrate, and switching rate. The solution of the formulated problem allocates different shares of the available bandwidth to users to maximize the overall QoE.
- To solve the formulated problem, first we relax the problem by converting the discrete optimization problem into continuous form to take advantage of well-known continuous optimization techniques and to decrease the required computational complexity inflicted by solving the discrete problem using the dynamic programming. Following the conversion to continuous domain, to overcome the challenge of finding a set of optimal solutions in a multi-objective optimization problem, we propose using a scalarization method to form a single objective problem. Then, we relax the single objective problem by dualizing the constraint sets with Lagrange multipliers to formulate the Lagrange dual problem. This relaxation allows us to propose an iterative gradient optimization algorithm to find the nearly optimal solution in reasonable amount of time compared to the primal problem.

1.6 Research Publications

In the process of completing this work, the following publications have been submitted, accepted or published:

1.6.1 Journals

1. A. Sobhani, A. Yassine, S. Shirmohammadi, "Optimizing QoE for HAS Video Streaming in 5G Wireless Networks", submitted to Journal of Computer Communication (ComCom) (submitted Mar 2017).
2. A. Sobhani, A. Yassine, S. Shirmohammadi, "A Video Bitrate Adaptation and Prediction Mechanism for HTTP Adaptive Streaming ", in ACM Transaction on Multimedia Computing, Communications, and Applications (TOMM), pp 1-25, 2017 (published).
3. T. Su, A. Sobhani, A. Yassine, S. Shirmohammadi, A.Javadtalab, "A DASH-based HEVC Multi-view video streaming system", in Journal of Real-Time Image Processing, Journal of Real-Time Image Processing. 2016 Aug 1; 12(2):329-42 (published).

1.6.2 Refereed Conferences

1. A. Sobhani, A. Yassine and S. Shirmohammadi. (2016, December). "QoE-Driven Optimization for DASH Service in Wireless Networks".), 2016 IEEE International Symposium on Multimedia (ISM), San Jose, USA, December 2016, pp. 232-237 (published).
2. A. Sobhani, A. Yassine and S. Shirmohammadi, "A fuzzy-based rate adaptation controller for DASH," in Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (ACM NOSSDAV), Portland, USA, March 18-20, 2015 ,pp. 31-36 (published).

1.7 Road map of thesis

The road map for the rest of thesis is outlined below:

Chapter 2- Background provides the background information on HAS, and a brief background on the mobile wireless communication and constituent entities.

Chapter 3– Related Works provides an extensive review of the existing works in the

areas of methods available for media streaming, rate adaptation techniques, fairness measurement techniques and QoE in HAS.

Chapter 4– Proposed a User-Centric Adaptation Mechanism presents the proposed framework and explains its components. We present the core components of a video bitrate adaptation and prediction mechanism based on Fuzzy logic for HAS players in order to guarantee fair sharing of available bandwidth among competing players at the bottleneck link and to eliminate the ON-OFF traffic phenomena associated with current approaches and increasing the QoE .

Chapter 5– Performance Analysis of User-Centric Adaptation Mechanism presents a simulated case study with extensive test results. In particular, the experiment results show a comparison of the proposed method and other existing methods with respect to fairness in sharing the available bandwidth, quality of the streams and etc.

Chapter 6– Proposed Network-Assisted Approach for HAS Video Streaming in Mobile Wireless Networks presents the proposed system architecture in EPS network. In this chapter we explain how the new features of QoE monitoring in DASH can be utilized by wireless network operators so as to optimize their networks. Also, we talk about a general use-case scenario of the interaction of different network entities involved in the proposed mechanism.

Chapter 7– Performance Analysis of Network-Assisted Approach for HAS Video Streaming in Mobile Wireless Networks presents the evaluation results of the proposed QoE-aware optimization framework to allocate radio resource and bandwidth to HAS users in mobile wireless networks.

Chapter 8– Conclusion and Future Work provides a conclusion and an outlook on future works.

Chapter 2. Background

The various approaches can be used for transmission of content between nodes on a network. The condition of underlying network and type of content are two key points used in determination of the best methods for communication. In general, the various methods available for media streaming can be classified into three main categories: traditional, progressive, and adaptive streaming methods.

2.1 Traditional streaming methods

In traditional streaming methods, the network protocols for delivering audio and video over IP networks are employed in order to establish a session between the client and service provider. Although various network protocols for delivering audio and video over IP networks are available today, the Real-time Transport Protocol (RTP) along with the Real-Time Streaming Protocol (RTSP) are commonly used to implement the traditional approach [25].

2.2 Progressive download

Progressive download [26] typically can be realized using a regular HTTP web server rather than a streaming server. This approach has become very common and frequently used on the Internet. In this technique, the video is stored on the user's local buffer. Once there is enough multimedia content in local buffer, it starts to playback from the user' hard drive. The main issue of progressive download is playback interruption, which happens when the playback rate is higher than the download rate so that playback is interrupted until more data is downloaded. Typically, the playback delay is relatively short, but there is often a perceived delay. Beside the playback interruption, since the progressive download cannot be played until it gets downloaded to specific part, it is not able to support real-time

conversation as well as live streaming. Moreover, since the multimedia content is downloaded in local buffer, lack of security and waste of bandwidth are other drawbacks of this method. Hence, this method suffers from two major shortcomings.

First, the user or the player has to somehow select a suitable video bitrate, which may well lead to selecting a bitrate that does not match the incoming bandwidth. As a result, the user may suffer from interruptions and video freezes caused by buffer depletion. Second, users with fast Internet connections could waste huge amounts of bandwidth. For example, a user can download a long video very fast, but could stop playback after few seconds, wasting the bandwidth of the unwatched portion of the video.

2.3 Adaptive streaming technique

The adaptive streaming technique encodes multiple live or on-demand streams and switches them adaptively based upon changing user's available bandwidth and CPU capacity in order to adjust the quality of the video. This technique strives to provide the highest possible quality to the users based on their current circumstance. There are basically three adaptive streaming methods to adapt the video bitrate to variable bandwidth: transcoding, scalable coding, and stream switching.

2.3.1 Transcoding

In the transcoding method [27], the raw video is encoded once, and to support a range of required rates, the encoded video content is re-encoded to another encoding with different bitrate using the on-the-fly transcoder on the server to meet the desirable bitrate (Figure.1). This method is usually used when the raw video is not available. Since it is possible to transcode the stream based on the user's available bandwidth, this method can obtain fine granularity. It is worth to note that adapting the raw video content several times for several requests for different quality can result in high processing cost as well as poor scalability. Moreover, due to the high complexity, this method is not appropriate to be deployed in CDNs.

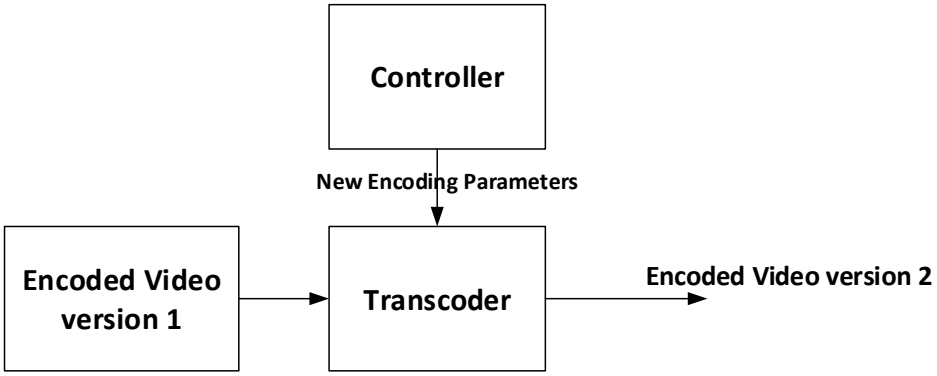


Fig. 1. Transcoding method

2.3.2 Scalable video coding

In scalable video coding [28], as depicted in Figure 2, a base layer along with a number of enhancement layers are prepared for each video sequence. The video streaming is started using the base layer and based on the available network bandwidth and the capabilities of the player device the enhancement layers can be added to the base layer to further enhance the quality of encoded video. Hence, the adapted video bit stream contains a base layer and one/multiple enchantment layer(s). The enhancement layers can be prepared by increasing the spatial resolution, video frame-rate or video quality, corresponding to spatial, temporal and quality/SNR scalability. The main drawback of this method is this approach is difficult for use in CDNs because this approach requires specialized servers implementing the adaptation logic. Moreover, the adaptation logic depends on the employed codec, thus restricting the content provider to use only a limited set of codecs [26].

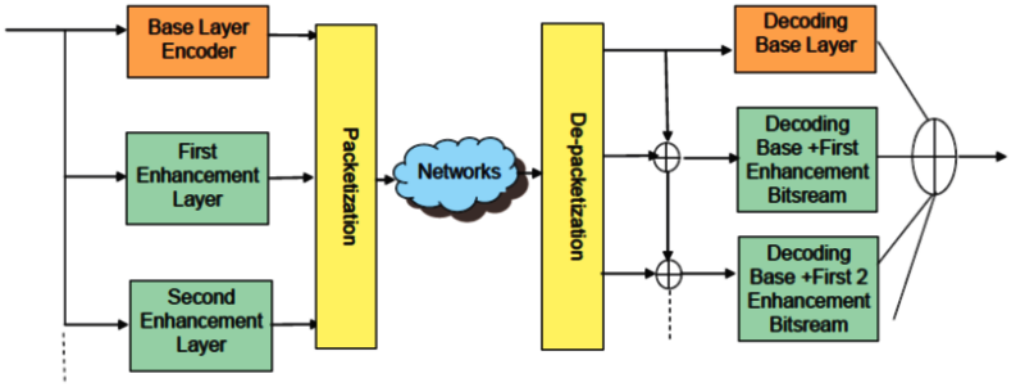


Fig. 2. Scalable Encoding [107]

2.3.3 Stream switching

As illustrated in Figure 3, to address the processing cost issue in the transcoding method, in the stream switching method the different increasing bitrates are provided by encoding the raw video at different bitrates. Hence, the video level will adapt dynamically according to the user's available bandwidth. Since this method does not require a specific codec format to be implemented, it can simply be used in CDNs. Nevertheless, this method increases the cost of storage space at the server side because the segments at different video qualities need to be stored.

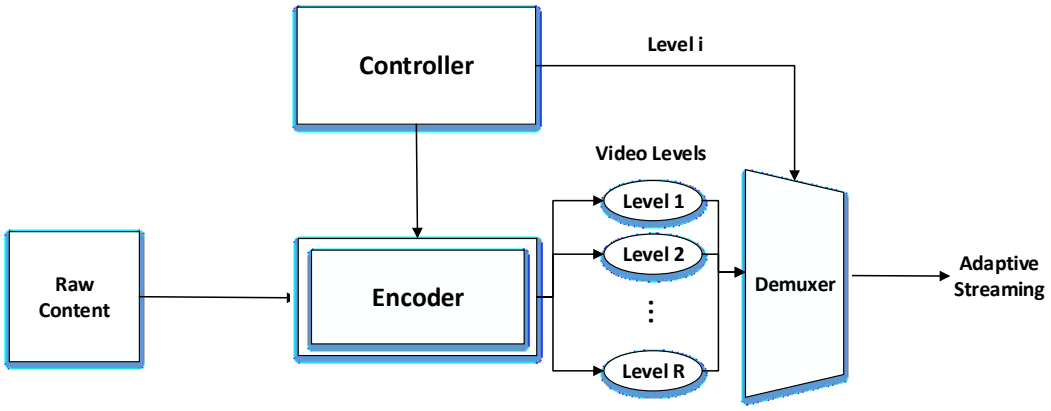


Fig. 3. Stream Switching method [29]

2.4 HTTP-based Adaptive Streaming

Recently HTTP-based adaptive streaming has become ubiquitous and accounts for a large amount of multimedia delivery over the Internet. This technique has been designed based on the stream switching method and it utilizes HTTP protocol as a session protocol, which is leveraged on the scalability of the whole Internet. Video and audio files are chunked into short segments of the same length. Eventually, all of these segments are encoded, compressed into a variety of video bitrates corresponding to different resolutions/qualities, and hosted on the HTTP server. These compressed versions are called representations, each of which is fragmented into several segments with constant duration stored at different servers. The URL addresses and the corresponding bitrates of the representations are provided by a manifest file called Media Presentation Description (MPD), which is generated during video encoding. Using the information provided in the MPD file, the player must select the best-suited video representation to increase the quality of experience. To do so, the player utilizes an adaptation logic which adapts the player's requests to the current conditions such as the available network bandwidth, the level of playback buffer, and the type of players' devices. The basic example of the HAS signaling sequence depicted in Figure 4.

The main goal of the player is to maximize the network utilization and maintain the highest perceived quality of the streaming video. However, achieving such a goal is not trivial. There are five challenges that must be resolved: Firstly, performing up and down switching between different representations of the video to keep up with bandwidth variations may lead to significant reduction in the Quality of Experience (QoE) [8]. Secondly, the player might encounter playback interruptions caused by sudden increases in congestion level or equivalently abrupt decline in incoming throughput. This means that the playback buffer must be held at a reliable level all the time and that the adaptation logic has to swiftly decrease the video bitrate to a reliable level. Third, there is always an inevitable mismatch between real network throughput and the selected video bitrate. Such mismatch is

the result of the innate variation of encoded video bitrate, throughput measurement uncertainties, and the limited number of available video representations. Fourth, selecting the right video bitrate requires a mechanism to continuously observe or predict the throughput and the buffer dynamics. Fifth, unfairness and instability are common among players competing for shared bandwidth over a bottleneck link. Therefore, video bitrate adaptation methods must be designed in a way to minimize the negative effect of the ON-OFF traffic pattern [10] and allow players to converge to an equilibrium point where they equally share the bandwidth at the bottleneck link.

HAS is now adopted by most popular video hosting service providers such as Apple HTTP Live Streaming [5], Microsoft Smooth Streaming [4], Adobe HTTP Dynamic Streaming [6], and Akamai [7].

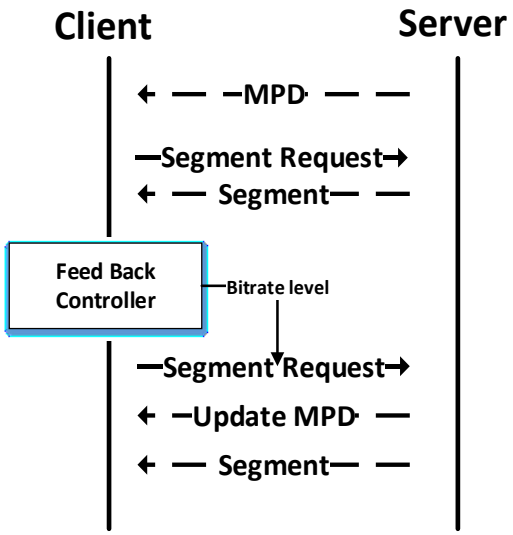


Fig. 4. General sequence of HAS signaling

2.5 Fairness in HAS

Parallel to the above works, there are several studies that focused on the momentous question of how available bandwidth should be shared among users competing for higher bandwidth over a bottleneck link. This competition can result in player instability, unfairness between users, and bandwidth underutilization.

The study [10] properly discusses the root cause of unfairness and instability, which is the well-known ON-OFF pattern of HAS flows. A video bitrate adaptation method should act in a way that players converge to an equilibrium point. To this end, the work in [61] proposes a bandwidth manager based on a traffic shaping mechanism to allocate fair bandwidth to players using home gateways. Although the experimental results show it could improve fairness among the players, the home gateway solution is not enough, as it is usually not the bottleneck link.

The work in [62] tries to justify the instability of players competing for a shared bandwidth in an undersubscribed bottle link. The authors claim that when the link is underutilized, the bandwidth estimation is not as accurate as when the link is over utilized. They also compare this effect with the well-known bandwidth cliff effect (congestion collapse), which in our opinion, is unrelated to the fluctuation in video bitrates selected by competing players. As explained in [63], in case of congestion collapse, the real throughput (Goodput) experienced by each player is much less than its fair designated throughput. But experimental results in [62] show that, after a bandwidth cliff event, the players reach their fair shares. So, it cannot be a congestion collapse.

We have a different explanation: when a sudden load is imposed on a switching node, due to the limited space in the buffer, packets are lost such that the player cannot keep up with updating its measurement; subsequently, the retransmission timer would not be updated accordingly. Consequently, the host commences sending more copies of the late packets, the buffers in the switching node are overrun, and, as a result, the new arriving packets are dropped. In this context, the player will experience throughput much less than expected. In [64], the pros and cons of different segment scheduling methods including immediate download and periodic download are presented, and randomized chunk scheduling is proposed to improve stability and fairness among competing players.

First, before scheduling the next download, the player generates a random buffer threshold within the predefined range centered at the intended buffer level. Then, if the current level of playback buffer is less than the generated threshold, the next

download will take place immediately; otherwise, the download of the next segment is scheduled for later, to the extent of the difference between the current level buffer and that random threshold. This type of scheduling strives to eliminate synchronization among downloads of different players. It should be noted that this synchronization event has been reported by [62] as well. In addition to the aforementioned works, the study [17] strives to address the fairness issue among competing players by proposing in-network proxies cooperating to facilitate fair resource sharing. On the contrary, our proposed method explained in the following chapter does not take advantage of any intermediate network elements to solve the fairness issue. We focus only on the player itself, which can be more practical in the real world, as it is more difficult to gain access to and manage the network elements in access network from a service provider's side.

2.6 Mobile wireless communications

According to [2], and supported by the recent report published by Nielsen [29], there is an increasing trend for multimedia content watched on mobile smartphones, resulting in an unprecedented increase in mobile traffic. This shift is greatly attributable to the rising adoption of smartphone technologies along with recent evolutions in wireless communications [2]. As a result, video traffic is considered as a driving force for designing innovative and reliable solutions in next-generation wireless mobile networks.

To address the challenges imposed by the stringent requirements of video applications, e.g. higher data rate and lower latencies, the Third-generation Partnership Project (3GPP) introduced an end-to-end system, called the Evolved Packet System (EPS), also referred to as Long Term Evolution (LTE). Essentially, the EPS, like its prior generation, UMTS, consists of two major network domains: the access stratum (AS) and the non-access stratum (NAS). The AS layer generally encompasses all protocols related to radio access technology i.e. Evolved Universal Terrestrial Radio Access (E-UTRAN), while the NAS layer includes the non-radio

related protocols active between the user terminals and the core network, referred to as the Evolved Packet Core (EPC) in EPS. Figure 5 shows the reference network architecture of the Evolved Packet System (EPS), which includes support for different 3GPP radio access technologies (LTE, GSM, and WCDMA/HSPA) and non-3GPP access technologies. In this architecture, each box represents a network entity that provides a set of protocols and network capabilities. In this section, we briefly explain the functional entities constituting the EPS.

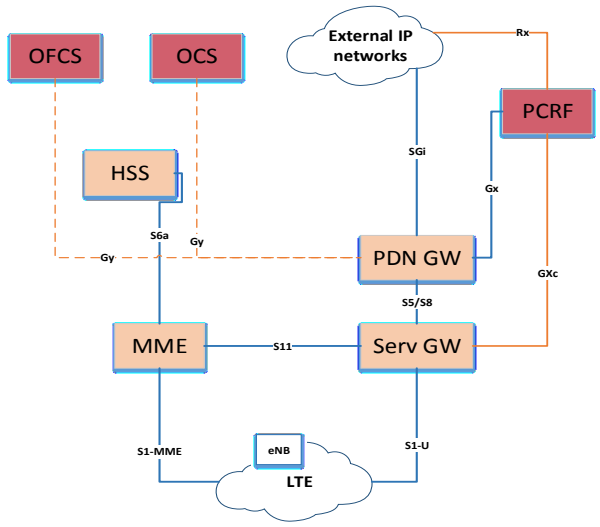


Fig. 5. Reference network architecture in EPS [30]

2.6.1 Access Stratum (AS) in EPS

Evolved NodeB (eNB): This functional entity, i.e. the base station in the LTE, is in charge of providing the E-UTRAN air interface including radio channel modulation/demodulation and channel coding/decoding and multiplexing/ demultiplexing. The procedures and functions are designed to support the interaction with the user terminals e.g. broadcasting system information, radio resource management, and enforcing the quality of service at the radio level.

One of the main ideas followed in EPS architecture is splitting the signaling of the control plane (C-plane) from the user plane (U-plane) signaling, which brings more flexibility in network deployment.

2.6.2 Evolved Packet Core (EPC) in EPS

Mobility Management Entity (MME): MME mainly is responsible to handle the control signaling, namely the NAS signaling message. Two main services are provided by mobility management and session management. The main procedures performed in mobility management are security, authentication, different types of handovers (inter and intra RAT), and paging in idle state. Session management service provides procedures for establishing, changing, and releasing default and dedicated user plane bearers.

Serving Gateway (S-GW): The S-GW in the EPS acts as a gateway which receives the user plane traffic from the E-UTRAN (radio access network) and forwards them to the EPC, and vice-versa. In addition, when the UE is moving into the neighboring cells and in the case of inter-eNB handover, the S-GW also acts as the local mobility anchor for data bearers. Moreover, when the UE is in idle mode, meaning that there are no radio bearers established, the user-plane data packets received from EPC (downlink data) are buffered at S-GW until the corresponding radio bearers are ready to forward the traffic towards the UE.

Packet Data Network Gateway (PDN-GW, or P-GW): The PDN-GW in EPS connects the core network to external IP network. The PDN-GW is in charge of allocating IP addresses to UEs, charging and controlling the user plane flows, and enforcing the transport level QoS for both uplink and downlink IP flows through marking IP packets using appropriate DiffServ codes.

Home Subscriber Server (HSS): All of users' subscription and management related parameters are stored and kept in this functional entity in the EPS. These parameters include subscribed QoS profiles, access restrictions and credentials as

well as access point name (APN) and mobility related information.

Quality of Service (QoS) and policy control in EPS:

Like other previous generations of mobile cellular systems, e.g. GSM and UTMS, as multiple services share the radio and core network resources, EPS provides an efficient QoS solution to satisfy the different QoS requirements in terms of bitrate and packet loss rate. QoS requirements are only guaranteed over the logical transport channel between the UE and the PDN, namely the EPS bearer. Moreover, the same set of QoS parameters is applied to treat all traffic travelling over each EPS bearer. This set includes the QoS Class Identifier (QCI) and the Allocation and Retention Priority (ARP) and bitrate parameters for some EPS bearers which should support a fixed bitrate. Hence, based on the requirement of supporting a guaranteed bitrate, EPS bearers can be categorized into two types, GBR bearers and non-GBR bearers. When a GBR bearer is established, a certain amount of radio capacity (bandwidth) is being reserved regardless of whether it is really used or not. However, for the non-GBR bearer, a fixed radio resource is not pre-allocated and consequently, the bitrate is not guaranteed.

To have multiple services with different QoS requirements, their corresponding IP flows should be sent over different EPS bearers. To do so, each EPS bearer is associated with a series of packet filters called Traffic Flow Templates (TFTs), which are used to extract IP packets belonging to the service mapped to the EPS bearer. As illustrated in Figure 6, TFTs can be located at both the terminal and the PDN GW for uplink and downlink directions respectively. Typically, the TFTs are created during the establishment of a new EPS bearer, and can be modified during their lifetime. These operations on EPS bearers are centrally managed and controlled by the Policy and Charging Control (PCC) system.

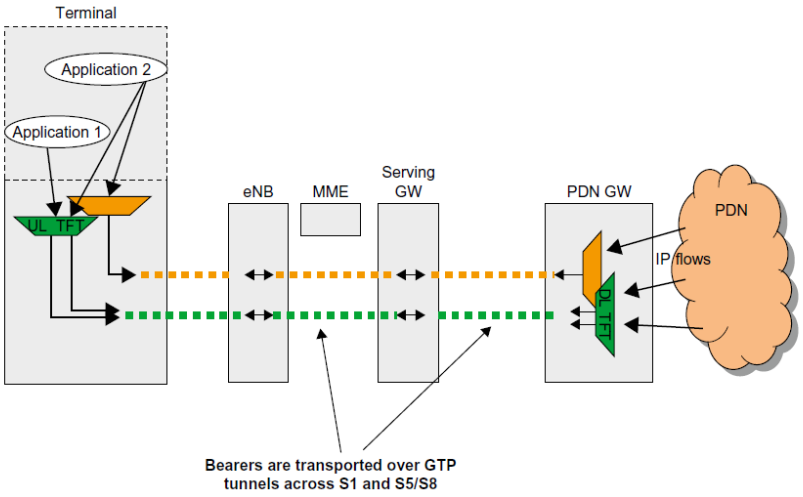


Fig. 6. TFT in EPS bearer

2.6.3 Policy and Charging Control (PCC) Architecture

PCC provides operators with an advanced control mechanism for controlling and charging QoS aware services such that the appropriate QoSs are arranged for service sessions. As illustrated in Figure 7, the PCC architecture in EPS supports multiple access technologies e.g. E-UTRAN, UTRAN and GERAN, roaming and multi-access mobility to define an access-agnostic policy control framework. The functional entities in the PCC architecture are briefly described below.

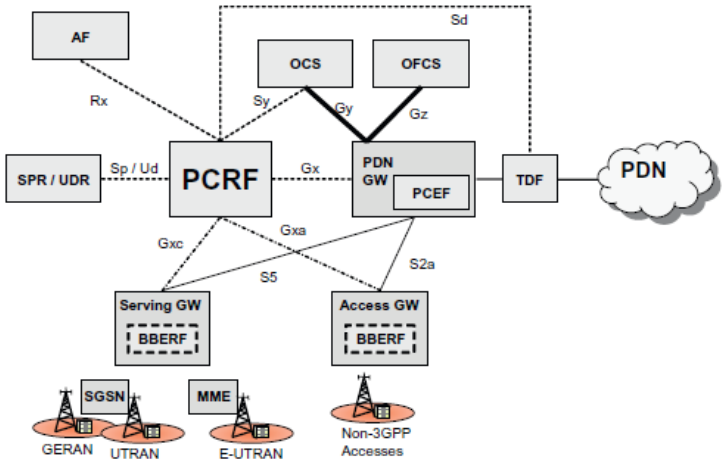


Fig. 7. The reference network architecture for PCC in EPS[30]

Application Function (AF): As defined in the 3GPP standardized PCC architecture, AF is in charge of interacting with the application running at UEs, e.g. the SIP agent and the HAS application, to extract the session-level information from the corresponding session protocols. The application signaling can either pass through the AF or is terminated at the AF. In this way, the AF can extract session-level information and provides the Policy and Charging Rules Function (PCRF) with this information to establish the transport channels with the required QoS parameters deduced from the session level information. This information is reported to the PCRF entity through the Attribute-Value-Pairs (AVPs), which are embedded in Authenticate and Authorize Request (AAR) messages over the Rx interface. For instance, the Proxy Call Session Control Function (P-CSCF) can be AF in the IP Multimedia Subsystem (IMS), and in media streaming services, the AF corresponds to a video streaming server.

Policy Control and Charging Rules Function (PCRF): The PCRF is responsible for making decisions regarding policies and authorizations in terms of QoS parameters (i.e., QoS class identifier and bit rates as discussed beforehand) to be applied in treating user-plane flows. The decisions made by the PCRF for each flow are then converted to a set of PCC rules, and sent to the Policy and Charging Enforcement Function PCEF to be enforced.

Policy Control Enforcement Function (PCEF):

The PCEF is located at the PDN Gateway, and its main responsibility is to map the rules, received from PCRF over Rx interface, to a particular ESP bearer. The PCEF is also responsible for performing policy enforcement and providing user data flow handling as well as QoS handling.

Subscription Profile Repository (SPR):

Essentially, SPR is a database for storing the subscriber related information and the corresponding QoS parameters and policies e.g. restrictions, provisioned services.

Online Charging System (OCS):

This entity manages the credits in pre-paid charging system, and responds to the requests from PCEF regarding reporting credit statuses.

The Offline Charging System (OFCS):

In case of offline charging, the charging events reported by the PCEF are handled by this entity to generate Charging Data Records (CDRs) to be transferred to the billing system.

Chapter 3. Related Works

In HAS, as mentioned before, the adaptation logic resides in the client side, as it is assumed that client has the best view of its current conditions as well as the network condition. Several research studies have proposed user-centric approaches to enhance the QoE of HAS users individually. However, in scenarios where there are multiple HAS players using a bottleneck link, as HAS players have an incomplete view of the access network, the HAS players are incapable of accurately estimating the available bandwidth, which may result in instability and unfair bandwidth allocation among the HAS players [31]. To address these issues, a number of strategies proposed in the literature try to take advantage of in-network information by defining cooperation between network elements and HAS players. In general, these approaches are referred to as network-assisted HAS. This section explores these studies and highlights the differences with our work.

3.1 User-Centric Rate Adaptation Techniques

Recently, a wide range of adaptation methods have been proposed to support adaptive HTTP streaming. In general, these adaptation methods can be divided into two broad groups [32]:

- Throughput-based methods, in which the player chooses the appropriate video representation just by considering the measured bandwidth.
- Buffer-based methods [33]–[36], in which players make decisions regarding the next video representation based on the current context of the playback buffer in order to playback the video in a smooth manner.

In this section, we review the existing literature on user-centric adaptation methods in which the rate adaptation logic deployed at the HAS player individually is used to find the most appropriate quality level among the available representations based on the current network conditions.

3.1.1 Throughput-based methods

In throughput-based methods, [4], [37], [38], the player chooses the appropriate video representation just by considering the measured bandwidth (e.g. instant throughput and smoothed throughput). As the adaptation logic in such methods does not have any information about the buffer, they are vulnerable to abrupt bandwidth changes. In addition, the innate fluctuation in the throughput causes unnecessary fluctuation in selected video bit rates. In other words, knowing the underlying network condition is complex and hence precisely measuring or estimating the TCP throughput is not trivial.

To address the uncertainty issues in estimating the TCP throughput, various methods are proposed to predict the end-to-end available bandwidth [34]; the key differences between these methods are the way to forecast the throughput and/or how to use the predicted throughput. During the past decade, a vast literature on future traffic estimation has been generated that offers a rich body of techniques for answering crucial challenges in the concept of Dynamic Adaptive Streaming over HTTP (DASH). Beside the different bandwidth estimation method discussed later, a cross-session approach proposed by [39] can be used to obtain an accurate throughput prediction for a new session using temporal and spatial similarity between the new session and other sessions within a time window, where there is no profile of measured segment throughputs for the new session. In the following, we will provide a brief overview of the most-used methods of bandwidth estimation.

3.1.1.1 Exponentially Weighted Moving Average (EWMA)

This technique [11], which is used mostly for smoothing time series data, tries to predict the future available bandwidth by recursively averaging the past throughput observations, which are weighted exponentially decreasing weights over time. In fact, it acts as a low-pass filter removing changes in high-frequency. There are different variants of this technique which are used for data with different characteristics. These variants are single, double and triple smoothing methods.

3.1.1.1.1 Single exponential smoothing method

As the name of single smoothing shows, in this form of smoothing, we use just one recursive moving average process equation (1) for prediction. In this model, it is deemed that the data fluctuates around a reasonably stable mean, and there is no trend or consistent pattern of growth.

$$\mathbf{ST(i) = (1 - \alpha) \times ST(i - 1) + \alpha \times T(i)} \quad (1)$$

Where T(i) is the computed throughput of the ith segment, ST(i) is the corresponding smoothed throughput, and $\alpha = \frac{2}{l+1}$ where (l) is the number of samples such that the smaller value of α corresponds to a longer history and vice versa.

3.1.1.1.2 Double exponential smoothing method

In the case that the data has a trend, we need to use two separate smoothing processes, equations (2) and (3), which update and smooth both the values of level and trend at the same time. Equation (2) is similar to (1) and computes a smoothed value of the data using a smoothing factor (α) as a prediction of data level, while equation (2) computes the smoothed value of average growth using smoothing factor (γ) as a prediction of trend.

$$ST(i) = (1 - \alpha) \times ST(i - 1) + \alpha \times T(i) \quad (2)$$

$$VT(i) = (1 - \gamma) \times VT(i - 1) + \gamma \times (ST(i) - ST(i - 1)) \quad (3)$$

3.1.1.1.3 Triple Exponential Smoothing method

To capture seasonality in a time series in which the data samples vary in a regular and predictable pattern, another parameter and, consequently, another equation is needed to keep and update the smoothed version of the seasonality. The set of these equations is called the "Holt-Winter" method [13]. This method has two variations, additive and multiplicative. We encourage interested readers to refer to [13] for further discussions.

Beside the discussed methods, Support Vector Regression (SVR)[12] and Multi-Layer Perceptron (MLP)[12], [40], [41] are other techniques that are used to predict the future traffic. As is well-discussed in [37, 38], although the EWMA approach has less accuracy, it is more favorable, because it works passively and does not impose any further communication burden on the network. There have been a number of research studies that employed the Exponential Weighted Moving Average (EWMA) mechanism to predict the future load on edge based on the packet-based traffic estimation to address the large fluctuations issue in video bitrate. For example [42], proposed the EWMA-based method to improve throughput and bitrate prediction. In these methods, otherwise known as instant throughput based (ITB), the measured segment throughput is computed as the ratio of a given segment's data size and the delivery duration of that segment.

As is well-discussed in [43], the throughput estimation can be computed from values of instant throughput and round trip time. In another work [44], the author proposed a crowd-sourcing method based on stored GPS coordinates and corresponding bandwidth measurements that enable the client to predict bandwidth. The main drawback of these methods is that they experience short-term fluctuations. The smoothed throughput is used to tackle the fluctuations issue; hence, this method is also known as the smoothed through based (STB) method. The smoothed throughput causes late reaction of the client to a large throughput decrease, which can be easily solved by having a large buffer [42]. Beside the STB method, authors in [42] proposed a TCP-like method that detects bandwidth changes using a smoothed throughput and switch up/down between the different representations of the content that is encoded at multiple bit rates based on a step-wise increase/ aggressive decrease method. Because the controller conservatively increases the selected bitrate step by step, this method is referred to as conservative throughput based (CTB).

3.1.1.2 Autoregressive model

Another model which can be used for prediction is Auto Regression (AR). As the term *auto*-regression shows, the bandwidth prediction is computed using a linear combination of former values of the sampled bandwidth. In this model, the number of past values of the variable to be predicted is known as an order of model. Equation (4) shows the computation of an AR model of order p.

$$\mathrm{ST(i)} = c + \sum_{j=1}^{p} \emptyset_i \mathrm{ST(i-j)} + \epsilon \quad (4)$$

c is a constant and ϵ denotes a random variable of white noise.

3.1.2 Buffer-based methods

In the buffer-based methods, players make decisions regarding the next video representation based on the current context of the playback buffer to play out the video in a smooth manner. As pointed out in [32], different ranges of buffer level are defined in buffer-based methods such that different actions can be taken by the controller accordingly. Since buffer under-run causes video freezes, which dramatically decrease the QoE, avoiding underflow at the playback buffer has the highest priority in such methods. However, determining the overflow and underflow thresholds is very challenging. Hence, it can be considered as the main drawback of buffer-based methods. The authors of [45] propose an interesting optimization problem, which aims to find the optimum buffer thresholds to maximize the QoE modeled using two metrics of the average video bitrate and playback smoothness.

In addition, [46] propose a method that makes use of a pre-computed buffer map to find the highest available video representation such that by downloading that, avoiding video freeze is guaranteed. In [38], a buffer-based method is proposed on the client side that enables clients to achieve a balance between the need for buffer stability by using the future buffer estimation. Muller et al [106] have used the same approach for the buffer estimation. Moreover, it distributes the buffer in various ranges and different actions are applied when the buffer level stays in different ranges. Jarnikov et al. in [47], employ a Markov decision process to take

an action based on future estimated buffer. While using the Markov decision process is the attractiveness of this approach, this method does not consider dynamic throughput such that it fails to minimize the on-off traffic fluctuations.

Since buffer-based methods sluggishly react to bandwidth variations compared to throughput based methods, some works, e.g. [48], leverage throughput prediction as well in order to react faster to the bandwidth variations. Also [48], [49] goes further, and proposes using the information of segment sizes provided in the MPD file to proportionally assign different weights to be used in the weighted harmonic mean download rate. Such information can be used later by the adaptation mechanism to make a more accurate decision. For instance, [48] utilizes the weighted harmonic mean of previously measured segment throughputs whose weights are proportional to the corresponding segments' sizes to improve throughput prediction.

3.1.3 Quality of experience (QoE) based methods

The core idea of HAS is the quality adaptation algorithm that controls the quality level of segments to be downloaded. There are several studies that directly aim at maximizing the QoE [18], [40], [50], [51]. Since switching up and down between different representations can lead to a reduction in QoE [19], [20], [52], some adaptation algorithms such as [34], [35], [39], [53] use Additive Increase and Multiplicative Decrease (AIMD) to improve the player's QoE smoothly instead of in rough increases. As pointed out by [54], the main reason for such a conservative step-wise approach is to reduce the risk of playback interruptions that might be possible in case of aggressive switch up. AIMD allows the player to deal with sudden increases in congestion level or equivalently abrupt declines in incoming bandwidth. The proposed adaptation method in [55] is based on model predictive control (MPC). This method considers both throughput prediction and buffer occupancy, and tries to select the next video bitrate that maximizes an objective function modeling the QoE. Although [55] proposed using a rule-based decision table to address the issue of the high computational complexity required for solving such an optimization method, preparing such a table for different sets of

representations seems impractical. In addition, it does not make a decision about imposing delay, which is inevitable under some circumstances. The authors in [56] proposed a Markov decision-based approach to capture the dynamics of a video streaming system. Even though the reward function defined in [56] considers the most important factors influencing the QoE, and a greedy algorithm was proposed to reduce the computational complexity of finding the optimal solution, it needs the channel state transition probabilities, which are difficult to acquire in heterogeneous networks and even more difficult for non-stationary channels. Therefore, the performance of this method can be compromised with an inaccurate Morkov channel model. Also, this method does not provide any policy for situations where postponing the download of the following segment is inevitable. A Q-Learning based approach was also proposed by [57] whose reward function is defined to be an indication of the QoE, and directs the client to pick the most desirable policy in order to maximize the QoE. Although the influencing factors considered by the reward function are somewhat analogous to those taken into account by our proposed fuzzy method (explained in Chapter 4), in FLC, those factors are first converted to the fuzzy domain, and accordingly, the defined fuzzy rules are being used to make decisions about the final actions, instead of using a reward function in [57].

In addition, a predication model is proposed in [20] to forecast the time-varying subjective quality (TVSQ) of rate-adaptive videos in an online manner. This model is able to conduct QoE-optimized online rate-adaptation for HAS by using the database containing the measured TVSQs prepared via a subjective study. Similarly, DeCicco et al. in [21] proposed a model of the automatic video stream-switching employed by one of these leading video streaming services along with a description of the client-side communication and control protocol to maximize the client's QoE. Another QoE-enhanced adaptation algorithm was also introduced in [58] that preserves the minimum buffer length to avoid the playback interruption and consequently to minimize the quality changes during the playback.

It should be noted that to evaluate the QoE, several traditional metrics e.g. Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM), are still widely used in the literature. However, they do not correlate well with the perceived quality of HAS users [22]. In this respect, several QoE models, [50], [52], [59], [60], have been proposed to predict the perceived quality of HAS. The average quality of selected representations, the number and magnitude of switches among different representations, and frequency and duration of freezes are considered as the most important factors having an impact on QoE in these models.

3.2 Network-Assisted Techniques

As explained before, client based approaches are not optimal since usually there are multiple HAS users competing for the same resource on the bottleneck link, which results in inaccurate bandwidth estimations measured by the HAS clients. To address the shortcomings in individually adapting HAS streams, some research studies have been conducted to define interaction between video and network elements in different types to leverage the in-network information. In this regard, MPEG introduces a new baseline architecture referred to as Server and Network-assisted DASH (SAND), illustrated in Fig. 8, that defines the standard signaling mechanism to enable the cooperation and exchange of assisting information between HAS clients and network components to manage traffic and to support complex QoS.

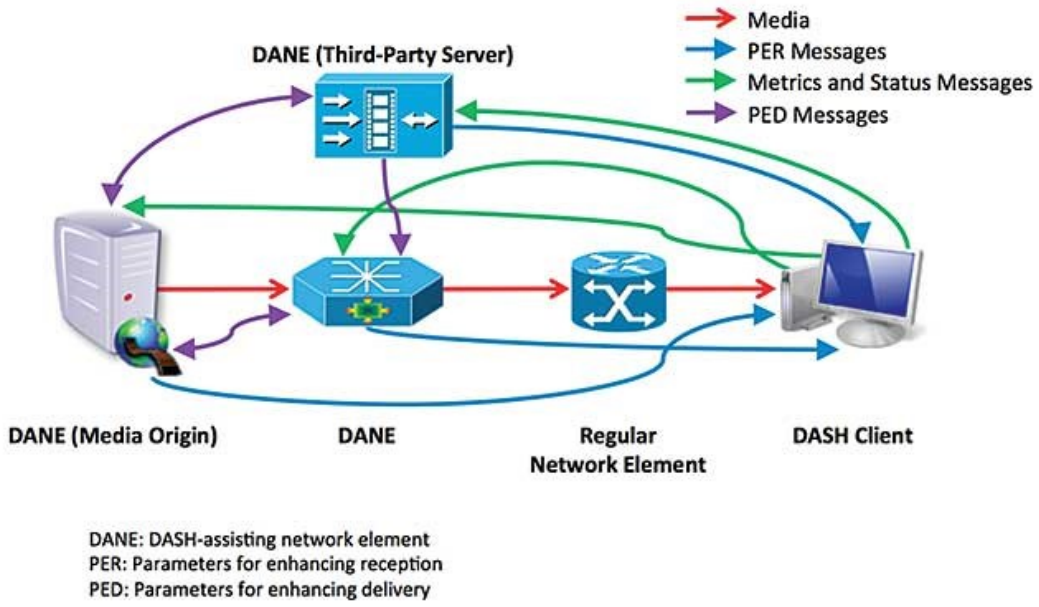


Fig. 8. SAND architecture [79]

The essential concept in SAND is a functional network entity, called DASH Aware Network Element (DANE), which is able to understand and collect the HAS session related content and to provide HAS clients with assisting parameters to enhance the service delivery. As thoroughly discussed in [65], different network-assisted strategies can be used depending on the type of access network as well as how network elements are able to interact with HAS clients. One approach is bandwidth reservation in which there is a bandwidth manager in the access network allocating bandwidth shares to HAS flows. In this method, the quality of the next video segment is determined by the client itself based on its estimates of bandwidth. Recently, since Software Defined Networking (SDN) has been emerged as a promising approach in managing networks in a central manner, it has been used in multiple network-assisted mechanisms to manage the network resources e.g. bandwidth.

In [66] Openflow controller has been proposed to prevent HAS flows from being routed through the congested links. Such approach can only be used in networks managed by a service provider whereas congestion usually occurs in access networks, which are not controlled by the proposed controller. There are also some

other works, proposed in [61] and [8] trying to confine the bandwidth allocated to each HAS client by employing traffic shaping methods. Accordingly, HAS clients are indirectly compelled to adapt their requested video bitrates to the bandwidth limits determined by the traffic shaper. As sensing changes in the available bandwidth and adjusting to the intended bitrates are slow, applying such techniques in an access network with wildly varying bandwidth is not efficient.

Similarly, the authors of [67] proposed an OpenFlow controller to maximize the QoE of all HAS players competing on the bottleneck link by allocating the fair shares of bandwidth. In another approach, referred to as bitrate guidance, there is a central network element whose function is to specify the quality of video segments for all HAS clients using the access network. It should be noted that in this strategy, the client does not have any role in the adaptation process, and just obliviously follows the decisions made by the central controller. In this regard, a few researchers, e.g. [68], [69] propose multi-user rate adaptation methods from the network side. The work in [69] considers the network management scheme to determine the bitrates for all HAS users. The study in [69] addresses bandwidth underutilization, which could lead to unfairness among DASH clients due to lack of cooperation between users and the network. In such cases, the network resource cannot be efficiently allocated, and consequently the ultimate goal, i.e. QoE, would be compromised. The proposed method in [69] is tailored for WiFi networks, and can be deployed on the access point router. Moreover, the authors of [69] have not considered the impact of video bitrate fluctuations in their cost function. The trade-off between maintaining an acceptable video streaming quality and wireless service cost is investigated in [70], [71] using the Markov Decision Process. This solution mainly addresses the problem of bitrate adaptation over multiple heterogeneous wireless access networks. In [72]–[74], the authors propose a video adaptation proxy technique to optimize multiple concurrent DASH flows in 3G networks and consequently, improve the QoE of DASH clients. A joint optimization approach is discussed in [75] that considers multiuser packet scheduling along with wireless

resource allocation in order to maximize the video quality of mobile users under delay-bound constraints.

The issue of maintaining the highest perceived quality of adaptive video streaming over wireless networks is addressed by several studies through different mechanisms. The authors in [68] propose using a streaming proxy located at the wireless base station to fairly allocate the resource to the active HAS clients. It should be noted that in wireless networks, HAS clients adapt the video quality to the available resources using the scheduler located at the base station. In [76], the authors propose a new mechanism that optimizes the adaptive HTTP media delivery to multiple clients in a wireless cell. The proposed mechanism enhances the QoE of HAS users by adjusting the throughput variations and allocates resources while considering the streamed content. The work in [77] proposes a framework somewhat similar to our optimization framework to deliver fair video quality and to achieve fair play-out buffer levels among the active HAS users connected to the same eNodeB. However, the authors of [77] assume having a media-aware network element (MANE) without defining the interactions of the core network entities in realistic scenarios. As opposed to [77], we precisely define a realistic framework based on LTE core network architecture and the standard procedures available for interaction of the involved entities. Another approach which can be identified in Network-Assisted is flow prioritization. The authors of [78] introduced the general idea of this approach for the first time such that each client independently prioritizes the next segment based on its buffer status, and informs the HAS server of these priorities when requesting the segment from the HAS server. Accordingly, the server sets the Differentiated Services Code Point (DSCP) field of all outgoing IP packets belonging to that segment proportional to the priority received along with the segment request. Then, the network elements with Differentiated services (DiffServ) capabilities apply the prioritization in the network. In this approach, although the network elements assist HAS clients to enforce their requested priorities, the network-related information is not considered in decision-making performed in clients. Furthermore, this approach suffers from another drawback

relating to fake priority requests. As there is no mechanism defined for identifying and penalizing misbehaving HAS clients in a network, there might be clients generating fake priorities, which leads to vast degradation of efficiency in applying such approach.

Chapter 4. Proposed a User-Centric Adaptation Mechanism

4.1 Main Idea

Different from the aforementioned works trying to address unfairness in HAS, the stimulating idea in our FLC mechanism is to improve the fairness and consequently the overall QoE of competing HAS players by reducing the ON-OFF traffic. Essentially, the underlying TCP congestion control mechanism was designed to achieve fairness in case of long lived contiguous TCP streams. In relation to this fact and to take advantage of the TCP congestion controller, our proposed method aims at reducing the OFF periods to the feasible extent to turn HAS streams into contiguous TCP streams. Hence, by reducing the OFF periods, the underlying TCP controllers remain active longer during the sessions, and consequently, the available bandwidth is expected to be shared more fairly among the HAS players. However, reducing OFF periods is not trivial since OFF periods usually happen when the playback buffer is almost full, and the HAS player postpones the download of the following segments to prevent buffer overflows. To reduce the OFF periods, Akhshabi et al. [108] proposed a server-based traffic shaping method aiming at eliminating OFF periods. In this method, when the server detects that a player oscillates between different video representations, the server confines the network bandwidth for the stream of each video segment to the corresponding video bitrate so that the download duration would be roughly equal to the segment duration, and consequently, the player would remain ON. This method suffers from several issues. First, it incurs extra overhead on the server, especially when the number of oscillating streams is high. Also, applying this method requires that all players receive most successive video segments from the same server which is not

the case in real-world scenarios. Furthermore, if traffic shaping is turned on all the time, the server cannot detect the bandwidth variations.

In order to address the aforementioned issues, we propose a pure client based approach to reduce the OFF periods. First we need to predict the situation in which the OFF period is created in a timelier manner and more accurately. To do so, we use the buffer level predictor as well as the bandwidth predictor to detect this situation in time that allows the FLC to take the appropriate action ahead of time. To give insight into the rationale behind our method, Fig. 9 illustrates various possible approaches which can be taken by a client where the playback buffer is almost full and there is a mismatch between available bandwidth and the highest available video bitrate (denoted by level i in Fig. 9) is lower than the available bandwidth. As shown in Fig. 9, due to this mismatch, the download of a segment belonging to level i is finished earlier than the time at which the player needs to pick another segment from the buffer (video time line boundary t2). This gap is indicated by Δ in Fig. 9. Now, if the download of the following segment is performed immediately, as illustrated in Fig. 9 top, buffer overflow condition is likely.

In another approach, which is common among the HAS players, the download of the following segment is postponed to the next video time line boundary. By doing so, the buffer level is maintained, but as can be seen in Fig. 9 middle, an OFF period is created.

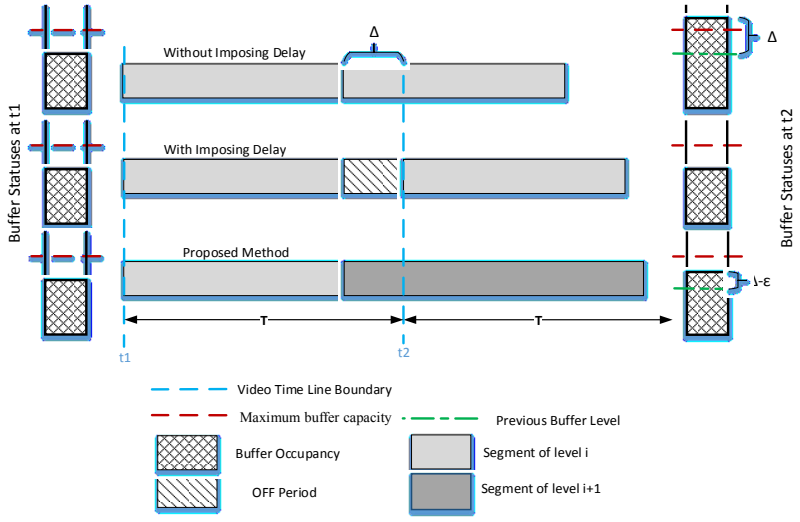


Fig. 9. Different possible approaches to download the following segment

However, as shown in Fig. 9 bottom, creation of a continuous stream is possible when there is a representation with a bitrate higher than the available bandwidth (denoted by level i+1). Immediate download of the following segment with the video bitrate greater than the available bandwidth allows for a larger segment to be downloaded for a longer time than the segment play time, T, (the difference between download time and segment play time is denoted by ε in Fig. 9). As a result, in addition to eliminating the OFF period, the buffer size is kept away from the full limit (red dashed line) compared to the first approach, which increases the risk of buffer overflow. It should be noted that our proposed method requires quality changes, which in theory may have a negative impact on QoE. However, according to the recommended representation sets given by Apple, Microsoft, and Netflix [80], video bitrates are usually selected in a way to reflect Weber's Law of Just Noticeable Difference (JND). If we assume that two adjacent video representations are spaced in such a way that the difference between them in terms of JND is less than 3 (which is common in practice [32]), the impact of quality change is not obviously noticeable to the viewer, as investigated and confirmed in [81]. Obviously, this cannot be applied where there is no available representation with a bitrate higher than the available bandwidth, so our method does not always eliminate the OFF period.

4.2 Proposed Framework

The central part of our mechanism is the fuzzy controller that takes the adaptive smoothed observed throughput and the predicted buffer dynamics as inputs. Here we introduce a system that takes two inputs for decision making: the available bandwidth estimated by means of KAMA and the buffer dynamics predicted by a Grey predictor. In Fig. 10, the high level architecture of the system is presented. We first start by introducing the details of KAMA and the Grey prediction model and then explain the FLC mechanism.

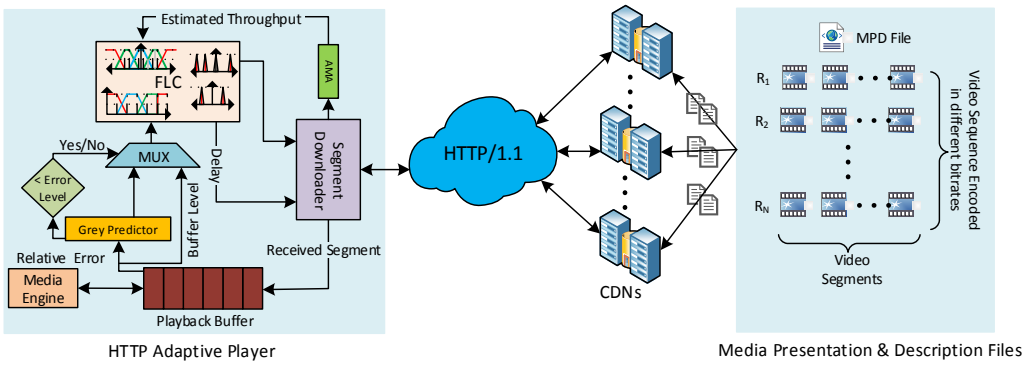


Fig. 10.Block diagram of video-streaming architecture

4.3 Kaufman's Adaptive Moving Average (KAMA)

There are several methods for estimating the end-to-end available bandwidth in best effort networks e.g.[11]–[13]. The details of these methods are discussed in section 2.5. In our proposed framework, we are interested in methods that work passively and do not impose any further communication burden on the network. One such method is the Exponential Weighted Moving Average (EWMA) [11] shown in equation (5). EWMA is usually used to smoothen the previously experienced segment throughputs so that the short-term fluctuations are canceled out, and the smoothed version of the segment throughput can be thought of as an estimation of available bandwidth.

$$ST(i) = (1-\alpha) \times ST(i-1) + \alpha \times T(i), \quad (5)$$

where $T(i)$ the computed throughput of the ith segment, $ST(i)$ is the corresponding smoothed throughput, and $\alpha = \frac{2}{l+1}$ where (l) is the number of samples such that the smaller value of α corresponds to a longer history and vice versa. However, the selection value of the smoothing factor (α) in EMWA is challenging. Using more samples, i.e. smaller α, in exponential averaging can cancel unnecessary short-term throughput variations, and as a result, minimizing unnecessary fluctuation in video quality. However, the player cannot use the maximum available bandwidth, and it

reacts sluggishly to long term bandwidth variations and needs a larger buffer size to cope with large throughput decreases. On the contrary, if fewer samples, i.e. larger α, are used in exponential averaging, the player reacts faster to varying bandwidth which means being responsive and maximizing the bandwidth utilization. But, some fluctuations would not be filtered out, which adversely impacts the QoE. In general, it would be more appropriate if the value of α is dynamically selected based on the current network condition. To this end, we take advantage of the Kaufman's Adaptive Moving Average (KAMA) concept proposed in [14], mostly used in financial markets to extract more understandable trends.

As shown in equation (6), KAMA is quite similar to EMWA with the difference that, in KAMA, the smoothing factor (C) is dynamically calculated for every sample while in EMWA, α is fixed.

$$ST(i) = (1 - \mathrm{C}) \times ST(i-1) + \mathrm{C} \times T(i) \quad (6)$$

In order to calculate C, first, the efficiency ratio (ER) has to be determined. ER is defined as the ratio of the direction of sampled throughput series to the amount of volatility in throughput sample series. The equation for ER is shown in (7).

$$ER = |Direction / Volatility| \quad (7)$$

As explained in [15], the direction and volatility can be computed using (8) and (9).

$$Direction = T(i) - T(i-n) \quad (8)$$

$$Volatility = \sum_{t=1}^{n} |T(i-t) - T(i-t-1)| \quad (9)$$

where $T(i)$ is the computed throughput of the ith segment and $T(i-n)$ is used to show the n^{th} sampled throughput ago (usually n is set to 10). Therefore, when ER is between 0 and 1, it can be thought of as the ratio of the direction. Afterwards, we establish boundaries including shortest and longest length of look-back samples, denoted by $Fast_{SC}$ and $Slow_{SC}$ respectively, for KAMA. To do so, we use the evaluation of $\alpha = \frac{2}{l+1}$ at the values of l, which are the number of samples to be considered as the length of shortest and longest history for $Fast_{SC}$ and $Slow_{SC}$,

respectively. For instance, if 2 and 30 samples are used as shortest and longest of look-back samples respectively, $Fast_{SC}$ and $Slow_{SC}$ are computed as follows.

$$Fast_{SC} = \left[\frac{1}{1+l}\right]_{l=2} = 0.667 \quad (10)$$

$$Slow_{SC} = \left[\frac{1}{1+l}\right]_{l=30} = 0.0645 \quad (11)$$

Now, ER can be used as a scale factor to compute the corresponding average smoothing constant (SSC) in the range of $Slow_{SC}$ and $Fast_{SC}$ using equation (6).

$$SSC = ER \times (Fast_{SC} - Slow_{SC}) + Slow_{SC} \quad (12)$$

In this case, if the trend of measured throughput moves sideways, ER will be close to 0, and, as it can be inferred from equation (12), SSC will be close to Slow_{SC}. On the other hand, if the measured throughput has either increasing or decreasing trend, ER will be close to 1, and the value of SSC goes towards Fast_{SC}. Finally, the smoothing factor (C) in equation (6) is determined using equation (13).

$$C = SSC^2 \quad (13)$$

Fig. 11 shows an example of sample throughputs (blue curve) along with three smoothed curves: slow and fast smoothed versions of EWMA (green and black curves) and the smoothed version using KAMA (red curve). The samples of (l) used in EWMA are 2 and 15, corresponding to fast and slow EWMA curves. As illustrated in the magnified plot, it is obvious that fast EWMA curve follows closely the original throughput curve while the slow one filters out the short-term variation. Moreover, when there is a steady trend in the original curve, the slow EWMA reacts with more latency than the fast EWMA, which is intuitive. Also, the KAMA curve reveals that although KAMA reacts much faster to the throughput variations than the slow EWMA, it cancels out the short-term fluctuations comparable to what slow EWMA does. This behavior of KAMA stems from using a variable smoothing factor which changes with respect to the network condition.

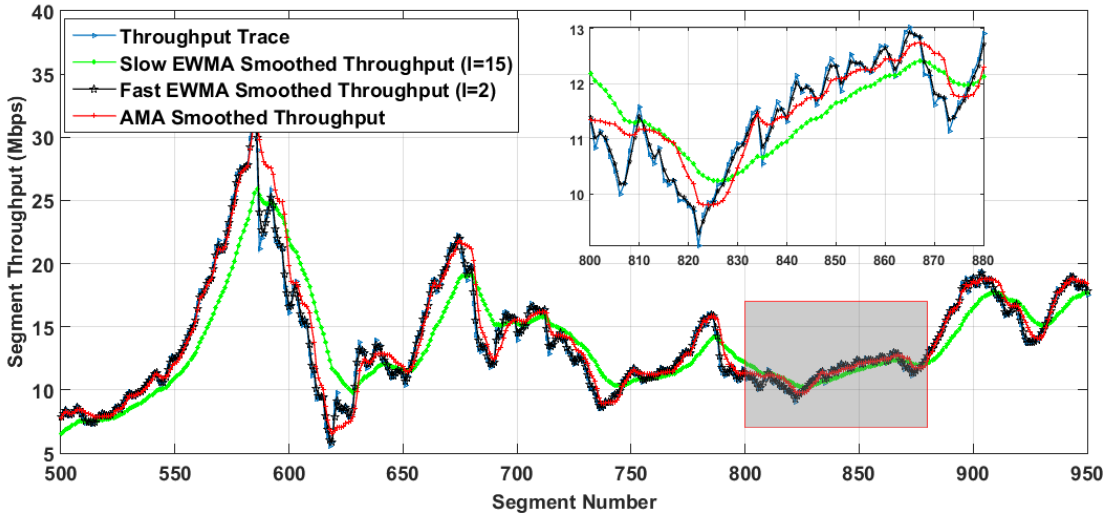


Fig. 11.An example of throughput trace along with different of smoothed ones

4.4 Grey prediction model

As mentioned earlier, buffer-based methods use the current occupied level of the buffer in order to make decisions about changing the video bitrate. In this regard, they react sluggishly to the steady variation of incoming bandwidth, and hence cannot efficiently use the available network bandwidth. Our proposed Grey predictor extracts the steady trend of the buffer level variation and predicts the buffer occupancy level ahead of time. The trend of changes in the buffer level indirectly shows the trend of variations in incoming network bandwidth. This is important as we feed such predictions to the FLC which tries to uninterruptedly download video segments. By so doing, the FLC can proactively respond to the current context using the predicted buffer level and take an appropriate action sooner, especially when the chance of buffer underrun/overrun is high.

We assume that the consecutive buffer levels, measured at time intervals, comprise a time series. We use GM(1,1) [82], a time series predicting model, as a predictor, because it gives reliable performance with small sample sizes. Furthermore, it can predict the future outputs of the system with acceptable accuracy. Assume that sequence $X^{(0)} = \left(x^{(0)}(1), x^{(0)}(2), \ldots, x^{(0)}(n)\right)$ is a sequence of recent buffer levels, sampled after n consecutive segment downloads. A first-order

weakening operator as defined in (14) is applied to $X^{(0)}$ to generate a sequence, denoted by $X^{(0)}D$ and shown in (15). $X^{(0)}D$ is slower than the original sequence, $X^{(0)}$, which means it decreases or increases slower than $X^{(0)}$. In the case of fluctuation, it fluctuates in a smaller range than $X^{(0)}$.

$$X^{(0)}(k)d = \frac{1}{n-k+1}\sum_{i=k}^{n} x^{(0)}(i), k = 1,2,\dots,n \quad (14)$$

$$X^{(0)}D = \left(x^{(0)}(1)d, x^{(0)}(2)d, \dots, x^{(0)}(n)d\right) \quad (15)$$

In order to find the inclination in the weakened sequence, $X^{(0)}D$, GM(1,1) produces a new sequence with less randomness, called accumulation generated sequence (AGS) and denoted by $X^{(1)} = \left(x^{(1)}(1), x^{(1)}(2), \dots, x^{(1)}(n)\right)$. The one time accumulated generation operation (1-AGO), used for producing AGS, is defined in (16).

$$X^{(1)}(k) = \sum_{l=1}^{k} X^{(0)}(l)d\,, k = 1,2,\dots,n \quad (16)$$

Then, the background sequence, $Z^{(1)} = \left(z^{(1)}(2), z^{(1)}(3), \dots, z^{(1)}(n)\right)$, is generated by weighted average of the adjacent members in AGS as shown in equation (17).

$$z^{(1)}(k) = \left(p \times x^{(1)}(k) + (1-p)x^{(1)}(k-1)\right)\,, k = 2,3,\dots,n \quad (17)$$

The coefficient p is selected from [0,1].

Now, by using a first order differential equation, a system can be modeled, as demonstrated in equation (18). This model is referred to as a basic form of the GM(1, 1) model.

$$x^{(0)}(k) + \omega z^{(1)}(k) = \psi \quad (18)$$

If we define

$$Y = \begin{bmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \vdots \\ x^{(0)}(n) \end{bmatrix} \quad (19)$$

and

$$B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \vdots & \vdots \\ -z^{(1)}(n) & 1 \end{bmatrix} \tag{20}$$

then parameters ω and ψ can be calculated using the Least Square method as follows:

$$\begin{bmatrix} \omega \\ \psi \end{bmatrix} = (B^T B)^{-1} B^T Y \tag{21}$$

Having the computed parameters ω and ψ, the predicted value of $\hat{x}^{(1)}(k+1)$ can be obtained using the solution of whitenization equation (16) of the GM(1,1), as in (23).

$$\frac{dx^{(1)}}{dt} + \omega x^{(1)}(k) = \psi \tag{22}$$

$$\hat{x}^{(1)}(k+1) = \left(x^{(1)}(1) - \frac{\psi}{\omega}\right) e^{-\omega k} + \frac{\omega}{\psi} \tag{23}$$

So the predicted buffer level at time interval $k+1$, $\hat{x}^{(0)}(k+1)$, is given as follows:

$$\hat{x}^{(0)}(k+1) = \hat{x}^{(1)}(k+1) - \hat{x}^{(1)}(k) = (1 - e^{\omega})(x^{(0)}(1) - \frac{\psi}{\omega}) e^{-\omega k} \tag{24}$$

Before using the predicted value of the buffer level, the constructed model must be evaluated to find out if the generated results are applicable. For doing so, different criteria can be used such as Mean Relative Error (MRE), Degree of Incidence, Ratio of Mean Square deviations, and Small Error Probability. Since, MRE is commonly used, it is computed after each round in order to check the validity of GM(1,1). If $\hat{X}^{(0)} = \left(\hat{x}^{(0)}(1), \hat{x}^{(0)}(2), \dots, \hat{x}^{(0)}(n)\right)$ is the corresponding sequence generated by the GM(1,1), the error sequence, $E^{(0)} = \left(\varepsilon^{(0)}(1), \varepsilon^{(0)}(2), \dots, \varepsilon^{(0)}(n)\right)$, between the actual data and the predicted data is simply computed by (25).

$$\varepsilon^{(0)}(k) = (x^{(0)}(k) - \hat{x}^{(0)}(k)) \tag{25}$$

Then the sequence of relative error, $\Delta = (\Delta_1, \Delta_2, \dots, \Delta_n)$, is computed using (26).

$$\Delta_k = \left| \varepsilon^{(0)}(k) \Big/ x^{(0)}(k) \right| \tag{26}$$

Finally, the average of these relative errors, $\overline{\Delta} = \frac{1}{n}\sum_{i=1}^{n} \Delta_i$, is used as a criterion so that the smaller $\overline{\Delta}$, the more accurate the model. In order to evaluate the model, the mean relative error (MRE) is compared with one pre-defined value called a critical value (cv). This critical value shows a level of accuracy so that if $\overline{\Delta} < \text{cv}$ then the model is accurate enough to be used; otherwise, the model is not accurate enough to be used. Table I shows the frequently used levels of accuracy [82].

Afterwards, the predicted value by the validated model is given to the Fuzzy controller for decision making. Once the new sample of buffer level becomes available, the new Grey model is re-established and revalidated using the aforementioned accuracy check. In the case of accuracy check failure, the current level of the buffer would be passed to the Fuzzy controller as an input.

Table I. Definition of accuracy levels [51]

Level of Relative Error	Critical value (cv)
1	0.01
2	0.05
3	0.10
4	0.20

To demonstrate how Grey model predicts the level of the buffer, an example is shown in Fig. 12. The curves of actual and predicted values of the buffer level are depicted and the bars plotted under the curves shows the relative error levels, as defined in Table I. As we can see from the figure, the Grey model for the most part is able to predict the behavior of buffer dynamics with acceptable range (according to Table I). However, for some segments e.g. 10 and 40 the relative error level is not in the acceptable range. In this case the predictions will be ignored, and the FLC

will take the current buffer level as an input. In this example, the relative error level was considered to be 3 (Table I) so that the prediction results which have an average relative error less than 0.1 ($\overline{\Delta} < 0.1$) will be considered by the FLC.

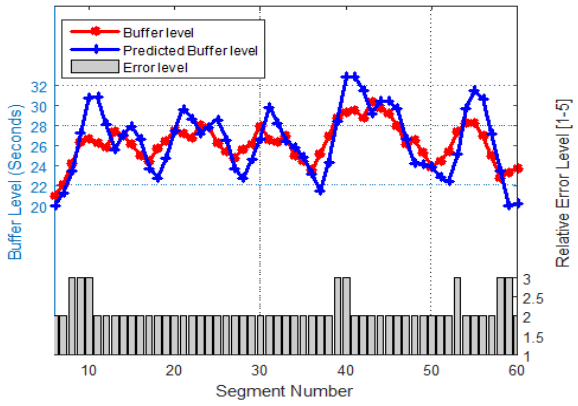


Fig. 12.Prediction results of GM(1,1) model and the corresponding relative error level along with the buffer level dynamic

4.5 FLC design

Our proposed FLC considers both the estimated throughput and the buffer status for selecting the most appropriate video bitrate and the decision on continuous download of segments. Fig. 13 shows the components of our FLC design including: Fuzzifier, fuzzy inference engine and defuzzzifier. Fuzzy-based controller is useful for running a complex process, like HAS, which can be comprehended better using imprecise qualitative knowledge of experts rather than using precise quantitative models. When Fuzzy logic is used for controlling a process, it can be seen as a person with expert knowledge controls that process. In contrast to the definition of membership in the Boolean subset, in which an element definitely either belongs or does not to a subset, in Fuzzy logic an element can be a member of a subset to some degree and the extent of being a member of the subset can be determined by a function, called membership function. In this sense, the fuzzifier is required to map the crisp input variables to the linguistic variables using membership functions. These functions are defined based on experts' knowledge for all fuzzy subsets within the numerical range of input variables. Therefore, each input variable, depending

on its crisp value, has various degrees of membership to different fuzzy subsets [83]. In our model, the Fuzzy-based controller is used for bitrate adaptation, this process is expressed as follows:

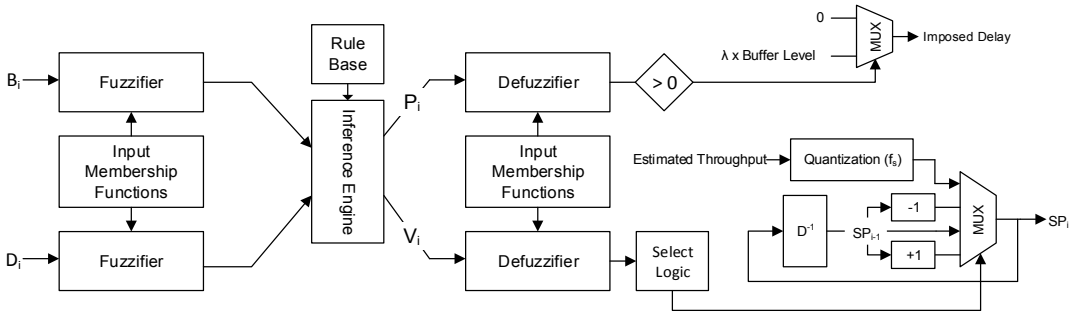


Fig. 13.Block diagram of the FLC

The first variable taken by FLC is the normalized difference between the predicted throughput and the current selected video bitrate (D_i). D_i indicates the normalized mismatch and is computed using equation (27).

$$D_i = \begin{cases} \frac{r_k - T_k}{r_k - r_{k-1}} & if\ r_k - T_k < 0\ ,\ r_k \neq r_{min} \\ \frac{r_k - T_k}{r_{k+1} - r_k} & if\ r_k - T_k > 0\ ,\ r_k \neq r_{max} \end{cases} \quad (27)$$

where r_k and T_k represent the rate of k^{th} representation and the estimated throughput after downloading the i^{th} segment respectively.

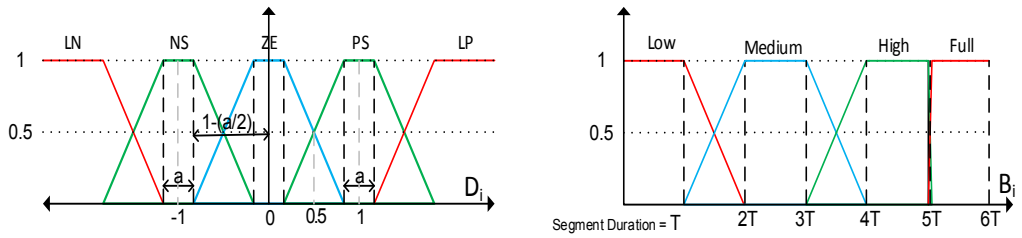
The second input to the FLC is the normalized playback buffer occupancy level ($B_i = \frac{occupied\ level\ of\ buffer}{size\ of\ buffer}$) or the normalized predicted buffer level ($\widehat{B_i} = \frac{Predicted\ occupied\ level\ of\ buffer}{size\ of\ buffer}$) based on the value of the relative error level as inputs. Crisp values of these two inputs are mapped to the corresponding linguistic variables using the membership functions, $\mu_F(D_i)\ and\ \mu_F(B_i)$, as illustrated in Fig. 14 (a) and (b). The membership function of each subset is an isosceles trapezoid where the intersection of adjacent membership functions is parameterized by the parameter a, such that $0 \leq a \leq 1$. It must be noted that the parameter a is design specific. The smaller the parameter a, the larger the intersection between the adjacent membership functions, which means more ambiguities. The larger the

parameter a, the smaller the intersection between the adjacent membership functions, which means less ambiguities. Also, the parameter T is used to parameterize the membership functions of buffer level, and it can be determined according to the segment duration. Let $L(x)$ be a set of linguistic values being mapped to all possible values associated with the measurement of crisp variable x. Then we define $L(D_i)$ and $L(B_i)$ as in (28) and (29).

$$L(D_i) = \left\{ \begin{matrix} Larg\ Negative\ \ (LN), Negative\ Small\ (NS), \\ Zero\ (ZE), Positive\ Small(PS)\ \ , Positive\ Larg\ (PL) \end{matrix} \right\} \tag{28}$$

$$L(B_i) = \{Low\ (S), Medium\ (M), High(H), Full(F)\} \tag{29}$$

Having mapped the crisp values to the linguistic values, the FLC then takes advantage of fuzzy "if/then" rules defined in Table II as a descriptive relationship in order to determine the values of the fuzzy outputs, the video bitrate (V_i) and the indication flag (P_i). It is worth noting that in this design we use the Mamdani model [53] to define fuzzy outputs.



(a) $\mu_F(D_i)$ (b) $\mu_F(B_i)$

Fig. 14.Input membership functions

Table II. Adaptation algorithm. Symbols for Di and Bi are defined in (28) and (29).

	B_i		
	S	M	H
LN	LD/ ND	De/ ND	De/ ND
NS	De/ ND	De/ ND	NC/ ND
ZE	De/ ND	NC/ ND	In/ ND
PS	NC/ ND	In/ ND	In/Dl

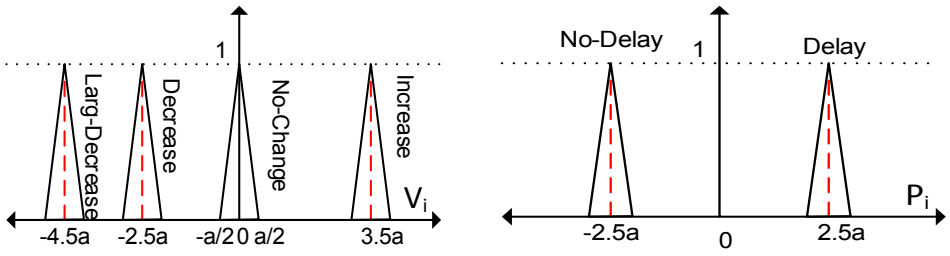
D_i	PL	In/ ND	In/ ND	In/Dl

As defined in equation (30), V_i takes four values: Increase (In), No Change (NC), Decrease (De), and Large-Decrease (LD). Increase and Decrease mean that the FLC increases and decreases the current video bitrate just by one step. Large-decrease means decreasing the video bitrate by more than one step and No-Change means the FLC keeps the current video bit-rate. In case of decreasing the video bit-rate, the FLC can either be aggressive or conservative, in the former case allowing the controller to decrease the video bitrate by more than one step to cope with a sudden drop in incoming bandwidth. Otherwise, in order to smoothly adapt the video bit-rate to the decreasing bandwidth, the controller conservatively decreases the selected bitrate step by step. In the case of increasing the video bit-rate, the FLC acts as a conservative player in order to smoothly improve the quality of video instead of abrupt improvements which degrade the QoE.

$$L(V_i) = \begin{Bmatrix} Increase\ (In), No - change\ (NC), Decrease\ (De), \\ Larg - Decrease\ (LD) \end{Bmatrix} \quad (30)$$

Equation (31) defines two possible values which can be assigned to P_i, $Delay$ and $No - Delay$. It is important to note that P_i only determines the need of download postponement, and does not specify the time of the download. If the linguistic value of P_i is $No - Delay$, the next download occurs immediately. Because the fuzzy outputs are discrete, the types of corresponding membership functions are triangular without any overlap, as shown in Fig. 15(a) and (b).

$$L(P_i) = \{Delay\ (Dl), No - Delay(ND)\} \quad (31)$$



(a) $\mu_F(V_i)$ (b) $\mu_F(P_i)$

Fig. 15.The outputs membership functions for the controller's outputs

Subsequently, the linguistic values of outputs are transformed to crisp values according to their degree of fulfillment (DOF). Mean of maxima (MOM) is selected as a method of defuzzification due to the discrete nature of the outputs [54]. Assuming that a linguistic output variable has N singleton type fuzzy subsets, denoted by k_i ($1 < i < N$), the MOM defuzzification finds the maximum DOF and returns the corresponding k_i. If there are more singletons with the same maximal DOF, then the output is calculated as the average of these singletons. Finally, the control parameters are generated based on the extracted crisp value outputs by means of equations (32) and (33):

$$SP_i = \begin{cases} f_s(ET) & if\ V_i < -3.5 * a \\ SP_{i-1} - 1 & if\ -3.5 * a < V_i < -1.5 * a\ \wedge\ SP_{i-1} > SP_{min} \\ SP_{i-1} & if\ -1.5 * a < V_i < 1.5 * a \\ SP_{i-1} + 1 & if\ V_i > 1.5 * a\ \wedge\ SP_{i-1} < SP_{max} \end{cases} \quad (32)$$

$$Dl_i = \begin{cases} 0 & if\ P_i < 0 \\ \lambda \times buffer\ level & if\ P_i > 0 \end{cases} \quad (33)$$

where $\boldsymbol{SP_i}$ and $\boldsymbol{Dl_i}$ are respectively the selected video bit-rate and the amount of delay that the scheduler has to wait before demonstrating the next segment; $\boldsymbol{f_s}$ is a function, which takes the estimated throughput as an input and gives the highest available video bitrate less than the given throughput. In equation (32), it is noticeable that for decreasing the video bitrate based on the current context such as preventing buffer under-run, the FLC has two options to be either aggressive or

conservative. For increasing the video bitrate, the FLC acts only in a conservative manner in order to smoothly improve the quality of video instead of abrupt jumps in quality improvement. Creation of continuous adapted streams, as explained in main idea subsection, is possible when there is a representation with greater video bitrate than the estimated throughput. This allows for a larger segment to be downloaded for a longer time than the segment play time and consequently decreases the buffer level. Our proposed method keeps the buffer size away from the full region (see Fig. 15), by increasing the requested video bitrate for the following downloads. In the situation that postponement of the next download is inevitable, as discussed in [64], random delay can decrease the probability of players' biasing. Hence, instead of generating a random variable, the amount of delay in equation (33) is drawn from the buffer occupancy, which is inherently stochastic. To prevent buffer over run, our design of the fuzzy rules allows the fuzzy controller to activate the flag P_i to delay the following segment download when the buffer level enters the full region, see inequality (34):

$$5 \times T \; < Buffer\ Level < 6 \times T \tag{34}$$

In addition, if we assume that the amount of delay is proportional to the buffer level, by a factor λ. After buffer depletion, the new buffer level would be $(1-\lambda) \times$ Buffer Level. Moreover, it is desirable that the new level of buffer falls in high region and holds the following inequality.

$$4 \times T \; < (1-\lambda) \times Buffer\ Level < 5 \times T \tag{35}$$

To hold both inequalities (34) and (35), λ should be in the range $\frac{1}{6} \leq \lambda \leq \frac{1}{5}$.

4.6 Summary of Work

The main goal of this chapter was to present our proposed user centric video bitrate adaptation and bandwidth prediction mechanism for HAS. Our proposed system takes into consideration the estimation of available network bandwidth as well as the predicted buffer occupancy level. Features that distinguish our system from other solutions are: First, we proposed a method to eliminate the ON-OFF traffic pattern when the estimated available bandwidth is less than the maximum

available video bitrate. Second, we applied KAMA to address the issue of selecting the proper smoothing factor based on network context. Third, we applied a prediction mechanism that allows the FLC to proactively respond to current situations using the predicted buffer level and take appropriate actions sooner, especially when the chance of buffer underrun/overrun is high. We have also shown that wrong predictions have no negative effect on the performance of the adaptation mechanism since the proposed fallback mechanism ignores accuracies that are not within an acceptable range. In next chapter, we will discuss the performance analysis of the proposed user centric video bitrate adaptation mechanism for HAS.

Chapter 5. Performance Analysis of User-Centric Adaptation Mechanism

In order to validate the proposed user-centric adaptation mechanism (discussed in chapter.4), in this section, we discuss the implementation set up and the experiments results.

5.1 Setup Configuration

We used the network topology illustrated in Fig. 16 to evaluate the proposed adaptation method against existing methods. The adaptation methods were implemented in C++ on top of Libdash sample player. At the server side, DummyNet tool [84] was used to limit the bandwidth of the bottleneck link. Also, the RedBull video sequence [85], [86] was used with 16 available representations, as another notable method proposed by [48] is also considered in our comparison as it makes decision regarding the video bitrate as well as the amount of delay to be imposed for downloading the following segment. Herein, we refer to [48] as SARA.

Table III. Video sequence characteristics

Sequence	**Frame rate (fps)**	**[Video Bit-rate (kbps),corresponding index]**	**GoP size**	**Segment duration**
RedBull	24	[100,1],[150,2],[200,3],[250,4],[300,5], [400,6],[500,7],[700,8],[900,9],[1200,10], [1500,11],[2000,12],[3000,13],[4000,14], [5000,15],[6000,16]	15	6s

We compare our results with the methods proposed in[42] and [34] as they can fairly represent the behavior of throughput-based methods and buffer-based methods, respectively. Below, we refer to these methods as TB and BB, respectively. Furthermore, we compare our method against [64], referred to as FESTIVE below, because it directly aims at addressing the problem of unfairness among competing players.

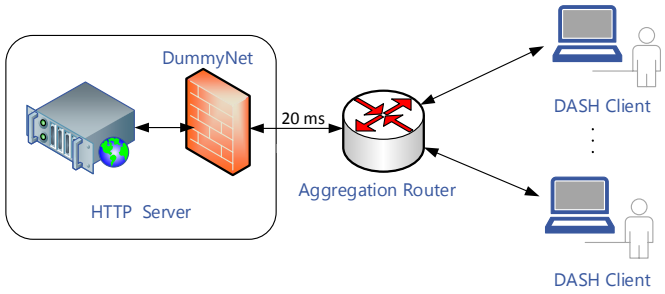


Fig. 16.Applied network topology for the test-bed

In TB, the most suitable video bitrate is chosen among the available representations based on smoothed segment throughput (defined in equation (5)) in which α is 0.3. Also, the level of pre-buffering has been considered to be 30 seconds. In the BB method, the adaptation mechanism is performed based on the current level of the playback buffer. For doing so, different buffer thresholds (B1 < B2 < Bmax) are determined to extract the buffer status and to select the most appropriate action in order to prevent buffer under run. In the experiment, we use values for B1, B2, and Bmax to be 20 seconds, 40 seconds, and 60 seconds respectively. FESTIVE uses harmonic mean to estimate the bandwidth, which is more robust for larger outliers. In addition, FESTIVE introduces the randomized scheduler so that if the playback buffer is more than a pre-defined target buffer, it postpones the download of the next segment using a random delay drawn from a randomized target buffer size. The target buffer size for FESTIVE in our setup is considered to be 20 seconds. Accordingly, the random target buffer size is drawn from a random variable ranging in (14, 26) with uniform distribution. Similar to FESTIVE, harmonic mean is also used in SARA. However, in order to have more

accurate prediction, SARA proposes weighting previously measured throughput proportionally to the corresponding segment sizes. Therefore, the segment size information should be provided to SARA along with the corresponding MPD file. Also, in our experiments, according to the segment duration, i.e 6 seconds, we set the parameters B_{α}, B_{β} and B_{max} to 10, 25 and 30 seconds respectively.

5.2 Evaluation

The defined experiment sets in this section are intended to demonstrate the dynamic behavior of the considered algorithms, and to show how our method reduces OFF periods compared to the other methods. Our experiments are divided into two sets: (1) single HAS player and (2) multiple HAS players. All of the experiments were separately conducted for our proposed method, TB, BB, SARA and FESTIVE.

In the scenario defined for the first set of experiments shown in 5.2.1, a single HAS player downloads and plays back 60 segments; i.e., 360 seconds. Also, in order to take into account the impact of cross traffic on the varying available bandwidth, in our tests, we utilized [87] as a tool to create an aggressive TCP stream. For this purpose, the client side where the HAS player resides, executes the Iperf server when the download of the 15th segment is finished, to receive the TCP flow sent from the server side where the HTTP server is located. This cross traffic is kept up for 180 seconds. In this scenario, the available bandwidth of the bottleneck link is fixed at 8 Mbps.

In the second scenario defined for the second experiment set shown in 5.2.2, 3 players compete on the bottleneck link with the fixed bandwidth during each simulation. The players start streaming randomly within the interval of the segment duration of 6 seconds. We made this assumption in order to reduce the chance of biasing. The experiment sets using the second scenario were repeated for different amounts of the fixed bandwidth ranging from 3 Mbps to 36 Mbps. Adjusting the bottleneck link's bandwidth allows us to evaluate the various degree of OFF periods imposed by our method as well as other considered adaptation

methods. It should be mentioned that the same pattern of cross traffic as described for the first scenario is used in this scenario as well.

5.2.1 Experiment Set 1

Fig. 17 shows the HAS player dynamics of all studied adaptation algorithms including Fuzzy, TB, BB, SARA and FESTIVE for the experiment set (1). As can be seen from Fig. 17 (a), (c), (e),(g) and (i), during the absence of cross traffic, all considered adaptation methods increased the video bitrate up to the maximum video bitrate available in the representation set; i.e., 6Mbps. As Fig. 17 (d) and (f) show, the playback buffer level of TB and BB remained the same when the cross traffic did not exist.

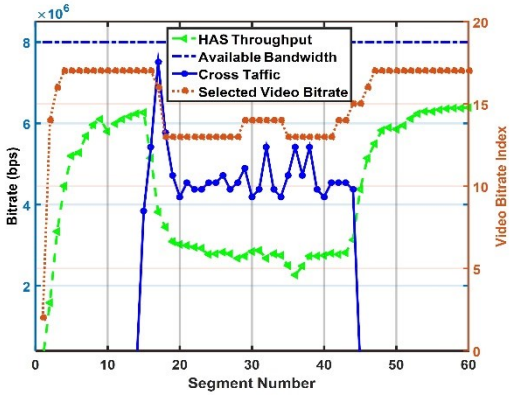
However, when the aggressive TCP stream acting as cross traffic started, the cross traffic and the HAS player started competing on the bottleneck link, resulting less bandwidth available for the HAS player. This reduction in the available bandwidth caused a sudden drop in the buffer level for all HAS players using different adaptation methods as shown in Fig. 17 (b), (d), (f), (h) and (j). As expected, the largest decrease in buffer was observed for BB (Fig. 17 (d)) as this method is sluggish to react to changes in the available bandwidth, and the smallest one happened for TB, as it follows the bandwidth changes very fast. In regard to the creation of OFF periods, it is obvious in Fig. 17 (b) that, during the competition period (referred to the period when the cross traffic exists), our method did not impose any delay while using the other methods resulted in creating OFF periods during the competition.

As illustrated in Fig. 17 (a), from segment 29 to segment 34, although the experienced segment throughput had a value in the range of (2.5 Mbps-3 Mbps) and it is expected that video index 13 be selected, our method selected the one-step higher video bitrate of video index 14, at 3 Mbps, when the corresponding buffer level was in the high region (24s – 30s), shown in Fig. 17 (b), and was going up to enter to the full region (30s – 36s). By doing so, our method kept the buffer away from being overflowed and consequently there was no need to impose any delay to

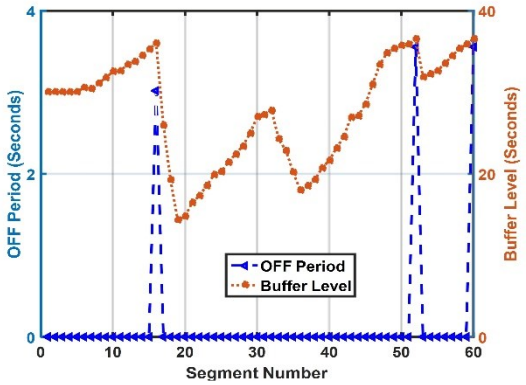
decrease the buffer level. For the buffer dynamics of TB, SARA and FESTIVE, the methods using standard bandwidth estimation, shown in Fig. 17 (d), (h) and (j), it can be seen that when the buffer was close to be full, in order to avoid the buffer overflow, these methods imposes delays to drain the buffer.

It is intuitive that if these methods did not impose any delay on segment downloading; i.e., if they aggressively downloaded, buffer overflow would have happened. It is also important to notice in Fig. 17 that the fluctuations in the cross traffic are highly correlated with the occurred OFF periods. On the contrary, as shown in Fig. 17 (b), during the downloads of the first 15 segments and last 15 segments when there was no cross traffic, since the available bandwidth (8 Mbps) is more than the highest available video bitrate of 6 Mbps, our proposed method had no means to decrease the buffer level when approaching to the full region (30s – 36s), and so it went into OFF mode.

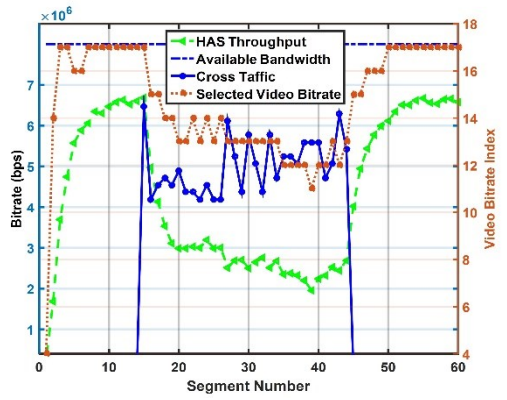
It is also worth mentioning that even though FESTIVE uses randomized delays, as shown in Fig. 17 (i), it produced considerable fluctuations in the experienced segment throughput and subsequently instability in the adapted video bitrate. In fact, FESTIVE just aims at eliminating biasing among HAS players competing on the bottleneck link, and not reducing the OFF periods. Hence, in scenarios where there is a cross-traffic TCP stream competing with the HAS stream, it would not be very efficient, and causes instability.



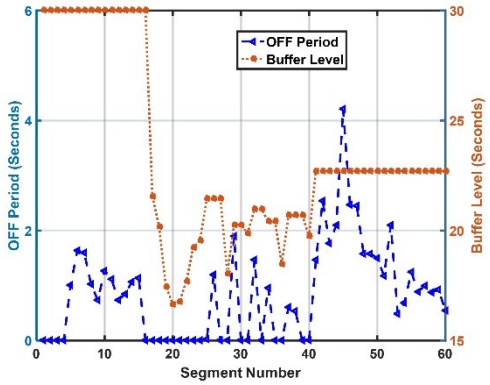
(a)



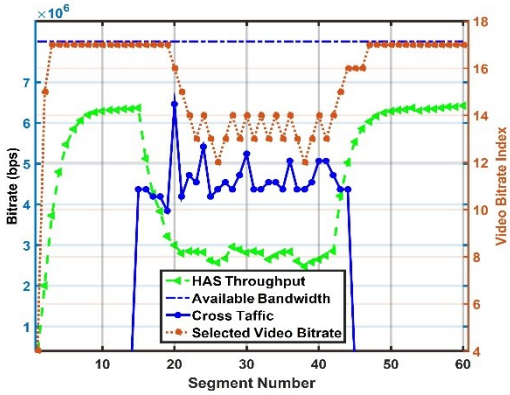
(b)


×10^6
HAS Throughput
Available Bandwidth
Cross Taffic
Selected Video Bitrate
Bitrate (bps)
Video Bitrate Index
Segment Number

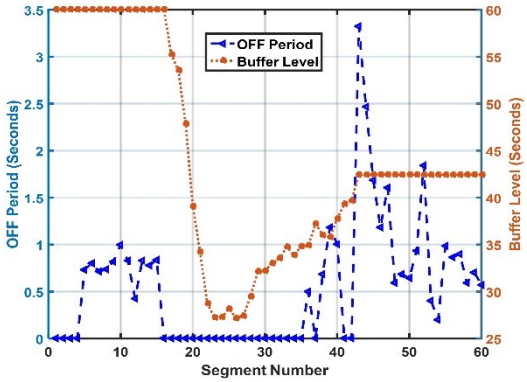
(c)


OFF Period
Buffer Level
OFF Period (Seconds)
Buffer Level (Seconds)
Segment Number

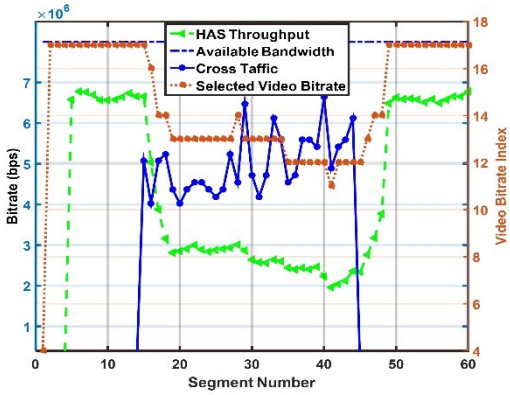
(d)


×10^6
HAS Throughput
Available Bandwidth
Cross Taffic
Selected Video Bitrate
Bitrate (bps)
Video Bitrate Index
Segment Number

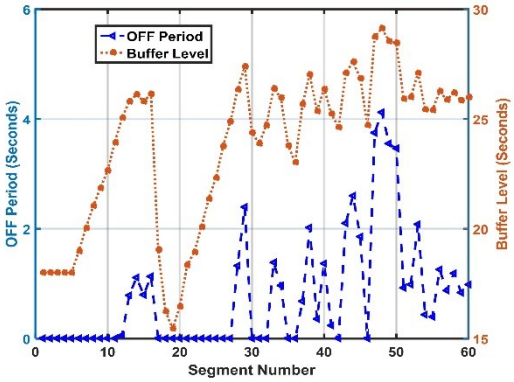
(e)


OFF Period
Buffer Level
OFF Period (Seconds)
Buffer Level (Seconds)
Segment Number

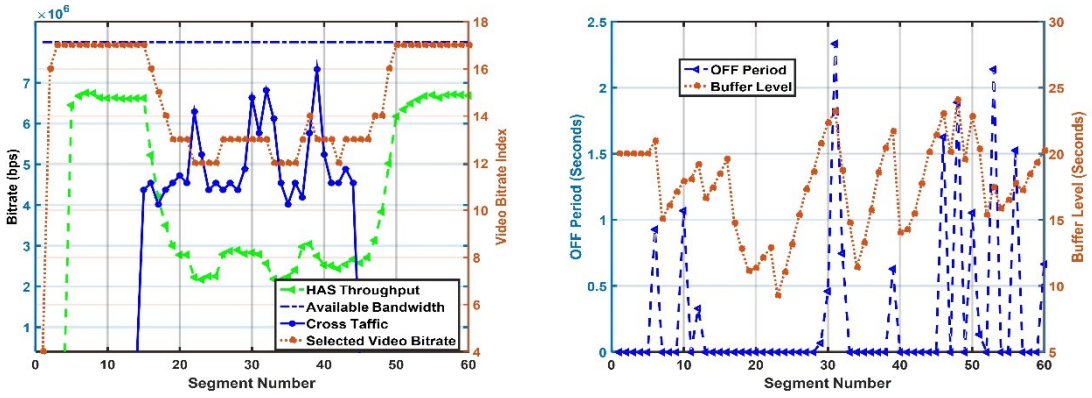
(f)


×10^6
HAS Throughput
Available Bandwidth
Cross Taffic
Selected Video Bitrate
Bitrate (bps)
Video Bitrate Index
Segment Number

(g)


OFF Period
Buffer Level
OFF Period (Seconds)
Buffer Level (Seconds)
Segment Number

(h)



(i) (j)

Fig. 17. HAS player dynamics of adaptation algorithms including experienced segment throughput, selected bitrate, and cross traffic of Fuzzy, TB, BB, SARA and FESTIVE shown in (a), (c), (e),(g) and (i) respectively, and OFF period and buffer status of Fuzzy, TB, BB, SARA and FESTIVE shown in (b), (d), (f),(h) and (j) respectively.

5.2.2 Experiment Set 2

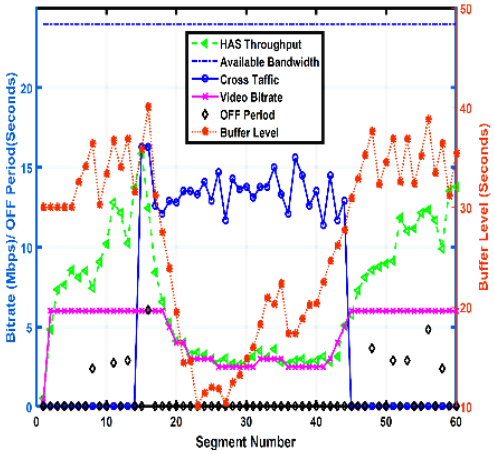
The main purpose of the second set of experiments is to measure the degree of OFF period by adjusting the bottleneck link's bandwidth when there are 3 players plus cross traffic competing on the bottleneck link. To better illustrate how the players compete for available bandwidth, the system dynamics of the 3 homogeneous players are shown in Fig. 18 to Fig. 22 for the Fuzzy, TB, BB, SARA and FESTIVE adaptation methods respectively. The bandwidth of the bottleneck link was set to 24 Mbps. When the aggressive cross traffic started, it took almost half of the available bandwidth at 12 Mbps. This phenomenon can be justified by the fact that the Iperf uses persistent connections while the HAS players use non-persistent connections. Hence, in the absence of cross traffic, the fair share of available bandwidth for each HAS player was 8 Mbps, and in the presence of cross traffic it dropped to 4 Mbps. Accordingly, it can be seen from Fig. 18 to Fig. 22, that all HAS players were able to reach the maximum video bitrate available in the video representation set; i.e., 6 Mbps. As illustrated in Fig. 18, the HAS players running our proposed method fairly share the bottleneck bandwidth when operating along with the cross traffic, and the curves of segment throughput experienced by the 3 players are relatively smoother. On the contrary, due to the OFF-period occurrence in FESTIVE, we see

the unfairness in the segment throughputs; see Fig. 22. Moreover, in Fig. 20, it can be easily observed that even though the HAS players using the BB method have not experienced any OFF period when sharing the bottleneck link with cross traffic, the segment throughput curves show severe fluctuations and unfairness among the HAS players. In the BB method, the players reacted slowly against available bandwidth variations as they were waiting for the buffer level to go below the predefined threshold. For instance, the cross traffic started at the 15th segment while all of HAS players started responding to this variation after the 20th segment. In order to restore the buffer level to the reliable threshold and to prevent from probable video freezes, they had to select bitrates lower than the fair share of bandwidth and consequently caused instability problems. This is a known problem of BB schemes and is also reported in[88].

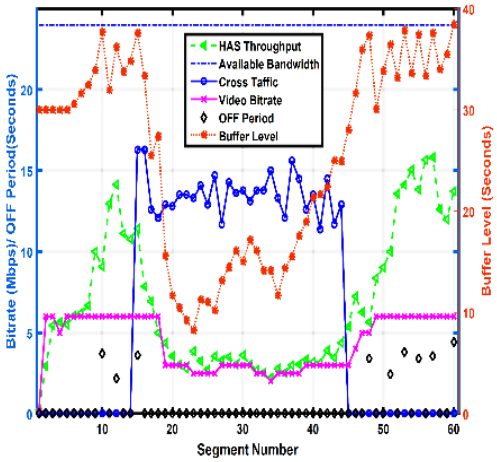
Table IV summarizes the overall performance of the considered adaptation methods in terms of the average video bitrate as well as the number of switches in the video level experienced by the 3 HAS players. In contrast to other adaptation methods, the players using our proposed method faced lower number of switches among different representations. Also, It can be seen that the Fuzzy and the TB methods led to higher average video bitrates compared to BB, SARA and FESTIVE.

Table IV. Performance comparison of the rate adaptation methods

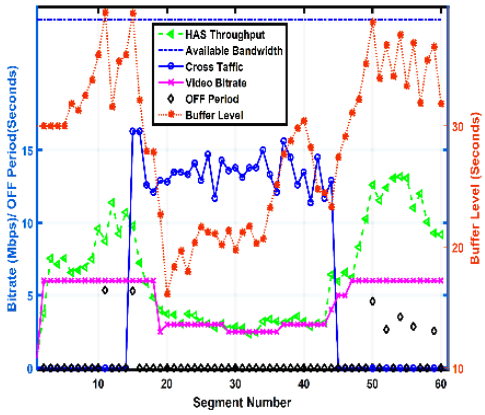
	The average video bitrate (Mbps)	The number of video quality switches
Fuzzy	4.607	9.66
TB	4.762	15.66
BB	4.509	22
SARA	4.468	16
FESTIVE	4.533	18.6



(a)Player 1

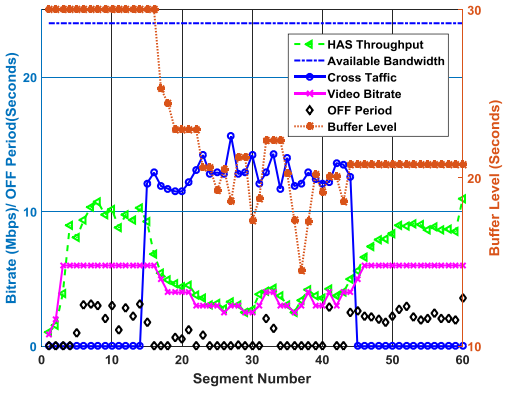


(b)Player 2

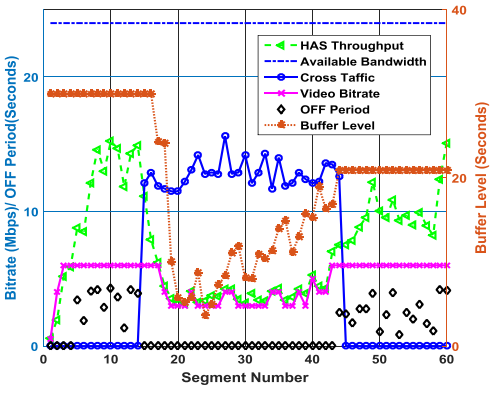


(c)Player 3

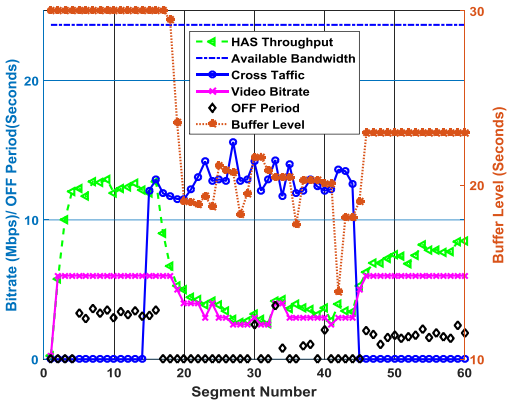
Fig. 18.System dynamics of HAS players using Fuzzy method



(a)Player 1

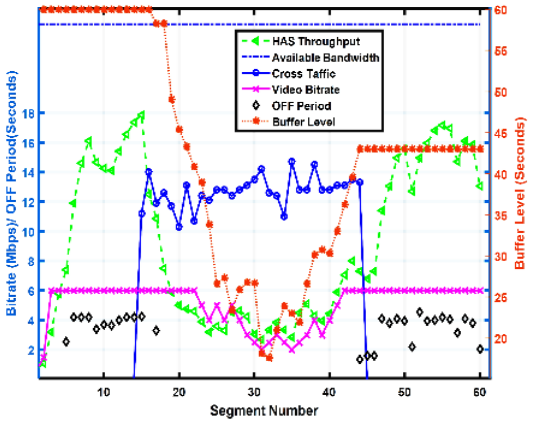


(b)Player

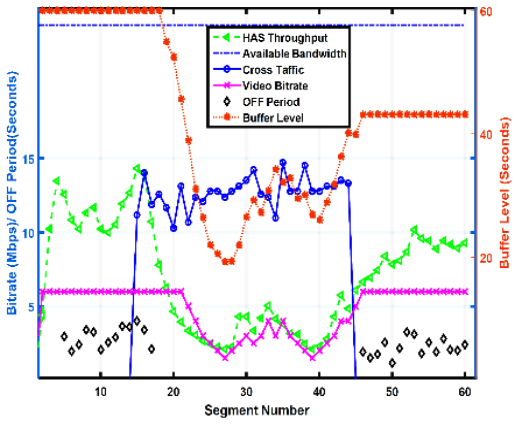


(c)Player

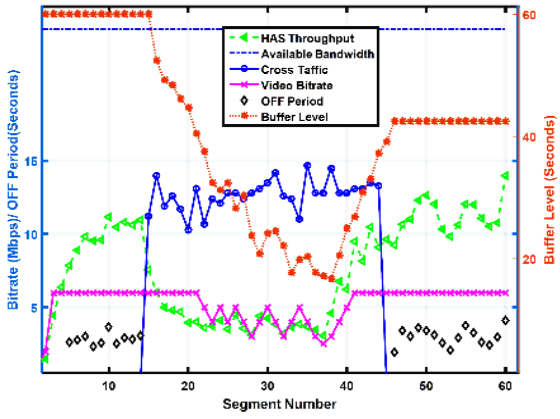
Fig. 19.System dynamics of HAS players using TB method



(a) Player 1

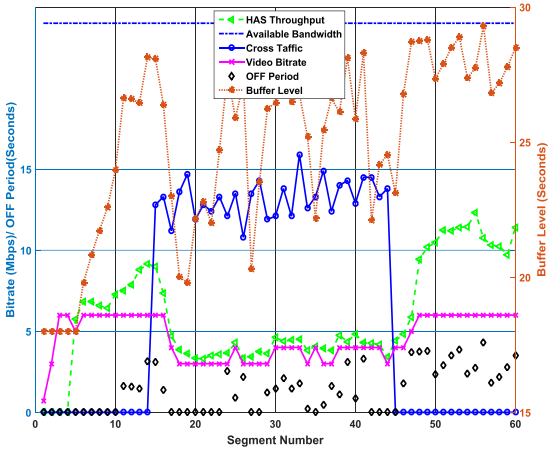


(b) Player 2

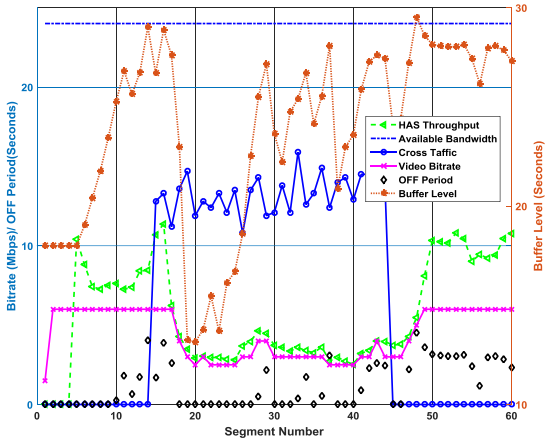


(c) Player 3

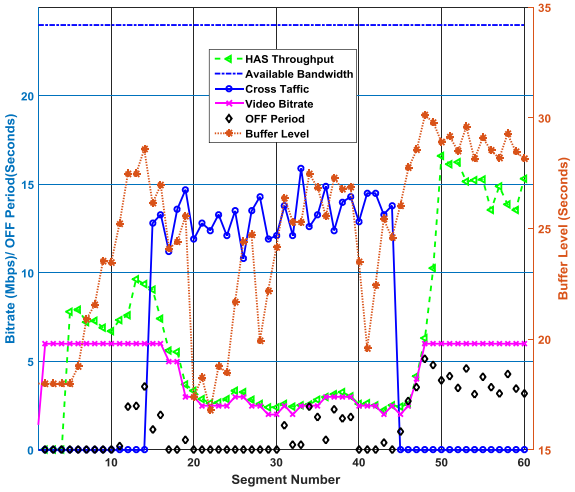
Fig. 20.System dynamics of HAS players using BB method



(a) Player1

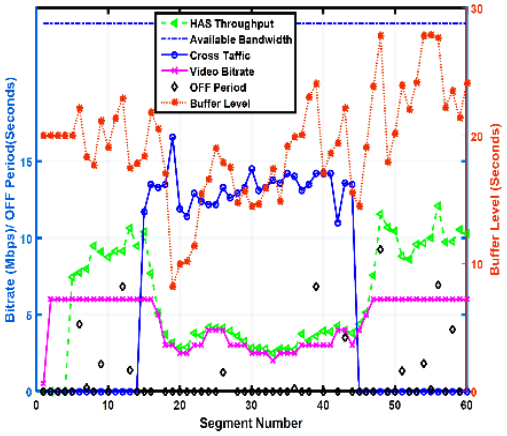


(b) Player 2

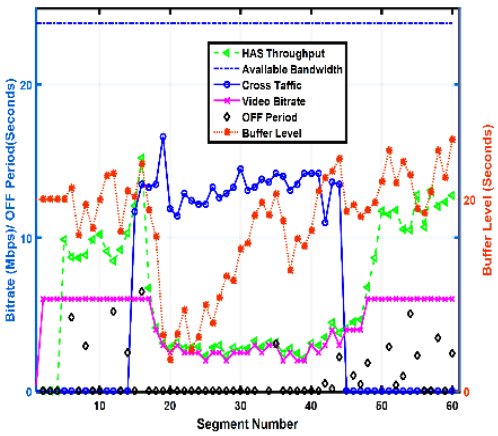


(c) Player 3

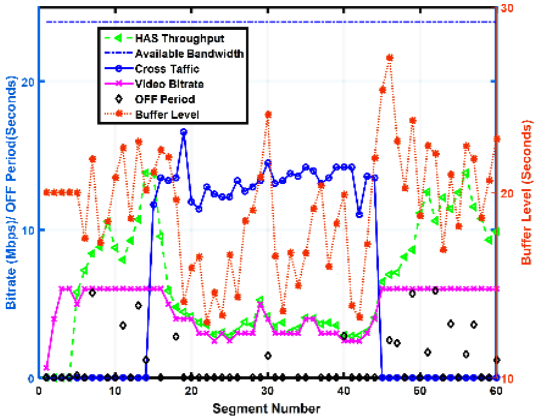
Fig. 21.System dynamics of HAS players using SARA method



(a) Player 1



(b) Player 2



(c) Player 3

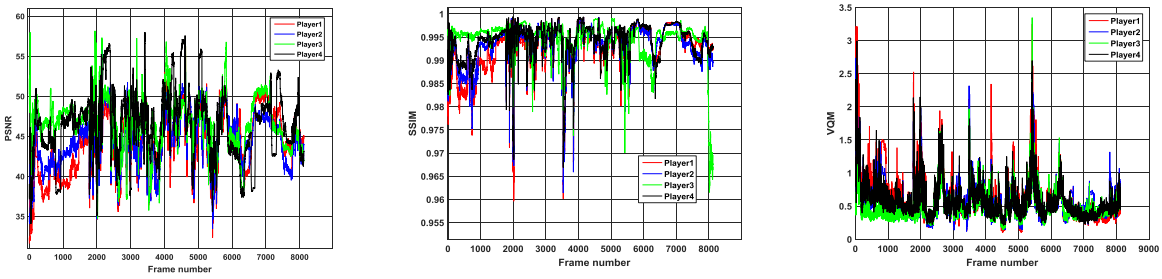
Fig. 22.System dynamics of HAS players using FESTIVE method

5.2.3 QoE analysis

In this section we present the evaluation of different adaptation methods compared to our proposed method in terms of QoE. It should be mentioned that performing subjective quality experiments based on ITU.T recommendations including J.343.4 and P.1203 are necessary to capture the real quality of adapted HAS streams perceived by end users. However, as they are expensive and time-consuming to perform, and particularly in HAS, any single test scenario is limited in scope and it requires too many scenarios to cover all combinations, we decided to use objective quality models as an alternative to subjective tests. It is worth mentioning that objective quality assessment comes with some limitations that should be considered. Firstly, the scale of objective MOS is generally different from

subjective MOS. Basically, the scale of subjective MOS is determined by the number of content quality and impairments present in a given subjective test while the scale of objective MOS can be infinite. Another limitation is related to the nature of subjective MOS which is a qualitative value obtained from statistical measurements. On the contrary, the MOS value estimated by objective quality models is a precise number and it can be considered valid as long as falls in the confidence interval of the subjective MOS value. This means that objective MOS should be declared with a confidence interval.

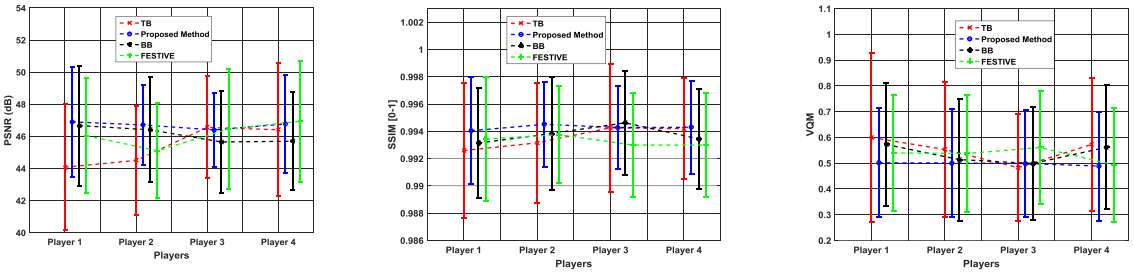
We first evaluate the quality of the streams in offline mode in which the objective quality scores are measured using three objective quality assessment metrics, Video Quality Metric (VQM) [59], Structural Similarity Index (SSIM) [60] and Peak Signal to Noise Ratio (PSNR) for the adapted streams. We used the MSU Video Measurement Tool [64] to compute the metrics. As can be seen in Fig. 23(a), (b) and (c) which illustrate the computed PSNR, SSIM and VQM, respectively for the players using our proposed method, the quality scores show the degradation in quality for each frame alone, while the viewer's perception is subject to both recent few-seconds observed frames as well as the worst quality section in a video sequence. Therefore, we need a temporal pooling method as a means to compute a single quality score for the entire video sequence. There are various pooling methods that can be used including histogram, Minkowski summation, exponentially-weighted Minkowski summation, Mean Value across (MVA) a sequence etc. [24]. For simplicity, we use the MVA for computing the quality score of each sequence.



(a) PSNR (b) SSIM (c) VQM

Fig. 23.Objective quality scores

We performed 15 runs for all players using different adaptation methods. Fig. 24(a), (b), and (c) show the average and the standard deviation of the computed quality scores for PSNR, SSIM and VQM respectively. It is worth mentioning that PSNR and SSIM give higher scores to video sequences with higher quality, whereas VQM gives lower scores to video sequences with higher quality so that its score would be zero for lossless video. From the figures, we can notice that almost all the players using our proposed method experienced higher average quality compared to other methods. In addition, in our proposed method, the players' QoE is relatively comparable, whereas the players using other methods face larger variances of QoE.



(a) PSNR **(b) SSIM** **(c) VQM**

Fig. 24.Mean and standard deviation of QoE for all players

Although the previously used objective quality metrics are most widely used, their results for HAS streams do not always correlate perfectly with the perceived visual quality of the human visual system which is non-linear [22]. To carry out the better assessment, we also present the evaluation of different adaptation methods compared to our proposed method using a model considering the most important factors influencing the viewers' perception. According to [50], [52], [59], [60] which propose QoE models for HAS, average quality of selected representations, number and magnitude of switches among different representations and frequency and duration of freezes are considered as the most important factors having impact on QoE. Hence, the linear model provided in equation (36) [60] is used to estimate the Mean Opinion Score (MOS) of the adapted video stream.

$$eMOS = 5.67 \times \mu - 6.72 \times \sigma - 4.95 \times \emptyset + 0.17 \quad (36)$$

Where μ and σ denote the average and standard deviation of quality of segments constituting the adapted video stream respectively, and ∅ indicates the impact of choppy playback considering the number and duration of video playback freezes. However, as explained in [52], in order to capture the quality of each chunk belonging to different representations, different metrics including PSNR, SSIM, chunk-MOS and the index of quality levels can be used. As chunk-MOS is usually not available, and PSNR and SSIM are very sensitive to video content, it is concluded in [52] , that using quality level indices yields the most reliable results in terms of predicting the real MOS. To this end, in our evaluation, the quality level indices were used such that the obtained results are within range [0.0 5.3697] which is very close to the typical range of MOS, [1 5].

Table V presents the computed eMOS using the results obtained from experiment set 2. The results of eMOS show that the Fuzzy outperforms TB, BB and FESTIVE methods in terms of average eMOS. Although the average eMOS of SARA is slightly greater than the Fuzzy, it can be seen that the players using the Fuzzy experienced quite similar qualities as opposed to the SARA players whose satisfaction levels are considerably different.

Table V. The computed eMOS for the rate adaptation methods

	Player 1	Player 2	Player 3
Fuzzy	4.1	4.2	4.1
TB	4	3.8	4
BB	4	3.7	4.1
SARA	4.2	3.9	4.4
FESTIVE	4.1	4	3.8

5.2.4 OFF period analysis

In order to evaluate the performance of our proposed method and the other studied adaptation methods in terms creating OFF periods, extensive experiments were conducted based on the experiment set 2 for different amounts of the

bottleneck link's bandwidth including 36 Mbps, 24 Mbps, 18 Mbps, 12 Mbps, 6 Mbps and 3 Mbps.

The results of these experiments are presented in Table VI in the form of average and the standard deviation of measured OFF periods created by different adaptation methods. According to the results provided for 36 Mbps, it is obvious that the average of OFF periods experienced by all HAS players using different adaptation methods are quite similar indicating that when the fair share of available network bandwidth is greater than the highest video bitrate available in the video representation set, adaptation methods have no means to prevent from buffer overflow other than delaying the download of subsequent video segments. However, regarding the standard deviations, it can be seen that the OFF periods created by Fuzzy method and FESTIVE method were drawn from wider range of values compared to other methods. This implies that Fuzzy and FESTIVE try to randomize chunk scheduling to improve stability and fairness among competing players. In the experiments where the available bottleneck link bandwidth was set to 24 Mbps, as shown in Fig. 18 to Fig. 22, the available bandwidth for each HAS player is less than the maximum video bitrate in the presence of cross traffic, and Fuzzy can use the one-step increase in the video bitrate as a means to not impose any delay. As expected, the results provided in Table VI shows that our proposed Fuzzy method reduced the average of OFF periods by almost 29%, 28%, 34% and 20% compared to TB, BB, FESTIVE and SARA respectively. When the available bandwidth was limited to 18 Mbps, we still see that Fuzzy method caused OFF period, as the fair share of bandwidth for each HAS player in the absence of cross traffic was equal to the maximum available video bitrate. On the other hand, for the scenarios in which the available bandwidth was set to 12 Mbps, 6 Mbps and 3 Mbps, our proposed method were able to completely eliminate the OFF periods.

Table VI. OFF period statistic results

Available Bandwidth (Mbps)	Fuzzy		TB		BB		FESTIVE		SARA	
	Avg(s)	Std(s)	Avg(s)	Std(s)	Avg(s)	Std(s)	Avg(s)	Std(s)	Avg(s)	Std(s)

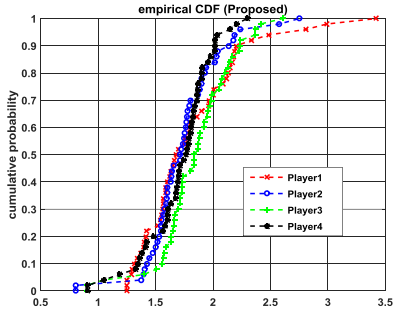
36	2.2381	3.32	2.517	0.767	2.531	0.765	2.421	2.682	2.35	0.977
24	0.775	2.157	1.093	0.582	1.079	0.598	1.177	2.013	0.975	0.687
18	0.202	1.099	1.255	0.416	0.548	0.552	1.015	2.004	0.719	0.682
12	0	0	0.825	0.307	0.958	0.33	0.964	1.74	0.855	0.465
6	0	0	1.007	0.318	1.043	0.316	1.088	1.742	0.917	0.463
3	0	0	1.30	0.412	1.36	0.393	1.381	2.73	1.232	0.555

5.2.5 Empirical Analysis

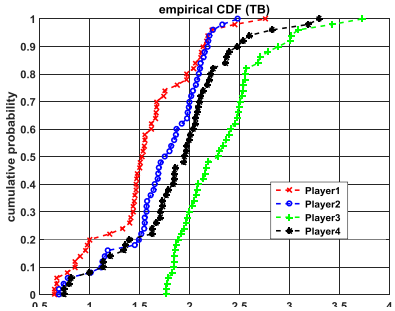
In this section we define a competition scenario in which four homogeneous players using different studied adaptation methods compete for 300 seconds on the bottleneck link whose available bandwidth is fixed at 8 Mbps. In this scenario, we did not consider the cross traffic to statistically investigate the behavior of HAS players themselves in terms of sharing the available bandwidth.

We have conducted 50 runs for each adaptation method and then the average of segment throughput experienced by each HAS player over each run was taken. The first 5 samples were not considered in the averaging process, to only take into account the samples measured when the system was stable. The average segment throughput obtained from each run can be thought of as a realization of a random process in which the bottleneck link is shared among the HAS players. Afterwards, the ensemble of these 50 sample realizations allows us to construct the empirical cumulative distribution function (ECDF) of the segment throughput experienced by each HAS as an estimation of the real CDF. Fig. 25(a), (b), (c), (d) and (e) present the distributions of the players' segment throughputs in the form of CDFs. We also use the two-sample Kolmogorov-Smirnov (2S-KS) test [89], [90] to capture the analysis of the experienced throughput. This test measures the distance between two empirical cumulative distribution functions (ECDFs). Based on 2S-KS, we test the null hypothesis (H_0) that the ECDFs of two players (denoted by P_i and P_j) using the same adaptation method are drawn from the same CDF at the 5% significance level ($H_0 : P_i = P_j, 1 \leq i \leq 4, \ 1 \leq j \leq 4, i \neq j$). The computed pairwise values of H_0 and p-value for the four players, considering the different types of adaptation methods, are reported in Table VII. According to the value of significance level, i.e. 5%, if p-value >0.05 then H_0 is accepted ($H_0 = 1$); otherwise, it is rejected ($H_0 = 0$).

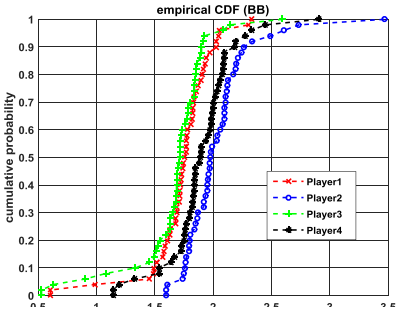
As can be seen from the reported values of the proposed method in Table VII, the test of the null hypothesis for the four players are accepted which means that the ECDFs of the four players are similar. We can conclude that when the players use our proposed method, they fairly share the available bandwidth. The results for TB reveal that the ECDF of player 2 (P2) is similar to the ECDF of player 4 (P4) while the ECDFs belonging to player 1 (P1) and player (P3) are akin to each other. Fig. 25 (b) shows that P4 on average uses more bandwidth than its fair share, whereas P1 uses less. The 2S-KS results for players using the BB method indicate that the ECDFs of P1 and P3 and the ECDFs of P2 and P4 are pairwise alike. Also, the 2S-KS results of BB players show that the ECDFs of P1, P2 and P4 are comparable while different that P3. In conclusion, the players using our proposed method share the available bandwidth in a fair way, whereas in the other methods, some players suffer from being over or under the fair share.



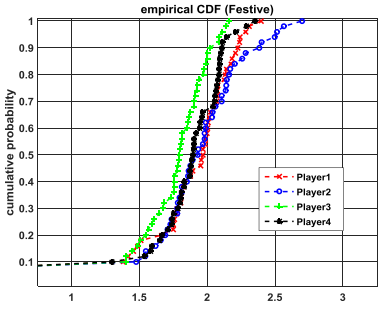
(a) Proposed method



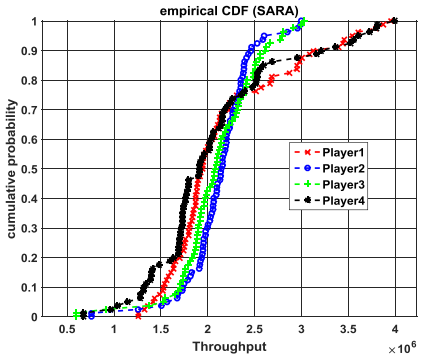
(b) TB method



(c) BB method



(d) FESTIVE method



(e) SARA method

Fig. 25.Empirical CDFs of experienced throughput

Table VII. Two-sample Kolmogorov-Smirnov results

Proposed method

(H_0,P-Value) a = 0.05

	P1	P2	P3	P4
P1	-	(1,0.507)	(1,0.317)	(1,0.2408)
P2	(1,0.507)	-	(1, 0.423)	(1,0.6779)
P3	(1,0.317)	(1,0.423)	-	(1,0.0560)
P4	(1,0.2408)	(1,0.6779)	(1,0.0560)	-

TB

(H_0,P-Value) a = 0.05

	P1	P2	P3	P4
P1	-	(0, 0.017)	(0,5.3e-13)	(0,7.1e-05)
P2	(0, 0.017)	-	(0,3.6e-06)	(1, 0.24)
P3	(0,5.3e-13)	(0,3.6e-06)	-	(0, 0.002)
P4	(0,7.1e-05)	(1, 0.24)	(0, 0.002)	-

BB

(H_0,P-Value) a = 0.05

	P1	P2	P3	P4
P1	-	(0,1.23e-06)	(1, 0.507)	(0, 0.002)
P2	(0,1.23e-06)	-	(0, 2.9e-09)	(1, 0.095)
P3	(1, 0.507)	(0, 2.9e-09)	-	(0,1.7e-04)
P4	(0, 0.002)	(1, 0.095)	(0,1.7e-04)	-

FESTIVE

(H_0,P-Value) a = 0.05

	P1	P2	P3	P4
P1	-	(1, 0.954)	(0, 0.004)	(1, 0.358)
P2	(1, 0.954)	-	(0, 0.031)	(1, 0.154)
P3	(0, 0.004)	(0, 0.031)	-	(0, 0.035)
P4	(1,0. 358)	(1, 0.154)	(0, 0.035)	-

SARA

(H_0,P-Value) a = 0.05

	P1	P2	P3	P4
P1	-	(1, 0.3125)	(0,0.1875)	(0,0.1765)
P2	(1, 0.3125)	-	(0, 0.1750)	(1,0.35)
P3	(0,0.1875)	(0, 0.1750)	-	(1,0.2875)
P4	(0,0.1765)	(1,0.35)	(1,0.2875)	-

Afterwards, we compute the estimated MOS using the model provided in subsection 5.2.3 to evaluate the different adaptation methods in terms of QoE. Fig. 26 shows the average eMOS along with its 95% confidence interval for each adaptation method. It is worth mentioning that the results were obtained for all

players using the same method across 50 runs of defined scenario in subsection 5.2.5.

It can be observed that on average, the players using our proposed method (Fuzzy) experienced higher average video quality compared to players using other methods. However, the average quality of SARA and FESTIVE are slightly lower than that of proposed method. Accordingly, it can be concluded that Fuzzy, SARA and FESTIVE perform rather similarly when taking into account the average eMOS. In addition, the results for TB and BB indicate that they achieved much lower average quality compared to other methods.

The main reason behind that is due to inaccurate estimation of throughput obtained from smoothing measured segment throughputs using the constant smoothing factor while SARA and FESTIVE use weighted harmonic mean to estimate the available bandwidth. It should be noticed that BB does not make use of estimated bandwidth when the buffer level falls in some predefined ranges; so it shows better performance than TB. Moreover, in order to measure the consistency between HAS clients using the same adaptation method, the standard deviations of eMOS, together with their 95% confidence intervals, are provided in Fig. 27. The magnitude of this standard deviation represents the size of difference among the average eMOS experienced by the HAS clients. On average, the Fuzzy method was able to reduce eMOS deviation by 34%, 43%, 6% and 18% with comparison to TB, BB, SARA and FESTIVE methods respectively. FESTIVE presents relatively good performance by randomized chunk scheduling, and mitigates the issue of unfair bandwidth allocation among competing HAS players. However, results of Fuzzy and SARA methods show the lower standard deviation compared to FESTIVE which result from this fact that both of which eliminate OFF periods when possible (for Fuzzy when there is representation with higher bitrate than estimated bandwidth, and for SARA when the playback buffer is not completely full) by continuously downloading the following segment. It can be observed that TB and BB shows the worst performance that indicates there is a considerable difference among the video quality perceived by the HAS clients using these methods. These results also

support this fact, which is neglected by many HAS adaptation methods, that the proper segment download scheduling method is able to improve the video quality perceived by different HAS clients sharing bandwidth on a bottleneck link.

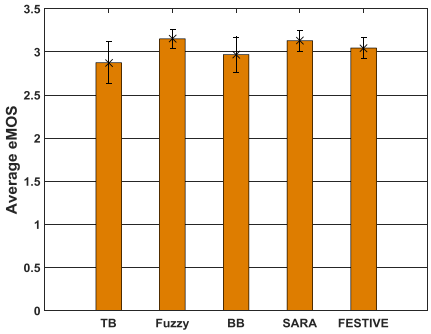


Fig. 26.Average eMOS for the players using TB, Fuzzy (proposed method), BB, SARA and FESTIVE

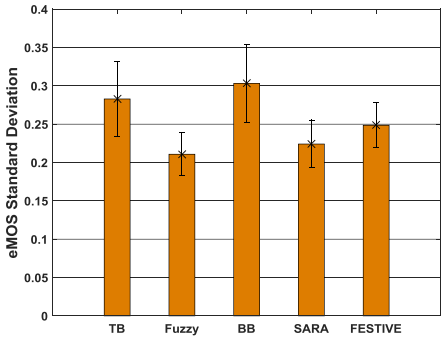


Fig. 27.Standard deviation of eMOS for the players using TB, Fuzzy (proposed method), BB, SARA and FESTIVE

5.2.6 Fairness analysis

Finally, we investigate the fairness of our proposed method compared to TB, BB and FESTIVE. We conduct an experiment with different number of players. We assume that the available bandwidth of the bottleneck link is fixed such that each player is assigned 2 Mbps of the share bandwidth. For instance, if 8 players use the bottleneck link, the available bandwidth is fixed at 16 Mbps.

In each run, we calculate the fairness using Jain index [91]. Fig. 28 plots the average of the unfairness index across the different number of the competing players. The average fairness measured by Jain index experienced by the players using our proposed method is almost 80%, 28%, 39%, %35 more than that of TB, BB, FESTIVE and SARA respectively. However, the unfairness indices of our proposed method and FESTIVE slightly vary for different number of players whereas the unfairness indices of TB and BB considerably change as the number of players increases.

These results reveal that although FESTIVE strives to eliminate the biasing event among the competing players and show a flat behavior across different number of players, the existence of OFF-periods gives the opportunity to take the bandwidth and cause unfair bandwidth utilization. The players in our proposed

method share the bandwidth fairly because they use progressive download when the shared bandwidth is less than the highest available video bitrate.

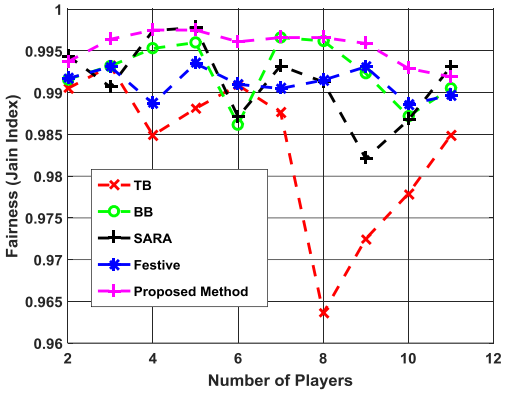


Fig. 28.Fairness for the different number of players

5.3 Summary of Work

In this chapter, we have presented the system simulation and experimental results to show that the proposed user-centric user centric video bitrate adaptation and prediction mechanism for HAS can provide better performance compared to other existing solutions in terms of fair bandwidth allocation and quality of streams perceived by players.

Chapter 6. Network-Assisted Approach for HAS Video Streaming in Mobile Wireless Networks

6.1 Main Idea

From the service providers' perspective, maximizing QoE for an individual HAS player is not only important, but also attaining fair satisfaction of all HAS users is their ultimate goal. However, according to current implementations of HAS players where the adaptation logic resides at the client side, achieving such a goal in providing HAS services is non-trivial due to the shortcomings discussed in subsection 3.2. Moreover, in case of mobile wireless network delivery, there are two major challenges that hinder the efficiency of HAS: First, HAS adaptation methods use the average network throughput measured for previously downloaded video segments as an indication of network condition and the available bandwidth.

In a wireless network, the radio channel conditions can significantly change over a short period of time, and due to device mobility, the measured TCP throughput does not reflect the real network condition. This leads to an inaccurate estimation of the network bandwidth. Secondly, in mobile wireless networks, the available bandwidth for each user is proportional to the amount of radio resource allocated to that user by a scheduler operating at a base station (e.g., eNodeB in Long-Term Evolution (LTE)). Since the scheduler does not have any knowledge regarding the streamed video, it does not take into account the content characteristics in the

process of scheduling. As a result, it potentially jeopardizes the video stream's perceived quality, which is the ultimate goal of all adaptation algorithms in HAS.

To overcome the aforementioned issues, the recent 3GPP DASH specification [92] provides HAS clients with QoE measurement and reporting features. These built-in features enable network operators to enhance the quality of experience in their service provisioning [93]. The new QoE monitoring and reporting framework and the complete knowledge of the mobile network regarding mobile devices' locations and channel qualities open up collaboration opportunities between the video hosting service providers and the network operators to consider the characteristics of multimedia contents in optimizing resource allocation and enhancing users' QoE. To this end, in this section, we propose a novel method for optimizing the QoE of videos delivered with HAS over the Evolved Packet System (EPS) based on the new built-in QoE measurement and reporting features. Our main objective is to maximize the HAS users' QoE. According to [92], factors such as average bitrate and temporal bitrate changes contribute to that main objective.

However, since under some circumstances, they are contradicting so that optimizing based on one factor might have a detrimental impact on others, maximizing the overall QoE while considering all contributing factors would not be easy to achieve. To overcome this issue, we formulate the problem of maximizing the overall QoE as a multi-objective problem that maximizes the average bitrates of all HAS users and minimizes the switching up and down among different representations (i.e. the main reason of QoE degradation [27]). Then, HAS users are being informed of the optimum bitrates by receiving the periodical updated MPD files from the mobile wireless network. To take advantage of well-known continuous optimization techniques and to decrease the computational complexity, we relax the formulated problem by converting the discrete optimization problem into a continuous form. Finally, we propose a gradient-based heuristic method to solve the continuous optimization problem.

6.2 Proposed Optimization Framework

Fig.29, shows the proposed signalling structure based on the policy and charging control (PCC) architecture [24], [94], [95] for the Evolved 3GPP Packet Switched domain. It should be mentioned that, for the first time, the concept of policy level in the PCC architecture was proposed in LTE-Advanced Release 10, known as 5G networks. In this proposed signalling model, it is assumed that each Internet media service provider is given a separate Access Point Name (APN), and each APN is associated with an Application Function (AF). As explained in 2.6.3, the AF is in charge of interacting with the application running at UEs, e.g. SIP agent and HAS application, to extract the session-level information from the corresponding session protocols. In accordance with the HAS service, the session information can be the information embedded in the MPD file like the available representations of the requested video. Moreover, the HAS clients are also informed to use the corresponding AF as a server to report their QoE feedbacks including average video bitrate, temporal quality changes, and re-buffering time. Upon reception of updates from either currently established HAS sessions or QoE updates, the AF is responsible for reporting the new information to the Policy and Charging Rules Function (PCRF) entity through the Attribute-Value-Pairs (AVPs), which are embedded in Authenticate and Authorize Request (AAR) messages over the Rx interface.

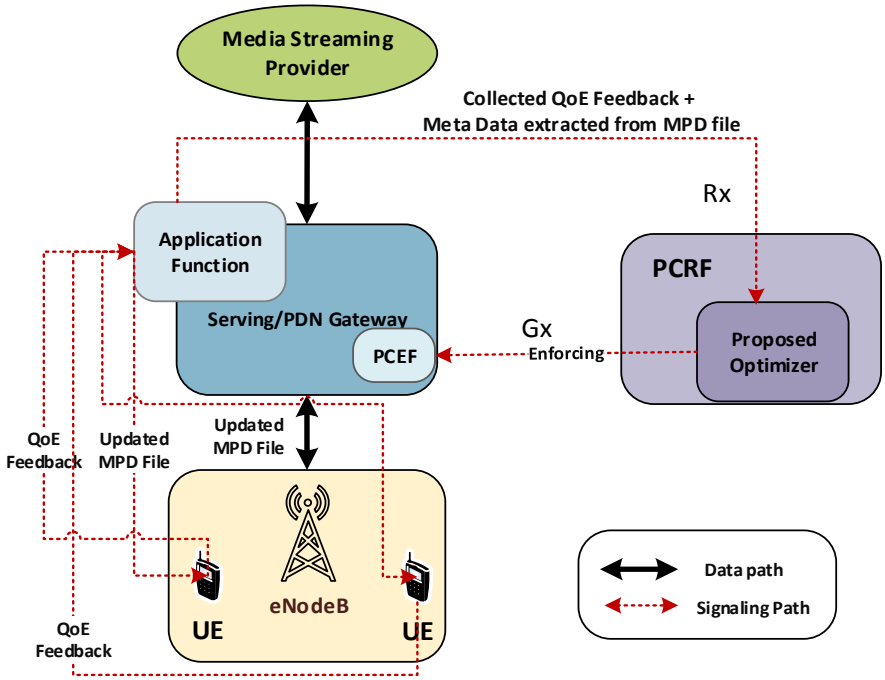


Fig.29. General Proposed Wireless Architecture for QoE-Aware DASH

Based on the above description, the received information is then passed to the proposed optimizer employed at the PCRF to specify and to manage QoS-related parameters for each HAS flow. It should be noted that in our proposed method, it is assumed that resource slicing techniques are applied such that the resource management of HAS flows is separated from other types of data traffic. Hence, the optimizer just determines the resources to be used across the HAS flows. By using the resource slicing technique, any change in one slice does not influence the allocation of resources to other slices. In addition to the information received from the AF, the optimizer may interrogate the Subscription Profile Repository (SPR) regarding subscriber related information, as described in 2.6.3. The collected information is then passed to the proposed optimizer as inputs to allocate the radio resource to the HAS flow to maximize the quality perceived by each HAS client. Then, the decisions regarding the resource allocated to each flow are converted to a set of PCC rules by the PCRF. Afterwards, the PCRF sends new or modified PCC rules to the Policy and Charging Enforcement Function (PCEF) located at the PDN Gateway where these rules are mapped to a particular IP connectivity access

network (IP-CAN) bearer by the Bearer Binding Function (BBF). The PCEF is also responsible for performing policy enforcement and providing user data flow handling as well as QoS handling. Moreover, the decisions made by the PCRF and authorized by the PCEF are transferred back to the AF using the already defined Authenticate and Authorize Answer (AAA) messages such that there is no need to define a new messaging mechanism. According to the authorized QoS parameters, the AF creates MPD updates to be downloaded by HAS players at pre-determined intervals. This cooperation enables HAS players to adapt their requests for the following video segments to the resources allocated by the radio network. As the main objective pursued in this model is maximizing the overall user satisfaction, our goal is to optimize the allocation of the radio resources according to quality indicators reported by HAS players. Consequently, the appropriate QoS parameters can be determined to drive the HAS players to select the best fitted video according to radio and network configurations. The objective functions of the proposed optimization mechanism are discussed next.

6.3 System model

The access network of LTE consists of a network of eNodeBs (eNBs). eNBs can be indexed by the set $E = \{e_1, e_2, \ldots, e_L\}$ where L is the total number of existing eNBs. The set of $UE_i = \{ ue_{i,1}, ue_{i,2}, \ldots, ue_{i,ki} \}$ is defined to represent the total k_i of UEs served by the i^{th} eNB, denoted by e_i. As explained before, since the resource management of HAS flows is separated from other types of data traffic, we consider only UEs that receive HAS flows. In accordance with the type of bearer, as explained in [96], EPS bearers are conceptually categorized into different classes, and each class is recognized by a QoS Class Identifier (QCI). QoS classes are generally divided into two main categories, namely, Guaranteed Bit Rate (GBR) and non-Guaranteed Bit Rate (non-GBR). In GBR, when establishing the bearers, they are associated with bitrate parameters to support allocation of a guaranteed bitrate, while in non-GBR bearers, bitrates are not guaranteed. According to 3GPP technical report TS 26.247 [97], using the GBR bearers for HAS flows produces

higher performance in contrast with non-GBR bearers. Therefore, it is assumed that one dedicated GBR bearer is set up for each HAS flow by the serving gateway and the corresponding eNB. Let $M_{i,j}$ be the total number of available video bitrates (available representations of a video sequence), which can be requested by $ue_{i,j}$ (j^{th} UE interacting with i^{th} eNB). For example, for $M_{1,1} = m$, where m is the available video bitrate levels, $ue_{1,1}$ has m available choices of bitrates, $\{r_{1,1,1}, r_{1,1,2},.., r_{1,1,m}\}$, to select from. After having defined the notations, the problem of resource allocation across multiple HAS UEs served by different eNBs is formulated as a multi-objective discrete-optimization problem in which two main objectives, including maximizing the overall QoE and maximizing the stability are pursued, these objectives are explained in the following section.

6.4 Objective functions

The first objective function, denoted by O_1, is defined to maximize the average perceived quality experienced by all HAS users using the same APN. To this end, a parametric utility model, $U(r)$, is derived for each user to capture the corresponding video characteristics. We adopt a sigmoid function as proposed in [98] to model the objective quality measured by either PSNR or SSIM for different values of video bitrate where $U(r) = l/(1 + e^{-\Delta r})$. Δ represents the rate of change in utility curve, and in other words, how quickly the PSNR and SSIM deteriorates or improves when the bit rate (r) is decreased or increased respectively for a particular video sequence.

Hence, the first objective is achieved by the maximizing the total utility of video bitrates assigned to multiple HAS users across all serving eNBs, and can be formulated as in equation (37).

$$O_1: \sum_{i=1}^{L} \sum_{j=1}^{k_i} \sum_{l=1}^{M_{i,j}} U_{ij}(r_{ijl}) x_{ijl} \quad (37)$$

where x_{ijl} is a binary variable, equal to 1 if the j^{th} UE in the i^{th} eNB is assigned the l^{th} level of video bitrate (r_{ijl}), and equal 0 otherwise. The second objective function, indicated by O_2, is defined to minimize the up and down switching between different representations during playback which may lead to QoE reduction [27]. Accordingly,

it is important to have a scale to quantify the impacts of quality changes. To do so, we employ Just Noticeable Difference (JND) [32] to capture such effects. Assuming that all video bitrates in a representation set are spread out with fixed distance in terms of JND (e.g. 1.5 JND), a function $D(r_i, r_j)$ can be defined as in equation (38) to estimate the perceived quality difference between two different qualities.

$$D(r_i, r_j) = |i - j| * FD \tag{38}$$

where r_i and r_j are the video bitrates of two different representations, and FD is a fixed distance in JND unit. Now we can define the objective function O_2 using $D(r_i, r_j)$ as follows:

$$O_2: \sum_{i=1}^{L} \sum_{j=1}^{k_i} \sum_{l=1}^{M_{i,j}} D(r_{ijl}, r_{ij}{}^*) x_{ijl} \tag{39}$$

where $r_{ij}{}^*$ denotes the video bitrate of the video segment selected in the last round for the j^{th} UE in the i^{th} eNB. Also l is used as an index specifying the l^{th} representation in the representation set available for the video sequence requested by $ue_{i,j}$.

6.5 Bi-objective discrete optimization problem

Having defined the two objectives, O_1 and O_2, allows us to introduce the bi-objective optimization problem P_1 as follows:

$$P_1: max\{O_1\}, min\{O_2\} \equiv min\{-O_1, O_2\} \tag{40}$$

Subject to:

$$\sum_{j=1}^{k_i} \sum_{l=1}^{M_{i,j}} \left\lceil \frac{r_{ijl}}{c_{ij}} \right\rceil x_{ijl} \leq N_{max} \; \forall i \in \{1,2,\dots,L\} \tag{41}$$

$$\sum_{l=1}^{M_{i,j}} x_{ijl} = 1 \; \forall i \in \{1,2,\dots,L\}, \forall j \in \{1,2,\dots,k_i\} \tag{42}$$

$$x_{ijl} \in \{0,1\} \; \forall i \in \{1,2,\dots,L\}, \forall j \in \{1,2,\dots,k_i\}, \forall l \in \{1,2,\dots,M_{i,j}\} \tag{43}$$

$$B_{ij} + (-T + \frac{T \times r_{ijl}}{AR_{ijl}}) x_{ijl} > B_{th} \; \forall i \in \{1,2,\dots,L\}, \forall j \in \{1,2,\dots,k_i\}, \forall l \in \{1,2,\dots,M_{i,j}\} \qquad (44)$$

Equation 40 represents the problem of bi-objective optimization that aims to jointly minimize the up and down switching between different representations during playback and maximize the overall quality among all HAS UEs. As explained in [99], it is assumed that N_{max} resource blocks, i.e. the quantum of allocatable physical radio resource, are assigned to the HAS flows by the resource allocator running at each eNB. Moreover, as the UEs can have various link qualities, depending on the modulation and coding scheme (MCS) that the i^{th} eNB utilizes for the j^{th} UE, the maximum achievable bitrate to transmit to the UE per resource block is denoted by c_{ij}. Hence, the total number of required resource blocks that satisfies the requested bitrate r_{ijl} would be $\left\lceil \frac{r_{ijl}}{c_{ij}} \right\rceil$. Constraint (41) ensures that the number of total resource blocks allocated to the UEs served by the same eNB cannot exceed the maximum capacity of resource blocks designated for HAS flows in the base station. Constraint (42) ensures that exactly a single video bitrate among the different video bitrates is being selected for each HAS UE. It is worth mentioning that as the total number of available representations, $M_{i,j}$, can be different for different users depending on the requested video sequence, x_{ijl} in constraint (43) is used as a decision variable to check whether the l^{th} level of the video bitrate is selected or not by the j^{th} UE. Finally, using the average throughput (denoted by AR_{ijl}) and the current level of playback buffer (B_{ijl}), which are measured and reported by each HAS UE as QoE feedback, enables us to define constraints (43) to keep the buffer level of each user above the predefined threshold (B_{th}). In constraints (44), T denotes the duration of each segment in seconds.

6.6 Bi-objective continuous optimization problem

Problem (40) can be considered as a 0-1 Knapsack problem with a slight difference in coefficients. Dynamic Programming (DP) is a commonly used technique to solve

such a discrete problem. However, the time complexity of DP increases in pseudo-polynomial fashion to the input size. To have better intuition regarding the scale of the input size of our problem, let us consider an LTE access network with 10 MHz bandwidth.

The total available resource blocks (RBs) per frame at 10 ms are around 500. If the proposed scheduling is performed every second, the number of RBs for downlink would be applied around 20000. Hence, the large value of input size makes the DP method impractical to apply for problem (40). To solve P_1 with reasonable and manageable complexity, we utilize the most widely used relaxation approach in which the discrete decision variables are replaced by continuous variables. These replacements relax our problem and remove the restriction of being discrete, and allow us to benefit from well-known continuous optimization techniques. Once the optimal solution of the relaxed continuous optimization problem is obtained, the optimum values can be quantized to the nearest feasible discrete solution. However, it is worth noting that the quantized optimal solution is not necessarily the optimal solution of the discrete optimization problem. Assuming that the rate allocated to each HAS UE, $\widetilde{r_{ij}}$, is a continuous variable ranging from the lowest video bitrate to the highest video bitrate available in the set of representations of the video sequence requested by that UE, the continuous form of P_1 can be expressed as (45).

$$P_2: min\{-\sum_{i=1}^{L}\sum_{j=1}^{k_i} U_{ij}(\widetilde{r_{ijl}}), \sum_{i=1}^{L}\sum_{j=1}^{k_i} D(\widetilde{r_{ij}}, {r_{ij}}^*)\} \quad (45)$$

Subject to:

$$\sum_{j=1}^{k_i} \frac{\widetilde{r_{ij}}}{c_{ij}} \leq N_{max} \; \forall i \in \{1,2,\dots,L\} \quad (46)$$

$$r_{ij1} \leq \widetilde{r_{ij}} \leq r_{ijM_{i,j}} \quad \forall i \in \{1,2,\dots,L\}, \forall j \in \{1,2,\dots,k_i\} \quad (47)$$

$$B_{ij} + \left(-T + \frac{T \times \widetilde{r_{ij}}}{AR_{ij}}\right) > B_{th} \; \forall i \in \{1,2,\dots,L\}, \forall j \in \{1,2,\dots,k_i\} \quad (48)$$

As can be seen in (45), the binary decision variable x_{ijl} , which selects a discrete value from the available video bitrates in the representation set $\{r_{ij1}, r_{ij2}, \dots, r_{ijM_{i,j}}\}$ for the HAS UE, is ignored, and r_{ijl} is substituted with a new continuous decision variable $\widetilde{r_{ijl}}$. Also, constraint (47) is defined to guarantee that the allocated bitrate is in the range of r_{ij1} and $r_{ijM_{i,j}}$, i.e. the lowest and the highest video bitrates available for the video requested by the j_{th} UE in the i_{th} eNB respectively. Similar to constraint (44), constraint (48) ensures that according to the current knowledge of average throughput experienced by each user (AR_{ij}), the allocated bitrate to each UE ($\widetilde{r_{ij}}$) does not result in a buffer level (B_{ij}) less than the predefined threshold (B_{th}).

After having formulated a multi-objective continuous optimization problem P_2, there is another important aspect of P_2 that needs to be dealt with. The optimization P_2 contains two conflicting objectives, maximizing both the utility and the stability simultaneously. In other words, P_2 tries to increase the overall bitrates allocated to all UEs, which in turn requires switching of the video bitrate. This switching increases the instability, which conflicts with the second objective of trying to maximize the stability. Hence, the solution to the problem depends on the trade-off level between the opposing objectives, and as different levels of trade-off can be defined, there can be more than one unique solution. A set of solutions for the multi-objectives problem is referred to as Pareto Optimal solutions or Pareto frontier.

To demonstrate how Pareto solutions of P_2 can be obtained, we employ the Normal Boundary Intersection (NBI) technique among the available approaches used for capturing the Pareto frontier. NBI generates a set of Pareto points evenly spread out in the design space [100]. Moreover, since NBI produces solution points uniformly distributed on the boundary, it outperforms the weighted sum in terms of finding the best solution point in the Pareto frontier. Before discussing the NBI algorithm, the following notations are introduced.

$\widetilde{r_{ij}}^{*}(z)(z \in \{1,2\}, i \in \{1,2,\dots,L\}, j \in \{1,2,\dots,k_i\})$ represents the optimal allocated bitrates to all users obtained for O_1 and O_2 separately. Variable z is used to indicate the number of objectives.

O_z^* ($z \in \{1,2\}$, $O_j^* \in \mathcal{R}$) denote the corresponding optimal values of objective functions ($O_1^* = O_1\left(\tilde{\mathrm{r}}_{\mathrm{ij}}^{\,*}(1)\right)$, $O_2^* = O_2\left(\tilde{\mathrm{r}}_{\mathrm{ij}}^{\,*}(2)\right)$).

Utopia Point (O^u) is defined as a vector of $O_z^*(z \in \{1,2\})$ on the objective space ($\mathcal{R}^2$), ($[O_1^*, O_2^*]$).

z^{th} anchor point (O^{z^*}) is a vector of values of all objective functions obtained for $\tilde{\mathrm{r}}_{\mathrm{ij}}^{\,*}(z)$.

NBI generally includes four steps:

Step -1: Individually perform optimization problems O_1 and O_2 that yield anchor points (O^{z^*}, $z \in \{1,2\}$).

Step -2: Generate 2×2 pay-off matrix Φ as follows:

$$\Phi = \begin{bmatrix} 0 & O_1\left(\tilde{\mathrm{r}}_{\mathrm{ij}}^{\,*}(2)\right) - O_1^* \\ O_2\left(\tilde{\mathrm{r}}_{\mathrm{ij}}^{\,*}(z)(1)\right) - O_2^* & 0 \end{bmatrix}, \forall \mathrm{i} \in \{1,2,\dots,\mathrm{L}\}, \forall \mathrm{j} \in \{1,2,\dots,k_i\}$$

Step -3: Generate the weights, ß=[β_1 β_2] so that $\beta_1 > 0, \beta_2 > 0$ and $\beta_1 + \beta_2 = 1$ are satisfied. Hence, Φß samples the line segment (so-called Utopia line) connecting 0^{1^*} and 0^{2^*}.

Step -4: Having selected ß, perform the following sub-problem (SP), defined in equation (45), to find the intersection point between the normal originating from Utopia line (Φß) toward the origin and the Pareto frontier that has the maximum distance (τ):

$$SP: \max \tau$$
$$\widetilde{\mathrm{r}_{\mathrm{ij}}}\ \forall \mathrm{i} \in \{1,2,\dots,\mathrm{L}\}, \forall \mathrm{j} \in \{1,2,\dots,k_i\}, \tau \tag{49}$$

Subject to:

$$\sum_{\mathrm{j}=1}^{k_i} \frac{\widetilde{\mathrm{r}_{\mathrm{ij}}}}{c_{ij}} \leq \mathrm{N}_{max}\ \forall \mathrm{i} \in \{1,2,\dots,\mathrm{L}\} \tag{50}$$

$$\mathrm{r}_{\mathrm{ij}1} \leq \widetilde{\mathrm{r}_{\mathrm{ij}}} \leq \mathrm{r}_{\mathrm{ij}M_{i,j}} \qquad \forall \mathrm{i} \in \{1,2,\dots,\mathrm{L}\}, \forall \mathrm{j} \in \{1,2,\dots,k_i\} \tag{51}$$

$$\mathrm{B_{ij}} + (-\mathrm{T} + \frac{\mathrm{T}\times\widetilde{\mathrm{r_{ij}}}}{\mathrm{AR_{ij}}}) > B_{th}\ \forall \mathrm{i} \in \{1,2,\dots,\mathrm{L}\}, \forall \mathrm{j} \in \{1,2,\dots,k_i\} \quad (52)$$

$$\Phi\beta + \tau.n = \left[O_1(\widetilde{\mathrm{r_{ij}}}) \quad O_2(\widetilde{\mathrm{r_{ij}}})\right]^T \forall \mathrm{i} \in \{1,2,\dots,\mathrm{L}\}, \forall \mathrm{j} \in \{1,2,\dots,k_i\} \quad (53)$$

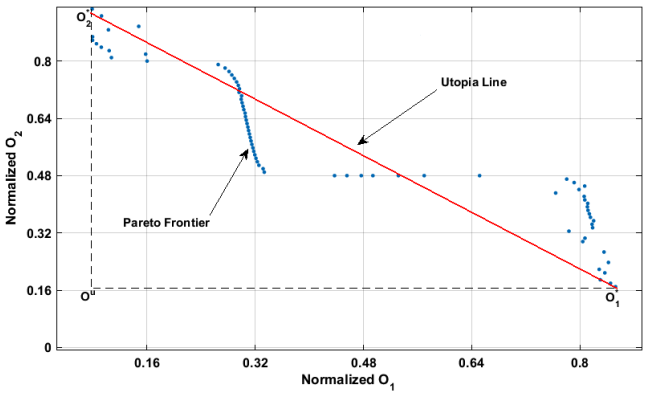


Fig. 30 Pareto frontier obtained using NBI

The constraints (50 51 and 52) ensure the feasibility of the selected bitrates ($\widetilde{\mathrm{r_{ij}}}, \forall \mathrm{i} \in \{1,2,\dots,\mathrm{L}\}, \forall \mathrm{j} \in \{1,2,\dots,\mathrm{k_i}\}$) for all HAS UEs with respect to the constraint set defined in P_2. Constraints in (53) ensure that $\left[\mathrm{O_1}(\widetilde{\mathrm{r_{ij}}}) \quad \mathrm{O_2}(\widetilde{\mathrm{r_{ij}}})\right]^{\mathrm{T}}$ are exactly located on the Pareto frontier. As an example, we consider a population of 12 users whose $\mathrm{B_{ij}}$ and $\mathrm{AR_{ij}}$ follow the uniform distribution. The obtained Pareto frontier is illustrated in Fig. 30. As depicted in Fig. 30 a total of 100 evenly spaced points on the Utopia line are used. Each of these points are associated with unique values of $\beta_1$and β_2 so that $\beta_1 + \beta_2 = 1$.

6.7 Single objective continuous optimization problem

As shown in the previous section, finding points of Pareto frontier (a set of optimal solutions) requires performing several recursions of a problem (49) for different combinations of $\beta_1$and β_2. Hence, capturing Pareto solutions is usually inefficient in terms of computational complexity, and considered as an NP-hard problem. Generally, finding the whole Pareto frontier is not important, and finding one

preferred point on the Pareto frontier would be enough. To address the challenge of finding a single point on the Pareto frontier in a multi-objective optimization problem, the multiple objectives can be combined into one single objective. This method is generally referred to as scalarization or the weighted-sum. In this method, first, each objective is multiplied by a weighting coefficient presenting the preference or importance of the corresponding objective determined by the decision maker. Afterwards, the single objective is formed by summing up the weighted objectives. In this way, we have a new single objective optimization problem whose solution is a Pareto point. The linear scalarized version of problem P_2 can be expressed as follows:

$$P_3: \min(-\omega_1 \sum_{i=1}^{L} \sum_{\mathrm{j}=1}^{k_i} \mathrm{U}_{ij}(\widetilde{\mathrm{r_{ij}}}) + \omega_2 \sum_{i=1}^{L} \sum_{\mathrm{j}=1}^{k_i} \mathrm{D}(\widetilde{\mathrm{r_{ij}}}, \mathrm{r_{ij}}^*)) =$$

$$\min(\sum_{i=1}^{L} \sum_{\mathrm{j}=1}^{k_i} (-\omega_1 \mathrm{U}_{ij}(\widetilde{\mathrm{r_{ij}}}) + \omega_2 \mathrm{D}(\widetilde{\mathrm{r_{ij}}}, \mathrm{r_{ij}}^*))) \quad (54)$$

Subject to:

$$\sum_{\mathrm{j}=1}^{k_i} \frac{\widetilde{\mathrm{r_{ij}}}}{c_{ij}} \leq \mathrm{N}_{max} \; \forall \mathrm{i} \in \{1,2,\dots,\mathrm{L}\} \quad (55)$$

$$\mathrm{r_{ij1}} \leq \widetilde{\mathrm{r_{ij}}} \leq \mathrm{r_{ijM_{i,j}}} \qquad \forall \mathrm{i} \in \{1,2,\dots,\mathrm{L}\}, \forall \mathrm{j} \in \{1,2,\dots,k_i\} \quad (56)$$

$$\mathrm{B_{ij}} + (-\mathrm{T} + \frac{\mathrm{T} \times \widetilde{\mathrm{r_{ij}}}}{\mathrm{AR_{ij}}}) > B_{th} \; \forall \mathrm{i} \in \{1,2,\dots,\mathrm{L}\}, \forall \mathrm{j} \in \{1,2,\dots,k_i\} \quad (57)$$

$$\sum_{n=1}^{2} \omega_n = 1 \; , \; \omega_n > 0 \; \forall \mathrm{n} \in \{1,2\} \quad (58)$$

Where ω_1 and ω_2 are weighting coefficients, and correspond to the relative importance of the objective O_1 and O_2 respectively. Although the weighted sum approach can be utilized to define the single objective problem as a convex combination of different objectives, determining the optimal weights is not a trivial task. Even if a priori satisfactory solution is known by the decision maker, setting the most appropriate weight coefficients is not straightforward since small changes in weights may result in dramatic changes in the optimal solution. To cope with this problem, we propose to use a QoE model that determines the significance of

different involving factors on the quality perceived by the HAS users. According to [50], [60], [101], which propose QoE models for HAS, the average quality of selected representations, number and magnitude of switches among different representations, frequency and duration of freezes are all considered as the most important factors influencing the QoE. The linear model provided in equation (59) [60] is used to estimate the Mean Opinion Score (MOS) of the adapted video playback.

$$eMOS = 5.67 \times \mu - 6.72 \times \sigma - 4.95 \times \emptyset + 0.17 \tag{59}$$

Where μ and σ denote the average and standard deviation of quality of segments constituting the adapted video stream respectively, and ∅ indicates the impact of choppy playback considering the number and duration of video playback freezes. With regard to the analogy between the objectives considered in problem P_3 (54) and the QoE model (59), we can utilize the coefficients of corresponding parameters in the QoE model. However, before applying them in problem P_3, these coefficients should be normalized to meet the constraint (58). Accordingly, 0.45 and 0.55 are obtained for ω_1 and ω_1 respectively.

6.8 Lagrange Dual Problem (Dual Problem)

The optimization problem P_3 in equation (54) is constrained nonlinear programming (NLP). Since Lagrange methods provide an effective approach to solve constrained NLPs, firstly, we define the Lagrangian associated with equation (54) by dualizing the constraint sets (55) and (57) with the Lagrange multiplier sets, λ and ν respectively. The Lagrange dual function can be expressed as below:

$$L(\widetilde{r_{ij}}, \lambda, \nu) = \left(-0.45 \sum_{i=1}^{L} \sum_{j=1}^{k_i} U_{ij}(\widetilde{r_{ij}}) + 0.55 \sum_{i=1}^{L} \sum_{j=1}^{k_i} D(\widetilde{r_{ij}}, {r_{ij}}^*)\right) + \sum_{i=1}^{L} \lambda_i \left(\sum_{j=1}^{k_i} \frac{\widetilde{r_{ij}}}{c_{ij}} - N_{max}\right) +$$
$$\sum_{i=1}^{L} \sum_{j=1}^{k_i} \nu_{ij} \left(-\left(B_{ij} + \left(-T + \frac{T \times \widetilde{r_{ij}}}{AR_{ij}}\right)\right) + B_{th}\right) \tag{60}$$

Then, the Lagrange dual function can be introduced as the infimum of Lagrangian over possible values of $\widetilde{r_{ij}}$ ($r_{ij1} \leq \widetilde{r_{ij}} \leq r_{ijM_{i,j}} \ \forall i \in \{1,2,\dots,L\}, \forall j \in \{1,2,\dots,k_i\}$).

$$L(\lambda,\nu) = inf_{\widetilde{r_{ij}}} L\left(\widetilde{r_{ij}},\lambda,\nu\right) = inf_{\widetilde{r_{ij}}}\left(f(x) + \sum_{i=1}^{L}\lambda_i\, g(x) + \sum_{i=1}^{L}\sum_{j=1}^{k_i}\nu_{ij}h(x)\right) \quad (61)$$

Where

$$f\left(\widetilde{r_{ij}}\right) = -0.45\sum_{i=1}^{L}\sum_{j=1}^{k_i} U_{ij}\left(\widetilde{r_{ij}}\right) + 0.55\sum_{i=1}^{L}\sum_{j=1}^{k_i} D(\widetilde{r_{ij}}, {r_{ij}}^*)$$

$$g\left(\widetilde{r_{ij}}\right) = \sum_{j=1}^{k_i}\frac{\widetilde{r_{ij}}}{c_{ij}} - N_{max}$$

$$h\left(\widetilde{r_{ij}}\right) = -\left(B_{ij} + \left(-T + \frac{T\times\widetilde{r_{ij}}}{AR_{ij}}\right)\right) + B_{th}$$

It should be noted that for any combinations of λ and ν with $\lambda \geq 0$ and $\nu \geq 0$, the dual function produces a lower bound on the optimal value of P_3 (54), referred to as the primal problem [102]. Hence, to find the best lower bound, we define the Lagrange dual problem as in (62).

$$P4: max\ L(\lambda,\nu) \quad (62)$$

Subject to:

$$\lambda_i \geq 0, \nu_{ij} \geq 0, \forall\, i \in \{1,\dots,L\}, \forall\, j \in \{1,\dots,k_i\}$$

Since the objective function of problem P_3 (54) is concave with linear constraints, and the Karush–Kuhn–Tucker (KKT) conditions are satisfied, the strong duality condition is held. Thus, the duality gap is zero and the optimal of primal and dual problems are same. Then, we can solve the dual problem to find the optimal value (i.e., the value equal to the optimal value of primal problem P_3). To do so, we adopt the gradient algorithm to iteratively find the Lagrangian multipliers as detailed in Algorithm 1.

Algorithm 1. Gradient Optimization Algorithm

1: **Inputs:**
λ_0: initial value for Lagrangian multipliers
ν_0: initial value for Lagrangian multipliers
K: number of Iterations
K_{max}: Maximum number of Iterations
α: step size
$\acute{\varepsilon}$: error index

2: **Initialize:**
$\lambda_i^k = \lambda_0$, $\nu_{ij}^k = \nu_0, k = 1, K_{max} = 100$

3: While ($\acute{\varepsilon} \geq .001$) || $K \leq K_{max}$
$\xi_k \leftarrow$ Calculate the gradient at current solution

4: Update Lagrangian multipliers

$$\lambda_i^{k+1} = \lambda_i^k + \alpha^k \frac{\xi^k}{|\xi^k|} , \forall\, i \in \{1, \dots, L\}$$

$$\nu_{ij}^{k+1} = \nu_{ij}^k + \alpha^k \frac{\xi^k}{|\xi^k|}, \forall\, i \in \{1, \dots, L\}, \forall\, j \in \{1, \dots, k_i\}$$

$$\acute{\varepsilon} = |l(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\nu}^{k+1}) - l(\boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)|$$

5: $\alpha^{k+1} = {}^1/_k$
$k = K + 1$

6: End while;

6.9 Summary of Work

In this chapter, we presented a QoE-aware optimization mechanism to allocate radio resource and bandwidth to HAS users in mobile wireless networks. The main objectives pursued by this framework are: Maximizing the overall average quality and minimizing the negative impact of temporal video quality changes for all HAS users. To achieve these contradicting goals, we formulate the problem as a discrete multi-objective optimization problem. Then, the discrete optimization problem is converted into continuous single objective form to be applicable in practical scenarios. Finally, since the strong duality condition is held, and the duality gap is zero, we propose a gradient-based algorithm to solve the Lagrange dual problem instead of the primal problem. In our proposed solution, the HAS user sends the amount of buffered content in the playback buffer and the quality of previously downloaded video segments to the optimization mechanism using the new features introduced by 3GPP DASH specification. The allocated bandwidth to each UE is

enforced at the serving/PDN gateway, and is periodically informed to the UE using the MPD updating mechanism. As a result, the central bandwidth allocation and coordination across UEs and the wireless mobile network leads to considerable improvement in video quality perceived by DASH users and diminishes the challenges of user mobility and uncertainty of dynamic radio environment.

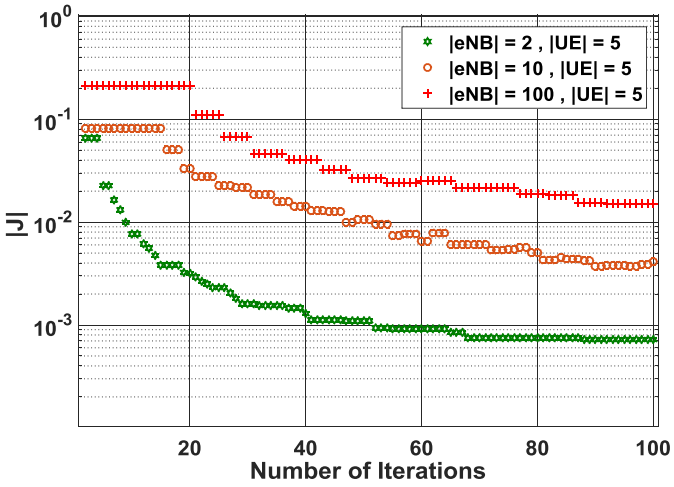
Chapter 7. Performance Analysis of Network-Assisted Approach for HAS Video Streaming in Mobile Wireless Networks

In this section, first we investigate the convergence of Algorithm 1 (Section 6.8). Then, we describe the simulation setup used in our experiments, and finally we evaluate the performance of the proposed optimization method on the QoE perceived by HAS UEs.

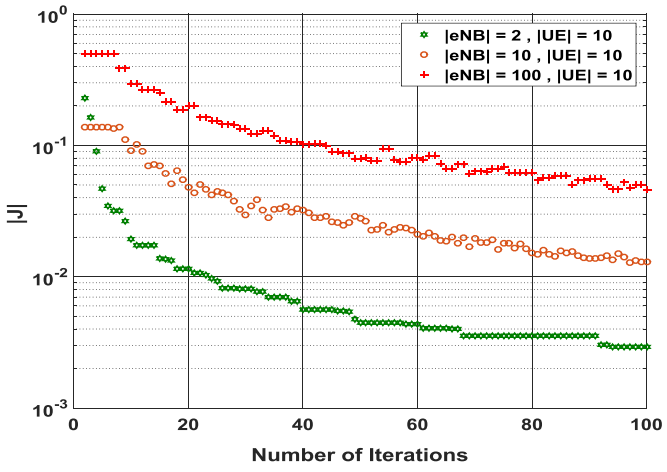
7.1 Convergence Analysis of Algorithm 1

To investigate the performance of the proposed Algorithm 1 in terms of converging to the optimal solution, first we solve the primal problem P_3 using the convex optimization package (CVX) [103] for different numbers of eNBs and UEs denoted by |eNB| and |UE| respectively. Two scenarios are defined based on the number of active UEs (|UE|) at each cell, which represents different scales of the problem. |UE| is considered to be 5 and 10 for the first and the second scenarios correspondingly. For each scenario, the number of eNBs (|eNB|) is considered to be 2, 10, and 100. We ran Algorithm 1 for each configuration separately and computed the absolute value of error (denoted by |J|) between the solution obtained after each iteration and the corresponding optimal solution obtained from CVX for that configuration. Fig. 31 (a) and (b) show the semi logarithmic convergence of the error between the solution of the dual problem and the optimal solution for the different

number of eNBs in both scenarios. As can be seen in Fig. 31 (a), if we consider the desirable error as 10^{-2}, the feasible solution is attained for the dual problem after 10, 50, 100 iterations for |eNB| = 2, |eNB| = 10 and |eNB| = 100 respectively. However, in the second scenario where |UE|=10, more iterations are required to obtain the desirable solution compared to the first scenario where |UE|=5. This convergence analysis shows that the accuracy of the solutions is completely dependent on the number of active UEs in each cell as well as the number of cells itself.



(a) |UE| = 5



(b) |UE| = 10

Fig. 31. The value of |J| versus the number of iterations for different number of eNBs

7.2 Simulation Setup

We use the LTE mobile network model in NS3 [104], which consists of two main models, LTE and EPC (Evolved Packet Core). The former simulates the LTE radio protocol stack (RRC, PDCP, RLC, MAC, PHY), and the latter emulates the core network functions. The Serving Gateway (SGW) and Packet data network Gateway (PGW) entities in the EPC model are implemented in a single entity (SGW/PGW node), and the proposed optimization method is implemented as a member function of the SGW/PGW node. We adopt the model explained in [105] for Adaptive Modulation and Coding (AMC) link adaptation, operating at eNB, to determine the best Modulation and Coding Scheme (MCS) based on the observed Channel Quality Indicator (CQI) for each UE. According to the specified MCSs, the maximum achievable transmission rates (c_{ij}) of UEs per resource block are computed and used in the proposed method.

The network topology in Fig. 32 was applied to evaluate the proposed method. Two cells were set up and all HAS within a cell were distributed at a random distance to eNB from 50m to 800m. Also, for each UE, a default EPS bearer is automatically activated when the UE is attached to the network to obtain the IP connectivity for that UE, and a dedicated downlink EPS bearer is set up manually using the Traffic Flow Template (TFT) to be used for sending the TCP packets belonging to the HAS flow. The system parameters applied in the simulation are summarized in Table VIII. Moreover, to simulate the behaviour of the HAS client and server; we extended the modules PacketSink and OnOffApplication in NS3.

In each HAS server, to represent the different available video bitrates, multiple instances of the OnOffApplication class are considered so that each instance of OnOffApplication is configured to generate a traffic flow with the bitrate equal to one video bitrate available in the representation set. Also, at any time, one instance is active and other instances are stopped. The HAS client receives the packets from the active OnOffApplication, and measures the incoming average throughput within a period of time equal to the segment duration. Then, the estimated throughput is fed into a throughput-based adaptation logic, discussed in [42], implemented as a

function to select the most fitted video bitrate for downloading the next segment. Once the next video bitrate is determined, the corresponding OnOffApplication in the HAS server is activated to generate the traffic for the duration of ON period; i.e., segment period. A set of available rates is considered as {100, 150, 200, 250, 300, 400, 500, 700, 900, 1200, 1500, 2000, 3000, 4000, 5000, 6000} Kbps in accordance with RedBull dataset [85].

Table VIII. System parameters

Parameters	value
System bandwidth	10MHz (50 RBs)
AMC model	Piro
Modulation and Coding Scheme (MCS) range	1-28
Fading loss model	Pedestrian (3 km/h)
Cell capacity	33.53 Mbits/s (for MCS 28)
Frame Structure	FDD
Scheduler	Proportional Fair
RBs per RB group (RBG)	1
Tx power of eNB	30 dBm
Tx power of UE	23 dBm
Noise figure	5 dB
Internet delay	20 ms
PDCP SDU	1024
Buffer size at eNB	100 packets
RLC mode	Unacknowledged mode
T	6 s
Buffer_Threshold	12 s
Simulation time	300 s

To evaluate the effectiveness of our proposed method, we consider two configurations in the conducted simulation. In the first configuration (configuration 1), a throughput based adaptation logic (referred to as TB) is enabled for all HAS UEs such that the most suitable video representation is chosen based on smoothed segment throughput. Hence, the HAS UEs can adapt to the varying downlink throughput without interacting with the core network, while in the second configuration (configuration 2), the local adaptation logic is disabled, and the HAS

UEs select the maximum available video bitrate in the representation set provided by the updated MPD file obtained from the optimizer in the SGW/PGW node.

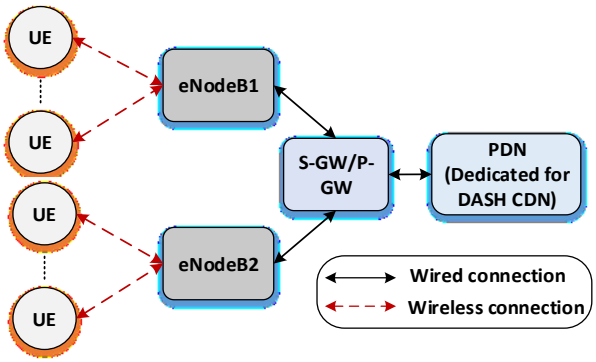
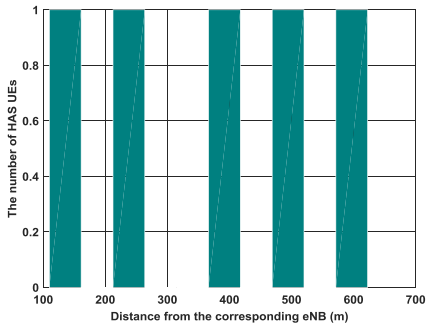
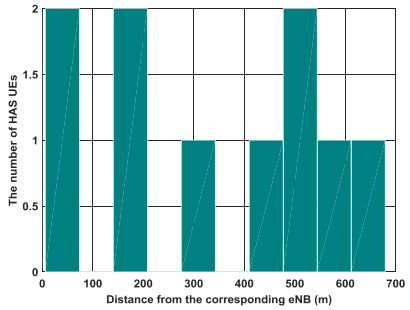


Fig. 32.Network topology used in the simulations

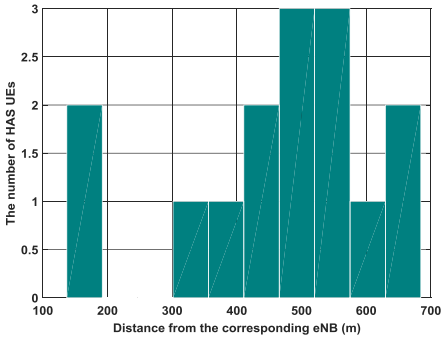
7.3 Simulation Results

To evaluate the performance of the proposed method, we conducted 30 simulation runs of 300 second periods for configurations 1 and 2 separately. This experiment was repeated for different numbers of HAS UEs active at each cell including 5, 10, 15 and 20. Fig. 33 (a), (b), (c) and (d) show the spatial distribution of UEs within a cell. It should be mentioned that the UEs' distribution considered for both cells are similar. In each experiment, we measured the average and standard deviation of quality of segments constituting the adapted video stream for each HAS UE, which allows us to estimate the perceived qualities by applying the eMOS model presented in (59).

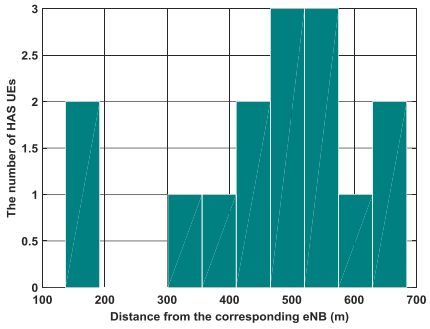




(a) |UE| = 5



(b) |UE| = 10

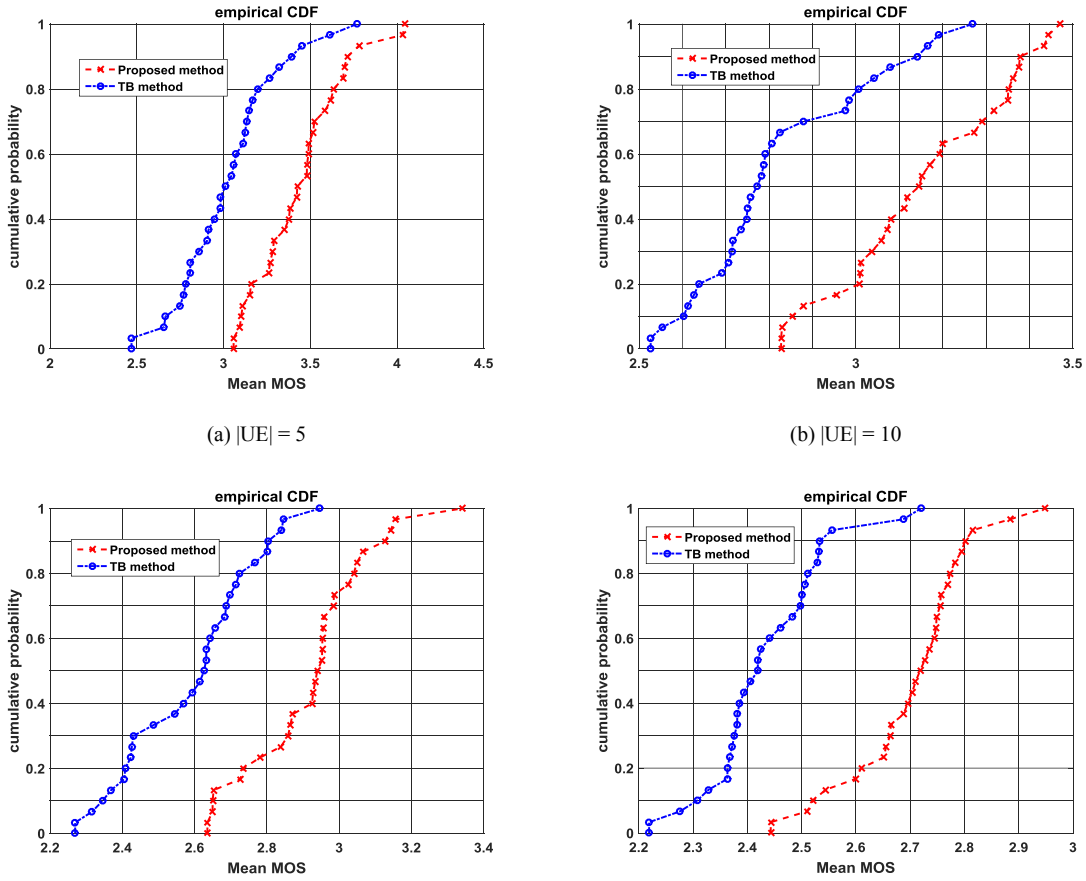


(c) |UE| = 15

(d) |UE| = 20

Fig. 33.The spatial distribution of HAS UEs within a cell

To evaluate the overall performance of our proposed method compared to the TB method in terms of the user's perceived quality, we computed the average of eMOS experienced by all HAS UEs after each experiment. Afterwards, the ensemble of these 30 sample realizations allows us to construct the empirical cumulative distribution function (ECDF) of the mean MOS experienced by all HAS UEs as an estimation of the real CDF. Fig. 34 (a), (b), (c) and (d) present the distribution of the mean MOS experienced by 5, 10, 15 and 20 HAS UEs within each cell. It is noticeable in figure 34 that our method yields higher average quality compared to the TB method for all different number of active HAS UEs. On average, our proposed method produces a gain of 0.3 eMOS across the different number of UEs when compared to the TB method. This improvement stems from the fact that the TB method just aims at increasing the bandwidth utilization whereas increasing the bandwidth does not necessarily result in improving QoE. More importantly, our proposed approach minimizes the fluctuation in QoE compared to the TB method, while the TB method solely follows the fluctuating bandwidth to select the appropriate video bitrate. According to Fig. 34, it can be confirmed that as the number of UEs increases due to the limited available radio resources, the average MOSs for both proposed method and TB method decrease.



(a) |UE| = 5

(b) |UE| = 10

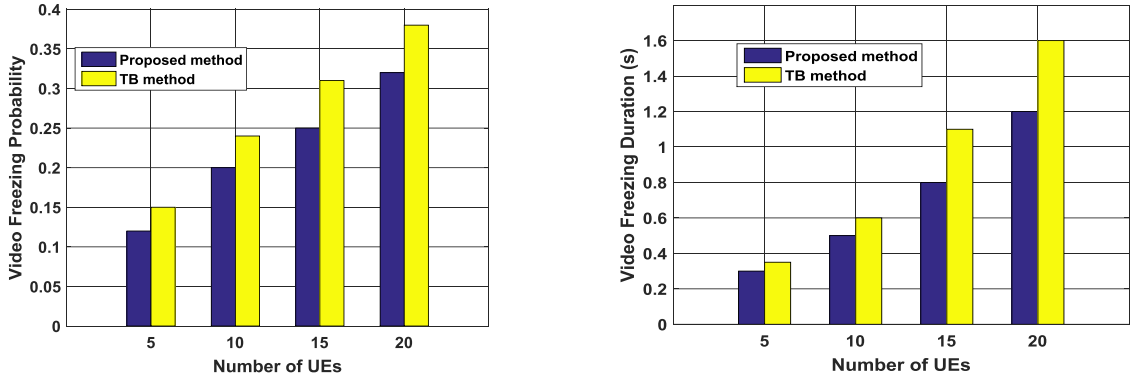
-(c) |UE| = 15

(d) |UE| = 20

Fig. 34.Empirical CDFs of mean MOS obtained from 30 simulation runs

In addition to the average MOS, we also demonstrate the average video freezing probability and average video freezing duration experienced by the HAS UE in Fig. 35 (a) and (b) respectively for different numbers of UEs. As shown in Fig. 35 (a), our proposed method outperforms the TB method by achieving almost 17% lower probability of video freezing. This improvement is the result of considering the playback buffer status after bandwidth allocation, and checking it against the predefined threshold. However, it can be confirmed that as the number of UEs increases, the chance of video interruption rises in both methods and the amount of improvement decreases. This stems from the fact that as the number of clients increases, the HAS UEs use all radio resources such that some UEs face starvation and consequently the performance of the scheduler decreases significantly. The

impact of video freezes on QoE depends not only on the number of occurrences, but also, video freeze duration may lead to perceptual degradation. So, we also measure the video freezing duration for different number of UEs.



(a) Probability (b) Duration

Fig. 35.Video freeze statistics for different number of UEs

As illustrated in Fig. 35 (b), once again our proposed method outperforms the TB method since the UEs using our proposed method experience less freezing duration by almost 20% in comparison with UEs using TB method. As expected, similar to the results of interruption probability, we see growth in the duration of freezes as the number of UEs increases.

7.4 Summary of Work

In this chapter, we presented our testbed to evaluate the proposed QoE-aware optimization framework to allocate radio resource and bandwidth to HAS users in mobile wireless networks. Our simulation results demonstrate that our system results in better perceived quality of video, measured by Mean Opinion Score (MOS), by almost 7% on average, while lowering the freezing period by almost 20% on average across HAS users when compared to other approaches where HAS users only rely on local adaptation logics.

Chapter 8. Conclusion and Future Work

In this chapter, we conclude the presented works in this thesis and suggest topics for future work.

8.1 Conclusion

In recent years, the HTTP adaptive streaming approach has been accepted as a main paradigm for streaming media content due to its scalability, simplicity, bandwidth waste reduction and its ability to pass through NATs and firewalls. However, as discussed in section 1.1, because the adaptation is performed at the client that has incomplete information about the underlying delivery network, which is typically heterogeneous and unmanaged, improving the viewer Quality of Experience (QoE) is very challenging. To this end, the overall focus of this thesis was on optimizing the HAS service to improve QoE by following the methodology described in section 1.4. The extensive study of the current state of the art documented in Chapter 3. Related Works chapter helped us identify the properties of HAS, obtain insights about the existing shortcomings in the current streaming solutions, and understand the factors influencing the perceived quality such as average video bitrate, video freezing event, and quality transitions. Throughout the thesis, we have been suggesting two user-centric and network-assisted solutions to enhance some aspects of users' QoE. In this respect, we have investigated a novel video bitrate adaptation and prediction mechanism based on Fuzzy logic for HAS players in Chapter 4. The proposed method takes into consideration the estimate of available network bandwidth as well as the predicted buffer occupancy level, to proactively and intelligently respond to current conditions. This leads to two contributions: First, it allows HAS players to take appropriate actions, sooner than existing methods, to prevent playback interruptions caused by buffer underrun, reducing the ON-OFF traffic phenomena associated with current approaches and increasing the QoE. Second, it facilitates fair sharing of bandwidth among

competing players at the bottleneck link. We have presented the implementation of our proposed mechanism and provided extensive empirical/QoE analysis and performance comparison with existing work. Our results show that compared to existing systems, our system has 1- better fairness among multiple competing players by almost 50% on average and as much as 80% as indicated by Jain's fairness index, and 2- better perceived quality of video by almost 8% on average and as much as 17%, according to the eMOS model.

Improved capabilities of smart phones as well as the recent advances in mobile wireless networks providing higher data rates and lower transport delays create considerable opportunities for video streaming applications, in particular OTT services. Moreover, considering the shortcomings discussed in subsection 3.2 and the challenges relating to the nature of mobile wireless networks such as inaccurate throughput measurement and inefficiency of current resource allocation methods, maximizing the average user satisfaction of the mobile users becomes more challenging for both video hosting service providers and mobile network operators. For this purpose, in addition to what we proposed for general adaptation scenarios where there is no DANE like entity available in the access network, we have presented a novel QoE optimization mechanism for HTTP Adaptive Streaming in the context of 5G wireless networks in Chapter 6. The proposed mechanism leverages recent advances in HAS specification, which incorporates new features for QoE measurements and reporting. First, we formulate a multi-objective discrete optimization problem aiming at maximizing the overall average quality, and minimizing the negative impact of temporal video quality changes for all HAS users simultaneously. Second, to take advantage of well-known continuous optimization techniques and to decrease the computational complexity, we relax the formulated problem by converting the discrete optimization problem into continuous form. Finally, we have introduced a gradient-based algorithm to solve the continuous optimization problem. The provided results of our simulations demonstrate that our system results in better perceived quality of video, measured by Mean Opinion Score (MOS), by almost 7% on average, while lowering the freezing period by almost

20% on average across HAS users when compared to other approaches where HAS users only rely on local adaptation logics.

8.2 Topics for future work

In the following, we consider interesting possibilities for extension of the current work.

- As shown in this thesis, the proposed adaptation heuristics are usually designed and tailored to networks with specific characteristics. Therefore, there is still research opportunity to design a client based adaptation mechanism that overcomes a wide range of highly dynamic network features. On this point, machine learning algorithms provide suitable approaches that allow HAS clients to exploit a wide range of useful network-related features to adapt their behaviours to optimize the Quality of Experience (QoE).
- As energy is an important resource for mobile handhelds, and switching between different representations of a video sequence may change the power consummation noticeably, another objective can be added to the optimizer proposed in Chapter 6, i.e. maximizing the battery life duration. In this regard, the level of battery and its rate of depletion can also be reported as a feedback along with other QoE related parameters. In this way, depending on the level of battery life, the optimization process can be biased to maximize the user experience or toward energy conservation.
- This thesis proposes an optimization framework in which each mobile device is assumed to have one radio connection to the radio access network. However, In LTE Release 10, new mechanisms have been added by 3GPP which enable a dual-radio terminal to have two simultaneous connections, one on 3GPP and one on non-3GPP access. From this perspective, three combinations of simultaneous connectivity are available i.e. Multi-access PDN connectivity (MAPCON), IP flow mobility (IFOM), and Non-seamless WLAN offloading (NSWO). Accordingly, different scenarios can be defined for

HAS UE with dual connectivity capability to enhance the QoE and radio resource efficiency.

- Another aspect worth investigating is resource allocation in heterogeneous cellular wireless networks. In this thesis, we consider the homogeneous network architecture where all base stations (referred to as eNB) have similar radio characteristics e.g. transmission power levels and modulation techniques. As such a deployment degrades the coverage and users near cell edges receive low capacity, a more flexible and scalable deployment approach using a hierarchical cell deployment model are introduced in the evolving LTE-Advanced systems. In this hierarchical model, namely a heterogeneous network architecture, each macro cell can be also covered by several smaller cells. Improving HAS users' experience in such a hierarchical model requires radio resource management that take into account the different requirements of multi-tier heterogeneous cellular network.

References

[1] "White paper: Cisco VNI Forecast and Methodology, 2015-2020 - Cisco." [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html. [Accessed: 14-Feb-2017].

[2] R. K. Panta, "Mobile Video Delivery: Challenges and Opportunities," *IEEE Internet Comput.*, vol. 19, no. 3, pp. 64–67, May 2015.

[3] C. Timmerer and A.C. Begen, "Over-the-Top Content Delivery: State of the Art and Challenges Ahead", in Proceedings of the ACM International Conference on Multimedia, Orlando, FL, USA, November 3, 2014, pp. 1231-1232.

[4] A.Zambelli, Smooth Streaming Technical Overview, available at: http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=03d22583-3ed6-44da-8464-b1b4b5ca7520, 2009. . [Accessed: 20-Feb-2017].

[5] R. Pantos, "HTTP live streaming", Available at: https://tools.ietf.org/html/draft-pantos-http-live-streaming-07, 2011.

[6] "Dynamic streaming in Flash Media Server 3.5 – Part 1: Overview of the new capabilities | Adobe Developer Connection." [Online]. Available: http://www.adobe.com/devnet/adobe-media-server/articles/dynstream_advanced_pt1.html. [Accessed: 14-Feb-2017].

[7] "Akamai Edge Flash Demo." [Online]. Available: http://wwwns.akamai.com/hdnetwork/demo/flash/default.html. [Accessed: 14-Feb-2017].

[8] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "QDASH," in *Proceedings of the 3rd Multimedia Systems Conference on - MMSys '12*, 2012, p. 11.

[9] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld, and P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.

[10] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "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 - NOSSDAV '12*, 2012, p. 9.

[11] R.G. Brown, "Exponential smoothing for predicting demand", USA: Arthur D. Little Inc., pp. 145-145, 1957.

[12] A. J. Smola and B. Sc Olkopf, "A tutorial on support vector regression ," *Stat. Comput.*, vol. 14, pp. 199–222, 2004.

[13] C. Chatfield, "The Holt-Winters Forecasting Procedure," *Appl. Stat.*, vol. 27, no. 3, p. 264, 1978.

[14] P. Kaufman, Smarter trading: I3 improving performance in changing markets. , New

York, USA: McGraw-Hill, 1995.

[15] S. Xiang, L. Cai, and J. Pan, “Adaptive scalable video streaming in wireless networks,” in *Proceedings of the 3rd Multimedia Systems Conference on - MMSys ’12*, 2012, p. 167.

[16] C. Liu, I. Bouazizi and M. Gabbouj, "Rate adaptation for adaptive HTTP streaming", in Proceedings of the second annual ACM conference on Multimedia systems, San Jose, CA, USA, February 23, 2011, pp. 169-174.

[17] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck, “QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming,” *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 12, no. 2, pp. 1–24, Oct. 2015.

[18] V. Joseph and G. de Veciana, "NOVA: QoE-driven optimization of DASH-based video delivery in networks", in Proceedings of INFOCOM conference, Toronto, ON, CANADA, April 27, 2014, pp. 82-90.

[19] C. Alberti *et al.*, “Automated QoE evaluation of Dynamic Adaptive Streaming over HTTP,” in *2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*, 2013, pp. 58–63.

[20] C. Chen, L. K. Choi, G. de Veciana, C. Caramanis, R. W. Heath, and A. C. Bovik, “A dynamic system model of time-varying subjective quality of video streams over HTTP,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 3602–3606.

[21] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, “ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH),” in *20th International Packet Video Workshop*, 2013, pp. 1–8.

[22] S. Winkler and P. Mohandas, “The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics,” *IEEE Trans. Broadcast.*, vol. 54, no. 3, pp. 660–668, Sep. 2008.

[23] S. Rimac-Drlje, M. Vranjes, and D. Zagar, “Influence of temporal pooling method on the objective video quality evaluation,” in *2009 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, 2009, pp. 1–5.

[24] TSGS, “TS 123 203 - V10.8.0 - Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Policy and charging control architecture (3GPP TS 23.203 version 10.8.0 Release 10),” 2012.

[25] M. Pathan, R. K. Sitaraman, and D. Robinson, Eds., *Advanced Content Delivery, Streaming, and Cloud Services*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2014.

[26] T. Stockhammer and Thomas, “Dynamic adaptive streaming over HTTP --,” in *Proceedings of the second annual ACM conference on Multimedia systems - MMSys ’11*, 2011, p. 133.

[27] L. De Cicco, S. Mascolo, and V. Palmisano, “Feedback control for adaptive live video

streaming," in *Proceedings of the second annual ACM conference on Multimedia systems - MMSys '11*, 2011, p. 145.

[28] "ITU-T and ISO/IEC JTC1, H.264 and ISO/IEC 14 496-10 (MPEG-4) AVC Recommendation. Advanced video coding for generic audiovisual services", (version 1:2003, version 2: 2004) version 3: 2005. [Accessed: 03-May-2017].

[29] "THE NIELSEN TOTAL AUDIENCE REPORT," available at: http://www.nielsen.com/us/en/insights/reports/2016/the-nielsen-total-audience-report-q3-2016.html, [Accessed: 03-May-2017].

[30] P. Lescuyer and T. Lucidarme, *Evolved packet system (EPS) : the LTE and SAE evolution of 3G UMTS*. J. Wiley & Sons, 2008.

[31] J. W. Kleinrouweler, S. Cabrero, R. van der Mei, and P. Cesar, "Modeling Stability and Bitrate of Network-Assisted HTTP Adaptive Streaming Players," in *27th International Teletraffic Congress*, 2015, pp. 177–184.

[32] T. C. Thang, H. T. Le, A. T. Pham, and Y. M. Ro, "An Evaluation of Bitrate Adaptation Methods for HTTP Live Streaming," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 693–705, Apr. 2014.

[33] K.MILLER, , E.QUACCHIO, G. GENNARI, AND A.WOLISZ. "Adaptation algorithm for adaptive streaming over HTTP". In Packet Video Workshop (PV), 19th International, IEEE, 173-178, 2012.

[34] S. Akhshabi, S. Narayanaswamy, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptive video players over HTTP," *Signal Process. Image Commun.*, vol. 27, no. 4, pp. 271–287, 2012.

[35] T.-Y. Huang *et al.*, "A buffer-based approach to rate adaptation," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 187–198, Aug. 2014.

[36] TIAN, G. AND LIU, Y. "Towards agile and smooth video adaptation in dynamic HTTP streaming". In Proceedings of the 8th international conference on Emerging networking experiments and technologies, ACM,109-120,2012.

[37] P. Xiong, J. Shen, Q. Wang, D. Jayasinghe, J. Li, and C. Pu, "NBS: A Network-Bandwidth-Aware Streaming Version Switcher for Mobile Streaming Applications under Fuzzy Logic Control," in *IEEE First International Conference on Mobile Services*, 2012, pp. 48–55.

[38] H. T. Le, D. V. Nguyen, N. P. Ngoc, A. T. Pham, and T. C. Thang, "Buffer-based bitrate adaptation for adaptive HTTP streaming," in *2013 International Conference on Advanced Technologies for Communications (ATC 2013)*, 2013, pp. 33–38.

[39] J. Jiang, V. Sekar, and Y. Sun, "DDA: Cross-Session Throughput Prediction with Applications to Video Bitrate Selection," May 2015.

[40] N. Bouten, R. de O. Schmidt, J. Famaey, S. Latré, A. Pras, and F. De Turck, "QoE-driven

in-network optimization for Adaptive Video Streaming based on packet sampling measurements," *Comput. Networks*, vol. 81, pp. 96–115, 2015.

[41] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter, "The multilayer perceptron as an approximation to a Bayes optimal discriminant function," *IEEE Trans. Neural Networks*, vol. 1, no. 4, pp. 296–298, 1990.

[42] T. Thang, Q.-D. Ho, J. Kang, and A. Pham, "Adaptive streaming of audiovisual content using MPEG DASH," *IEEE Trans. Consum. Electron.*, vol. 58, no. 1, pp. 78–85, Feb. 2012.

[43] T. C. Thang, H. T. Le, H. X. Nguyen, A. T. Pham, J. W. Kang, and Y. M. Ro, "Adaptive video streaming over HTTP with dynamic resource estimation," *J. Commun. Networks*, vol. 15, no. 6, pp. 635–644, Dec. 2013.

[44] K. Evensen *et al.*, "Demo," in *Proceedings of the 9th international conference on Mobile systems, applications, and services - MobiSys '11*, 2011, p. 355.

[45] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, 2016, pp. 1–9.

[46] A.BEBEN, P.WIŚNIEWSKI, JM BATALLA, P.KRAWIEC. ABMA+: lightweight and efficient algorithm for HTTP adaptive streaming. In Proceedings of the 7th ACM International Conference on Multimedia Systems 2016 May 10 (p. 2).

[47] D. Jarnikov, P. van der Stok, and C. C. Wust, "Predictive control of video quality under fluctuating bandwidth conditions," in *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, pp. 1051–1054.

[48] P. Juluri, V. Tamarapalli, and D. Medhi, "SARA: Segment aware rate adaptation algorithm for dynamic adaptive streaming over HTTP," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 1765–1770.

[49] P. Juluri, V. Tamarapalli, and D. Medhi, "Look-ahead rate adaptation algorithm for DASH under varying network environments," in *2015 11th International Conference on the Design of Reliable Communication Networks (DRCN)*, 2015, pp. 89–90.

[50] R. K. P. Mok, E. W. W. Chan, and R. K. C. Chang, "Measuring the quality of experience of HTTP video streaming," in *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, 2011, pp. 485–492.

[51] Z. Li, A. C. Begen, J. Gahm, Y. Shan, B. Osler, and D. Oran, "Streaming video over HTTP with consistent quality," in *Proceedings of the 5th ACM Multimedia Systems Conference on - MMSys '14*, pp. 248–258, 2014.

[52] J. De Vriendt, D.De Vleeschauwer, D.Robinson, "Model for estimating QoE of video delivered using HTTP adaptive streaming," in P*roceddings the IFIP/IEEE Intenational Symposium on Integrated Network Management,* p. 1418,2013.

[53] A. Sobhani, A. Yassine, and S. Shirmohammadi, “A fuzzy-based rate adaptation controller for DASH,” in *Proceedings of the 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video - NOSSDAV ’15*, 2015, pp. 31–36.

[54] C. Liu, I. Bouazizi, and M. Gabbouj, “Rate adaptation for adaptive HTTP streaming,” in *Proceedings of the second annual ACM conference on Multimedia systems - MMSys ’11*, 2011, p. 169.

[55] X. Yin *et al.*, “A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 325–338, Aug. 2015.

[56] C. Zhou and C.-W. Lin, “A Markov decision based rate adaption approach for dynamic HTTP streaming,” in *2015 Visual Communications and Image Processing (VCIP)*, pp. 1–4,2015.

[57] V. Martin, J. Cabrera, and N. Garcia, “Evaluation of Q-Learning approach for HTTP adaptive streaming,” in *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 293–294,2016.

[58] Dongeun Suh, Insun Jang, and Sangheon Pack, “QoE-enhanced adaptation algorithm over DASH for multimedia streaming,” in *The International Conference on Information Networking 2014 (ICOIN2014)*, pp. 497–501,2014.

[59] Y. Liu, S. Dey, D. Gillies, F. Ulupinar, and M. Luby, “User Experience Modeling for DASH Video,” in *2013 20th International Packet Video Workshop*, 2013, pp. 1–8.

[60] M. Claeys, S. Latre, J. Famaey, and F. De Turck, “Design and Evaluation of a Self-Learning HTTP Adaptive Video Streaming Client,” *IEEE Commun. Lett.*, vol. 18, no. 4, pp. 716–719, Apr. 2014.

[61] R. Houdaille and S. Gouache, “Shaping HTTP adaptive streams for a better user experience,” in *Proceedings of the 3rd Multimedia Systems Conference on - MMSys ’12*, p. 1,2012.

[62] LI, Z., ZHU, X., GAHM, J., PAN, R., HU, H., BEGEN, A. AND ORAN, D. .Probe and adapt: Rate adaptation for http video streaming at scale. Selected Areas in Communications, IEEE Journal on 32, 719-73, 2014.

[63] NAGLE, J. 1984. RFC 896: Congestion control in IP. TCP Internetworks (January 1984).

[64] J. Jiang, V. Sekar, and H. Zhang, “Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE,” in *Proceedings of the 8th international conference on Emerging networking experiments and technologies - CoNEXT ’12*, p. 97,2012.

[65] G. Cofano, L. De Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, and S. Mascolo, “Design and experimental evaluation of network-assisted strategies for HTTP adaptive streaming,” in *Proceedings of the 7th International Conference on Multimedia Systems -*

MMSys '16, pp. 1–12,2016.

[66] H. E. Egilmez, S. Civanlar, and A. M. Tekalp, "An Optimization Framework for QoS-Enabled Adaptive Video Streaming Over OpenFlow Networks," *IEEE Trans. Multimed.*, vol. 15, no. 3, pp. 710–715, Apr. 2013.

[67] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide QoE fairness using openflow-assisted adaptive video streaming," in *Proceedings of the ACM SIGCOMM workshop on Future human-centric multimedia networking - FhMN '13*, p. 15, 2013.

[68] W. Pu, Z. Zou, and C. W. Chen, "New TCP video streaming proxy design for last-hop wireless networks," in *18th IEEE International Conference on Image Processing*, pp. 2225–2228,2011.

[69] K. J. Ma and R. Bartos, "HTTP Live Streaming Bandwidth Management Using Intelligent Segment Selection," in *2011 IEEE Global Telecommunications Conference - GLOBECOM*, pp. 1–5,2011.

[70] Min Xing, Siyuan Xiang, and Lin Cai, "Rate adaptation strategy for video streaming over multiple wireless access networks," in *2012 IEEE Global Communications Conference (GLOBECOM)*, pp. 5745–5750,2012.

[71] M. Xing, S. Xiang, and L. Cai, "A Real-Time Adaptive Algorithm for Video Streaming over Multiple Wireless Access Networks," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 4, pp. 795–805, Apr. 2014.

[72] C. Mueller, S. Lederer, and C. Timmerer, "A proxy effect analyis and fair adatpation algorithm for multiple competing Dynamic Adaptive Streaming over HTTP clients," in *Visual Communications and Image Processing*, pp. 1–6,2012.

[73] A. El Essaili, D. Schroeder, D. Staehle, M. Shehada, W. Kellerer, and E. Steinbach, "Quality-of-experience driven adaptive HTTP media delivery," in *IEEE International Conference on Communications (ICC)*, pp. 2480–2485,2013.

[74] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng, "Scheduling and resource allocation for wireless dynamic adaptive streaming of scalable videos over HTTP," in *IEEE International Conference on Communications (ICC)*, pp. 1681–1686,2014.

[75] F. Li, P. Ren, and Q. Du, "Joint Packet Scheduling and Subcarrier Assignment for Video Communications Over Downlink OFDMA Systems," *IEEE Trans. Veh. Technol.*, vol. 61, no. 6, pp. 2753–2767, Jul. 2012.

[76] A. El Essaili, D. Schroeder, E. Steinbach, D. Staehle, and M. Shehada, "QoE-Based Traffic and Resource Management for Adaptive HTTP Video Delivery in LTE," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 6, pp. 988–1001, Jun. 2015.

[77] S. Cicalo, N. Changuel, V. Tralli, B. Sayadi, F. Faucheux, and S. Kerboeuf, "Improving QoE and Fairness in HTTP Adaptive Streaming Over LTE Network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 12, pp. 2284–2298, Dec. 2016.

[78] N. Bouten, M. Claeys, S. Latre, J. Famaey, W. Van Leekwijck, and F. De Turck, "Deadline-based approach for improving delivery of SVC-based HTTP Adaptive Streaming content," in *IEEE Network Operations and Management Symposium (NOMS)*, pp. 1–7,2014.

[79] "The State of MPEG-DASH 2015 - Streaming Media Magazine." [Online]. Available: http://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=102826&PageNum=2. [Accessed: 03-May-2017].

[80] L. Toni, R. Aparicio-Pardo, G. Simon, A. Blanc, and P. Frossard, "Optimal set of video representations in adaptive streaming," in *Proceedings of the 5th ACM Multimedia Systems Conference on - MMSys '14*, pp. 271–282, 2014.

[81] J. Janssen, T. Coppens, and D. De Vleeschauwer, "Quality Assessment Of Video Streaming In The Broadband Era.", In *Proceedings of ACIVS (Advanced Concepts for Intelligent Vision Systems),* 2002.

[82] S. Liu and J. Y.-L. Forrest, *Grey systems : theory and applications*. Springer, 2011.

[83] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—I," *Inf. Sci. (Ny).*, vol. 8, no. 3, pp. 199–249, Jan. 1975.

[84] L. Rizzo and Luigi, "Dummynet," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 1, pp. 31–41, Jan. 1997.

[85] "Red Bull PlayStreets 2017: ++ INFOS & LIVE STREAM ++." [Online]. Available: http://www.redbull.com/ca/en/snow/events/1331691404612/red-bull-playstreets. [Accessed: 14-Feb-2017].

[86] "Distributed DASH Datset | ITEC – Dynamic Adaptive Streaming over HTTP." [Online]. Available: http://www-itec.uni-klu.ac.at/dash/?page_id=958. [Accessed: 03-May-2017].

[87] "iPerf - The TCP, UDP and SCTP network bandwidth measurement tool." [Online]. Available: https://iperf.fr/. [Accessed: 03-May-2017].

[88] Jiwoo Park and Kwangsue Chung, "Rate adaptation scheme for HTTP-based streaming to achieve fairness with competing TCP traffic," in *International Conference on Information Networking (ICOIN)*, pp. 222–226, 2015.

[89] I. T. Young, "Proof without prejudice: use of the Kolmogorov-Smirnov test for the analysis of histograms from flow systems and other sources," The journal of histochemistry and cytochemistry : official journal of the Histochemistry Society vol. 25, no. 7, pp. 935–941, 1977.

[90] W. J. Conover, *Practical nonparametric statistics*. Wiley, 1999.

[91] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination For Resource Allocation In Shared Computer Systems," Sep. 1998.

[92] Samukic, Antun. "UMTS universal mobile telecommunications system: development of

standards for the third generation." In Global Telecommunications Conference,. GLOBECOM. The Bridge to Global Integration. IEEE, vol. 4, pp. 1976-1983, 1998.

[93] O. Oyman and S. Singh, "Quality of experience for HTTP adaptive streaming services," *IEEE Commun. Mag.*, vol. 50, no. 4, pp. 20–27, Apr. 2012.

[94] "TS 29.213 v11.0.0, "Policy and Charging Control Signalling Flows and Quality of Service (QoS) Parameter mapping, "2008.

[95] "TS 23.203 v11.3.0, "Policy and Charging Control Architecture, " 2008.

[96] M. Olsson and C. Mulligan, *EPC and 4G packet networks : driving the mobile broadband revolution*. Academic Press, 2012.

[97] TSGS, "TR 126 938 - V12.0.0 - Universal Mobile Telecommunications System (UMTS); LTE; Packet-switched Streaming Service (PSS); Improved support for dynamic adaptive streaming over HTTP in 3GPP (3GPP TR 26.938 version 12.0.0 Release 12)," 2014.

[98] J.Funge, M.Watson, W.Wei, D.Chen, inventors; Netflix, Inc., "Measuring user quality of experience for a streaming media service". United States patent US 9,479,562. 2016 Oct 25.

[99] J. Chen, "A Scheduling Framework for Adaptive Video Delivery over Cellular Networks Categories and Subject Descriptors," *Proc. ACM MobiCom*, no. c, 2013.

[100] I. Das and J. E. Dennis, "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, Aug. 1998.

[101] B. Li, Z. Wang, J. Liu, and W. Zhu, "Two decades of Internet video streaming," *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 9, no. 1s, pp. 1–20, Oct. 2013.

[102] S.Boyd, L.Vandenberghe. "Convex optimization". Cambridge university press; 2004 .

[103] "CVX: Matlab Software for Disciplined Convex Programming | CVX Research, Inc." [Online]. Available: http://cvxr.com/cvx/. [Accessed: 14-Feb-2017].

[104] "LTE Module — Model Library." [Online]. Available: https://www.nsnam.org/docs/models/html/lte.html. [Accessed: 14-Feb-2017].

[105] G. Piro, N. Baldo, and M. Miozzo, "An LTE module for the ns-3 network simulator."

[106] C.Müller, S.Lederer, and C.Timmerer. "An evaluation of dynamic adaptive streaming over HTTP in vehicular environments." In *Proceedings of the 4th Workshop on Mobile Video*, pp. 37-42. ACM, 2012.

[107] H. Sun, A. Vetro, and J. Xin, "An overview of scalable video stream- ing," Wireless Commun. Mobile Comput. , vol. 7, no. 2, pp. 159–172, Feb. 2007.

[108] S.Akhshabi, L.Anantakrishnan, C.Dovrolis, and A. C. Begen. "Server-based traffic shaping for stabilizing oscillating adaptive streaming players." Proceeding of the 23rd ACM

workshop on network and operating systems support for digital audio and video. ACM, (pp. 19-24), Feb 2013.