Content-aware and context-aware adaptive video streaming over HTTP

Ognen OGNENOSKI

Faculty of Science, Engineering and Computing

Kingston University

This dissertation is submitted for the degree of

Doctor of Philosophy

School of Computer Science and
Mathematics

July 2016

BRITISH LIBRARY

IMAGING SERVICES NORTH
Boston Spa, Wetherby
West Yorkshire, LS23 7BQ
www.bl.uk

PAGINATED BLANK PAGES ARE SCANNED AS FOUND IN ORIGINAL THESIS

NO INFORMATION IS MISSING

Nullius in verba

Acknowledgements

The completion of the PhD dissertation arouses an exhilarating feeling which makes up for all the encountered difficulties and struggles during the studies. It was a rewarding and life-shaping experience and here I would like to acknowledge and express my gratitude to a selected group of people who were always there.

Foremost, I would like to thank my supervisor Prof. Maria Martini for giving me this marvelous opportunity to pursue my PhD degree at Kingston University, for mentoring and encouraging me to strive for more, and for the invaluable feedback during the ongoing research work. Further, I would like to thank my industrial supervisor Peter Amon from Imaging and Computer Vision group, Siemens, Munich. His support, guidance and approach were essential and extremely helpful throughout my research.

I would like to thank all the members in the Wireless and Multimedia Networking (WMN) Research Group at Kingston University. In particular, Mr. Manzoor Razaak, Dr. Moustafa Nasralla, Dr. Thushara Hewage, Dr. Sergio Pezzulli, Dr. Olayinka Adigun, Mr. Fariborz Entezami, Ms. Bola Isawhe, Ms. Silvia Rossi and Mr. Nabajeet Barman for all the discussions, joint works, and for generating pleasant and enjoyable working environment.

In these last lines of my acknowledgment, I would like to express my greatest thankfulness to my parents Bratica and Zoran, and to my brother Mitko. I dedicate the PhD dissertation to the three of you - what you mean to me can not be expressed in words.

Abstract

Adaptive HTTP video streaming technologies are rapidly becoming the main method for video delivery over the Internet. From a conceptual viewpoint, adaptive HTTP video streaming systems enable adaptation of the video quality according to network conditions (link-awareness), content characteristics (content-awareness), user preferences (user-awareness) or device capabilities (device-awareness). Proprietary adaptive HTTP video streaming platforms from Apple, Adobe and Microsoft preceded the completion of a standard for adaptive HTTP video streaming, i.e., the MPEG-DASH standard. The dissertation presents modeling approaches, experiments, simulations and subjective tests tightly related to adaptive HTTP video streaming with particular emphasis on the MPEG-DASH standard.

Different case studies are investigated through novel models based on analytical and simulation approaches. In particular, adaptive HTTP video streaming over Long Term Evolution (LTE) networks, over cloud infrastructure, and streaming of medical videos are investigated and the relevant benefits and drawbacks of using adaptive HTTP video streaming for these cases are highlighted. Further, mathematical tools and concepts are used to acquire quantifiable knowledge related to the HTTP/TCP communication protocol stack and to investigate dependencies between adaptive HTTP video streaming parameters and the underlying Quality of Service (QoS) and Quality of Experience (QoE). Additionally, a novel method and model for QoE assessment are proposed, derived in a specific experimental setup. A more general setup is then considered and a QoE metric is derived. The QoE metric expresses the users' quality for adaptive HTTP video streaming by taking into consideration rebuffering, video quality and content-related parameters. In the end, a novel analytical model that captures the user's perception of quality via the experienced delay during streaming navigation is derived. The contributions in the dissertation and the relevant conclusions are obtained by simulations, experimental demo setups, subjective tests and analytical modeling.

Table of contents

List of figures xvii

List of tables xix

1 Introduction 1

1.1 Objectives and description . . . 2
1.2 Research methodology . . . 3
1.3 Contributions to knowledge . . . 4
1.4 Dissertation structure . . . 4

2 Literature overview 7

2.1 Background information . . . 8
2.1.1 Protocols and layer services . . . 8
2.1.2 HTTP/TCP protocol suite . . . 10
2.1.2.1 HTTP protocol overview . . . 10
2.1.2.2 TCP protocol overview . . . 11
2.1.3 Video signals, compression and quality . . . 13
2.1.3.1 Video signals . . . 13
2.1.3.2 Video compression principles . . . 15
2.1.3.3 Video quality . . . 15
2.2 Adaptive video streaming: state of the art . . . 16
2.2.1 Video streaming: protocol viewpoint . . . 17
2.2.2 MPEG-DASH overview . . . 19
2.2.2.1 Media Presentation Description . . . 20
2.2.2.2 Segment formats and types . . . 21
2.2.2.3 Media Presentation . . . 21
2.2.2.4 Profiles . . . 22
2.2.3 Relevant areas . . . 22

2.2.3.1 Content encoding and segmentation 22
2.2.3.2 Adaptation and QoS 24
2.2.3.3 Quality of Experience 27
2.2.3.4 Navigation experience 30
2.3 Chapter summary and conclusions 31

3 Case studies **33**
3.1 DASH-based video streaming over LTE 34
3.1.1 Related Work 34
3.1.2 System Model 36
3.1.2.1 Server & clients 37
3.1.2.2 eNodeB scheduling 37
3.1.2.3 Access and core links 38
3.1.3 Performance results 39
3.1.4 Study summary 42
3.2 DASH live video streaming over cloud 43
3.2.1 Related work 43
3.2.2 Experimental setup 45
3.2.2.1 Source side 45
3.2.2.2 Cloud server 46
3.2.2.3 Client side 46
3.2.3 Evaluation results 46
3.2.3.1 Throughput 47
3.2.3.2 Packet inter-arrival time 48
3.2.3.3 Confidence intervals 48
3.2.3.4 End-to-end QoS 50
3.2.4 Study summary 51
3.3 Medical video streaming utilizing DASH 51
3.3.1 Evaluation setup 52
3.3.1.1 Video encoding and segmentation 52
3.3.1.2 Streaming model 54
3.3.1.3 Client adaptation approaches 54
3.3.2 Performance evaluation 55
3.3.3 Study summary 59
3.4 Chapter summary and contributions 59

4 Teletraffic and statistical analysis of DASH systems 61
4.1 Teletraffic analysis . . . 62
4.1.1 Segment-based teletraffic model for DASH . . . 62
4.1.1.1 Model boundaries . . . 65
4.1.1.2 Segment selection . . . 66
4.1.1.3 Model metrics . . . 66
4.1.2 Performance results . . . 67
4.1.3 Study summary . . . 69
4.2 Statistical analysis . . . 70
4.2.1 Methodology . . . 71
4.2.2 System model and simulations . . . 72
4.2.2.1 Rebuffering parameters . . . 75
4.2.2.2 Video quality parameters . . . 76
4.2.3 Statistical data analysis . . . 76
4.2.3.1 Rebuffering frequency . . . 77
4.2.3.2 Average rebuffering duration . . . 77
4.2.3.3 Total rebuffering duration . . . 80
4.2.3.4 Video quality parameters . . . 81
4.2.4 Comparative system analysis . . . 81
4.2.5 Concluding remarks . . . 84
4.3 Chapter summary and contributions . . . 84

5 Quality of Service and Quality of Experience for DASH 87
5.1 A study on QoS and QoE for DASH . . . 88
5.1.1 Related work . . . 88
5.1.2 System model . . . 89
5.1.2.1 Content encoding and segmentation . . . 89
5.1.2.2 Adaptation algorithm . . . 90
5.1.2.3 Simulation and subjective tests . . . 91
5.1.3 Evaluation results . . . 92
5.1.3.1 Simulation results . . . 92
5.1.3.2 MOS scores . . . 96
5.1.4 QoE model and proposed method . . . 96
5.1.4.1 MOS scores from subjective tests . . . 98
5.1.4.2 MOS model . . . 99
5.1.4.3 Proposed method . . . 101
5.2 Content-aware quality metric for DASH . . . 104

5.2.1 Related work 104
5.2.2 Proposed QoE metric 107
5.2.2.1 CAQ metric formulation 108
5.2.3 Subjective tests & Evaluations 111
5.2.3.1 Testing session #1 112
5.2.3.2 Testing session #2 116
5.2.3.3 Evaluations 116
5.3 Chapter summary and contributions 120

6 Navigation experience for DASH **123**
6.1 Use case for non-linear DASH 123
6.2 Model for non-linear DASH video streaming 125
6.2.1 Derivation and evaluation of the model 126
6.3 Chapter summary and contribution 132

7 Conclusion **133**
7.1 Summary of the works 134
7.2 Future research directions 137

Acronyms **141**

References **145**

Appendix A: List of publications **157**

List of figures

2.1 Layer service(s). Extracted from [1]. 9
2.2 TCP pipe between HTTP client and HTTP server. Extracted from [1]. . . . 10
2.3 TCP high-level view. Extracted from [1]. 12
2.4 TCP sliding window at the sender and at the receiver. Extracted from [1]. . 13
2.5 High-level view of adaptive video streaming over HTTP. 18
2.6 MPEG-DASH client-server model. Extracted from [2]. 19
2.7 MPD hierarchical model. Extracted from [2]. 20

3.1 System model: topology and protocol view. 36
3.2 Rebuffering probability for different scheduling approaches. 40
3.3 Rebuffering duration for different scheduling approaches. 41
3.4 Video representation for different scheduling approaches. 42
3.5 Experimental setup for HTTP streaming over cloud. 45
3.6 Measurements and CDFs for uplink/downlink throughput for different segment sizes. 47
3.7 Measurements and CDFs for uplink/downlink packet inter-arrival time for different segment sizes. 49
3.8 CI for uplink/downlink throughput with different segment sizes. 50
3.9 CI for uplink/downlink packet inter-arrival time with different segment sizes. 50
3.10 Comparison between a frame from different representations. 53
3.11 Different approaches at the client. 54
3.12 Re-buffering frequency[for different segment sizes and margin indices. . . . 56
3.13 Duration of re-buffering for different segment sizes and margin indices. . . 57
3.14 Added time in playout. 57
3.15 PSNR results for different segment sizes and adaptation approaches at the client. 58

4.1 System model: network and teletraffic view. 62

4.2 Arrivals and serving rates in the queue. 63
4.3 DASH representations. 64
4.4 Snapshot of the model's input trace. 68
4.5 Average probability of buffer overflow. 68
4.6 Average probability of empty buffer. 69
4.7 Average probability of active buffer. 70
4.8 Methodology workflow. 71
4.9 Download and playout timelines at the client side. 73
4.10 Boxplots for the rebuffering frequency. 78
4.11 Boxplots for the average rebuffering durations where averaging is done over time. 78
4.12 Boxplots for the total rebuffering duration. 80
4.13 Comparison of segment sizes. 83

5.1 Peak Signal-to-Noise Ratio (PSNR) and bitrate variations: US sequence. . . 93
5.2 MOS for different segment sizes and Group of Pictures (GOP) sizes. 97
5.3 MOS scores for video streaming at a constant bitrate, SOCCER sequence. . 98
5.4 MOS scores for DASH video streaming, all sequences. 99
5.5 MOS score and MOS model vs. rebuffering frequency 100
5.6 Switching between DASH representations. 102
5.7 K ratio for DASH video streaming and video streaming at different constant bitrates. 103
5.8 Assignment of content related parameters. 110
5.9 Subjective quality evaluation setup. 111
5.10 Graphical user interface of the real-time quality meter. 112
5.11 PSNR and bitrate variations for testing session #1. 114
5.12 MOS scores for the first testing session. 117
5.13 PSNR and bitrate variations for testing session #2. 118
5.14 MOS scores for the second testing session. 119
5.15 State of the art metric vs. CAQ metric. 120

6.1 General use case for non-linear DASH. 124
6.2 Procedures and specific actions at the client side related to the expected navigation delay. 127
6.3 Navigation/seeking outcomes for arbitrary DASH segment. 129
6.4 Expected navigation delay for different GOP sizes. 132

List of tables

3.1 Scheduler performances for 40 users. 41
3.2 End-to-end QoS performance 50
3.3 Average PSNR for different segmentation/representations 53

4.1 Rebuffering frequency: summary statistics 79
4.2 Average rebuffering duration: summary statistics 79
4.3 Total rebuffering duration: summary statistics 82
4.4 Content quality: summary statistics 82

5.1 Average PSNR. 94
5.2 Average video bitrate. 94
5.3 Rebuffering frequency. 95
5.4 Model parameters. 101
5.5 Considered QoE metrics. 111
5.6 SI and TI for the sequences 113
5.7 Average PSNR of the segments. 113
5.8 Streaming impairments 115
5.9 Average PSNR of the segments. 116
5.10 MOS scores and MOS values from the metrics 121

6.1 Parameters and notations for the model. 128

Chapter 1

Introduction

"The improved 'limitless-distance' telephone was presently introduced, and the daily doings of the globe made visible to everybody, and audibly discussable too, by witnesses separated by any number of leagues.". That is a description of an invention called telelectroscope from a 1898 short story by Mark Twain called "From the 'London Times' of 1904," [3]. It is reasonable to conjecture that the telelectroscope is the first presented conceptual predecessor of the Internet as a network for transmitting video at a global scale. Since then, things have changed enormously. The Moore's law [4] outlines the immense growth in technological processing power/capabilities and the scale at which the Internet network has spread across the globe is staggering, allowing its users to communicate a diverse collection of data, including text, voice and video. Nowadays, data communication represents an important cornerstone of our culture, lifestyle, business, and entertainment. Recent projections and forecasts [5, 6] show that the demand for video data is greatly increasing by making full use of the established Internet network and the plethora of user devices.

Internet is a network of different interconnected networks of local and global scope that link billions of devices worldwide via a broad range of networking technologies, including electronic, wireless and optical. A large number of user devices like desktop and laptop computers, tablets and smartphones can communicate via the Internet network by following a specific set of functions and procedures specified in a protocol or a standard. Different communication protocols and standards are used depending on the device capability, underlying network infrastructure and the data type, i.e., the set of requirements for text, voice and video communication significantly differs. Although the data type can remain the same, the service offered to the users can be different and this additionally introduces complexity. For example, chat and email services are based on the same text data communication, however the delay tolerance for these services is different. Chat service is designed to emulate real

world conversation, therefore low delays are required, whereas email emulates exchange of messages that are checked occasionally, therefore higher delays are acceptable.

The dissertation broadly focuses on video communication over the Internet network. Similarly to the previously discussed example, video data communication is the basis for different services. In general, these services can be grouped in two broad categories: download/upload of video data and streaming of video data [7]. Download/upload assumes full transfer of the video data in a particular direction followed by a viewing process, whereas streaming assumes delivery of video data for immediate display, i.e., the viewing starts without the completion of the full transfer of the video data. Only the second category is considered in the dissertation by adopting an approach of using models which aim to make a particular part, aspect or feature easier to understand, define, quantify, or simulate. These models are labeled as video streaming models and represent an abstraction of the video delivery to a viewing device for immediate display. In the literature, there are three general groups of video streaming models: traditional streaming, progressive streaming and adaptive streaming. These groups are differentiated by several aspects, including the used protocols, scalability and implementation complexity.

The works and contributions presented in this dissertation are related the design of content-aware and context-aware solutions for adaptive HTTP video streaming. The content-awareness refers to obtaining knowledge or cognizance regarding the video content features and availability, whereas the context-awareness refers to the design principle that adapts to the dynamic changes in the network environment. Back in 2012, a study showed that adaptive video streaming constitutes only 17% of the video traffic in the Internet network [5]. However, an additional study showed an increase of an average 77% a year in the following three years, thus 51% of Internet video was delivered by adaptive video streaming by 2015 [8]. In addition, future projections presented in a recent study indicate that over two-thirds of the worlds mobile data traffic will be adaptive video by 2018 [6]. These statistics clearly indicate the expansiveness of adaptive video streaming deployment which is followed by a continuous growth of research works in the scientific literature in recent years. The research presented in this dissertation is along the aforementioned lines.

1.1 Objectives and description

Five main objectives are defined in the dissertation. First, background knowledge related to adaptive video streaming is acquired and a comprehensive account of the works in the literature is compiled in order to better understand the technology and to identify potential research directions. This enables in-depth understanding of adaptive video streaming at

a system level including the respective system components, thus facilitating the modeling which is essential for the following objectives. Second, different system models are proposed, implemented, and evaluated corresponding to considered use cases. As a result, the benefits and drawbacks of applying adaptive video streaming system to the use cases are observed. Third, several mathematical tools and concepts are applied in order to obtain quantifiable knowledge related to adaptive video streaming system performances. In particular, observations are made with respect to the configuration of the used protocols and the dependency between system parameters and Quality of Service (QoS) and Quality of Experience (QoE) related parameters. Fourth, in-depth study of QoS and QoE is presented. In the end, the fifth objective is to investigate the suitability of the systems for non-linear video streaming which is guided by the users. Each objective is achieved and the related material is presented in a separate chapter.

1.2 Research methodology

The methodology adopted in the dissertation is based upon quantitative and qualitative methods [9]. Quantitative methods are used for the collection of quantitative data and measuring parameters, whereas qualitative methods/concepts are used to understand meanings and experience of observable parameters related to the adaptive video streaming system. The contributions provided in the dissertation are obtained by combined usage of analytical models, simulations, test-bed experiments, and subjective tests as representatives of the quantitative and qualitative methods and concepts.

The initial point in the research work-flow is a very broad and general concept. A major requirement embedded in the concept is the correspondence with a particular scenario. When the concept and the corresponding scenario are defined, the appropriate quantitative/qualitative methods are applied in order to map the concept into a system model. The system model is then defined and a detail description of each of its components is presented. In the end, as an output of the system model analysis, a concrete results, conclusions and observations are obtained. The system modeling and the results in the dissertation are obtained using: 1) simulation tools including Matlab [10], R package [11], DASHEncoder [12], JM [13], LTE Simulator [14]; 2) test-bed setup including Wowza streaming cloud server and the client engine; and 3) a new mobile/tablet application (Android APP) developed within the Kingston Wireless Multimedia and Networking (WMN) lab which was used to collect subject's quality scores in real-time.

1.3 Contributions to knowledge

The dissertation is a collection of works in several narrowly defined topics related to adaptive HTTP video streaming. As a result of these works, the following main contributions to knowledge are presented in the dissertation:

- A novel system model for adaptive HTTP video streaming from the Internet network to mobile users, i.e., users of the Long-Term Evolution (LTE) technology in the last hop link, presented in Chapter 3, Section 3.1. The novel aspect in the modeling is the combined usage of analytical dependencies and simulation traces to model the end-to-end adaptive HTTP video streaming for LTE users.

- A novel model based on teletraffic queuing theory for probabilistic evaluation of adaptive HTTP video streaming where probabilistic metrics are derived and evaluated, presented in Chapter 4, Section 4.1. The novel aspect is the link between the probabilistic metrics derived from the model and parameters at the server and the client side.

- A content-aware quality metric for capturing the quality for DASH video streaming, presented in Chapter 5, Section 5.2. The novel aspect is consideration of rebuffering, video quality and content-related parameters into a single metric.

- A novel analytical model for navigation experience in non-linear DASH video streaming, presented in Chapter 6, Section 6.2. The novel aspect is expressing the expected navigation delay as a function of DASH parameters for random navigation/seeking.

The main contributions listed above and the additional works presented in this dissertation are published or submitted for peer-reviewed publications. Further details on the publications are provided in Appendix A.

1.4 Dissertation structure

The dissertation is organized in seven chapters. A brief, high-level introduction related to the research work is presented in the first chapter, whereas the last chapter presents a detailed summary of the contributions to knowledge and provides a discussion on possible future research. The remaining chapters are structured as follows.

In the second chapter, relevant background knowledge is presented, including the required protocol engineering and the video encoding fundamentals and quality aspects. Further,

this chapter presents a comprehensive state of the art in adaptive video streaming including the recent achievements in the area. The second chapter is organized in two sections: 1) section on background knowledge, where the Hypertext Transfer Protocol (HTTP) and Transmission Control Protocol (TCP) protocols, and the video encoding fundamentals and video quality aspects are presented; and 2) section on state-of-the-art in adaptive streaming, where an overview of the emerging MPEG-DASH standard and the works in the relevant areas are presented. This chapter provides the required theoretical knowledge for the research presented in the following chapters.

The third chapter presents three case studies where different models related to adaptive video streaming are used. Therefore, this chapter is organized in three sections corresponding to each case study: 1) in the first section relevant performances for adaptive video streaming from Internet to LTE multi-user setup are investigated; 2) the second section targets live adaptive video streaming over cloud and it presents results obtained through experiments carried out in an operational test-bed; and 3) the third section explores the possibility and discusses the requirements for using adaptive video streaming systems in the case of medical videos. These studies show the application of adaptive video streaming system in different use cases and provide useful guidelines for understanding the underlying technology with respect to different system parameters.

In the fourth chapter, mathematical concepts are used to obtain quantifiable knowledge for the adaptive HTTP streaming system. The chapter is organized in two sections: 1) section on analysis based on queuing theory, i.e., a probabilistic evaluation of the used protocols with respect to configuring system parameters is presented; and 2) section on statistics-based approach for obtaining measurable and observable from the adaptive video streaming system. Two models are discussed in the respective sections and quantitative insights on drawback and benefits with respect to configuration and output parameters are highlighted. This chapter presents valuable inputs for the models discussed in the following chapters.

The fifth chapter investigates the user's quality perception when video is streamed using adaptive video streaming system and is organized in two sections: 1) the first section presents a comprehensive QoS study and proposes a QoE model and method; and 2) the second section presents a QoE metric for adaptive video streaming that considers information related to the content.

In the sixth chapter, the term navigation experience is introduced to capture the users' perception of quality for non-linear video streaming. This chapter presents a novel analytical model derived to capture the navigation experience through the expected navigation delay by using several DASH system parameters.

Chapter 2

Literature overview

The wide range of text, voice and video services, i.e., multimedia services in the Internet have either no explicit requirements or have explicit requirements embedded in the data regarding the presentation timings [15]. An example of service with no explicit presentation timing requirement is the retrieval of an image from a web server to the browser of the client, whereas an example of service with embedded presentation timing requirement is the retrieval of an audio or video from a web server to the player of the client. Service with no explicit presentation timing requirement are delivered over the Internet network by using download/upload model, i.e., the media data for the service is first completely received, possibly cached in memory buffer/local file, and then decoded and viewed/played. The services with embedded presentation timing requirement can not be delivered by using the dowload/upload model since the embedded requirement will not be met in such case. Therefore, these services are delivered over the Internet network by using the streaming model, i.e., the media data for these services is being viewed/played while reception is in progress, and the media data can be visualized or played only once all the relevant amount of data has been received. The dissertation focuses only on multimedia services delivered by using the streaming model. More precisely, the topics presented in the dissertation are tightly linked to video delivery over Internet by using an emerging technology for streaming, i.e., adaptive HTTP video streaming which utilizes the HTTP/TCP protocol suite.

This chapter is organized in two main sections. Section 2.1 presents background information on communication protocols fundamentals (particularly HTTP and TCP) and on video signals, compression and quality. Section 2.2 presents a state of the art for adaptive HTTP video streaming. The state of the art includes a protocol viewpoint on streaming, a high-level view of adaptive video streaming, an overview of the emerging MPEG-DASH standard, and survey on notable works in areas relevant for the research presented in the following chapters. Section 2.3 concludes the chapter.

2.1 Background information

The background on communication protocols includes layer services in the referent Open Systems Interconnection (OSI) model and a description of the main features of the HTTP/TCP protocol suite, whereas the background on video signals, compression and quality includes a general description of the video signal, elaborates the basic principles for compressing video signals and discusses video quality.

2.1.1 Protocols and layer services

A communication network is a set of equipment and facilities that provides transfer of information between users located at various geographical points, allowing them to gather information and exercise action at a distance [1]. These two network capabilities are achieved via communication protocols. A protocol is a set of rules that governs how the users interact with the main purpose to deliver the information. The required information delivery functions in the communication network are organized into related and manageable sets called layers. The set of protocols that refer to the functionality of these layers defines the network architecture. In the early 1970s, the OSI developed a layered reference model that provided a framework for defining the overall network architecture and was intended to facilitate development of new communication standards. The OSI reference model defines the following seven layers: physical, link, network, transport, session, presentation, and application [16]. The session, presentation and application layer are often jointly considered as application layer, hence only five layers are considered as adopted in this dissertation.

In OSI terminology, a process in each layer on one machine communicates with the peer process in the same layer on the other machine by exchanging Protocol Data Units (PDUs). Each PDU contains control information, i.e., header information, and the actual user information in the form of Service Data Unit (SDU). The processes at layer n, where n assumes values $1, 2, .., 5$ for the physical, link, network, transport, and application layer, respectively, are referred as layer n entities and the protocol that governs the behavior of layer n is labeled as layer n protocol. Communication is enabled by using the service provided from the underlying layer, i.e., the entity at layer $n+1$ uses the service provided by layer n. The exchange of information between layers is done through a software port called Service Access Point (SAP), labeled n-SAP. The PDU of the entity at layer $n+1$ is passed through the n-SAP and becomes SDU for the entity at layer n, which attaches its control header that specifies the used protocol (encapsulation) and forms its PDU. The layering approach is shown in Figure 2.1.

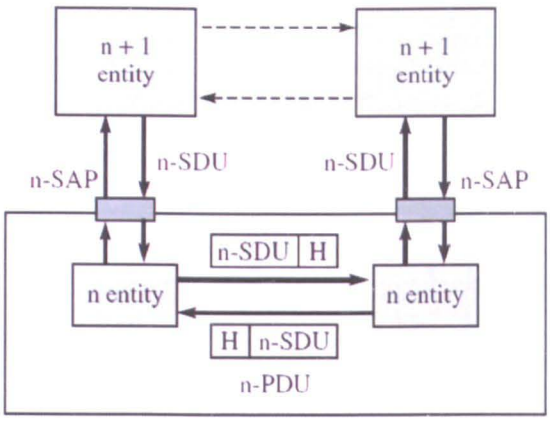


Fig. 2.1 Layer service(s). Extracted from [1].

This layering approach enables independence between the entities in each layer and reliance only on the underlying layer services. These services can be grouped in two main categories:

- Connection-oriented service. The connection-oriented service has three phases: establishing connection between the two layer SAPs by negotiating connection parameters and initializing state information (buffer allocations, sequence numbers, flow control, etc.), transfer of the n-SDU by using the layer n protocol, and terminating the connection hence releasing the allocated network resources;

- Connectionless service. The connectionless service does not require connection setup and each SDU is transferred directly from SAP to SAP and the headers contain all required addressing information.

The connection-oriented service is mostly confirmation based, whereas the connectionless one can be confirmed or unconfirmed depending on the protocol. A protocol is only concerned with the message exchange between the corresponding peers at the same layer. This is shown in Figure 2.2, where an HTTP client-server communication is established between two application layer entities. The HTTP protocol only defines the communication between the HTTP client and HTTP server entities which are not directly connected in the network. Therefore, a connection between these entities should be established. In the case of HTTP, this connection should provide a two-way, sequential and error free communication. This type of connection-oriented service between two machines in the network is provided with the TCP protocol. Hence, TCP provides a communication pipe between the HTTP client and the HTTP server.

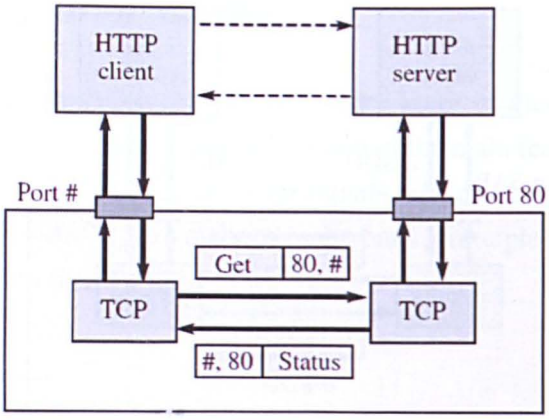


Fig. 2.2 TCP pipe between HTTP client and HTTP server. Extracted from [1].

2.1.2 HTTP/TCP protocol suite

The HTTP/TCP protocols are widely used in the Internet. This protocol suite represents the foundation of data communication for the World Wide Web (WWW).

2.1.2.1 HTTP protocol overview

HTTP is an application-level, pull-based, client-server stateless protocol [17]. The client uses this protocol to pull information from the server, i.e., the client sends requests and the server returns responses to these requests containing the required content. Each request is independent and contains the following three parts: method (indicates the type of request, the Uniform Resource Locator (URL), and the version of the HTTP protocol), header (used to specify optional parameters for the request), Multipurpose Internet Mail Extensions (MIME) document (optional MIME document can be attached to the request). The server response contains the following three parts: status line (indicates whether the request has been successful), header (additional information about the response), MIME document (optional MIME document can be attached to the response).

There is a broad range of HTTP request and response methods. For example, the GET method is used for retrieving documents from the server, the HEAD method is a variant of the GET method used for retrieving the header lines without retrieving the entire document (used for availability checks), and the POST method is commonly used for storing documents on the server. The headers in both the HTTP request and HTTP response are discussed in details in [18]. The exchanged MIME documents in both the request and the response refer to the representation of the document with the size and the type of the document being stated in the appropriate MIME header. The initial design of the HTTP protocol targeted sharing of self-containing documents in a simple manner. Therefore, the HTTP designers chose to open

a TCP connection for every HTTP request, i.e., for every Uniform Resource Locator (URL). These TCP connections are called non-persistent connections since they are closed after the transmission of a single web object. The rationale behind this design was that the client remains idle while reading the retrieved document. This is the HTTP version 1.0. However, with the development of rich web content (images, videos) this design became inefficient. Therefore, an extension to the HTTP protocol was proposed to support persistent connections (HTTP version 1.1 [19]), which supports sending multiple web objects across a single TCP connection. Two new headers are introduced for configuring the persistent TCP connections:

- Connection. This header is used by the client with Keep-Alive argument to indicate that a persistent TCP connection is expected. If used by any entity (client or server) with the Close argument, that indicates that the TCP connection is going to be closed after the current request/response.

- Keep-Alive. This header is used by the server to inform the client how it will use the TCP connection. It contains two parts that indicate the maximum number of HTTP requests served by the connection and the timeout period after which the idle connection is closed.

The new version of the HTTP protocol is the HTTP version 2.0 [20], which is developed by the Hypertext Transfer Protocol Bis (HTTPBis) working group of the Internet Engineering Task Force (IETF). The HTTP/2 standard enables a more efficient use of network resources and a reduced perception of latency by introducing header field compression and allowing multiple concurrent messages on the same connection (request-response pipelining). The backward compatibility with the previous protocol versions is fully maintained.

2.1.2.2 TCP protocol overview

The TCP protocol [21] provides a full-duplex connection between two application layer processes by exchanging PDUs labeled as TCP segments. TCP uses a form of Selective Repeat Automatic Repeat reQuest (ARQ) with positive acknowledgments based on a sliding window mechanism to provide reliable, connection-oriented and in-sequence service for the application information across unreliable connectionless network Internet Protocol (IP) layer. The application information is treated as a byte stream by the TCP protocol and can be split and organized in the most suitable manner to facilitate reliable segment delivery. Figure 2.3 shows a high-level view of the TCP functionality. The byte stream from the sender application is stored in the send buffer, organized in segments and send via the network layer. When correctly received, the segment is stored in the receive buffer and can be inserted into

the byte stream at the application level at the receiver. The receiver confirms the correct reception of the segment by sending an ACK packet to the sender. Every TCP connection is established by using a three-way handshake procedure that ensures that the sender and the receiver agree on their initial segments sequence numbers which are essential for the functionality of the sliding window mechanism. The relevant pointers of the transmitter and the receiver window that compose the TCP sliding window mechanism are depicted in Figure 2.4.

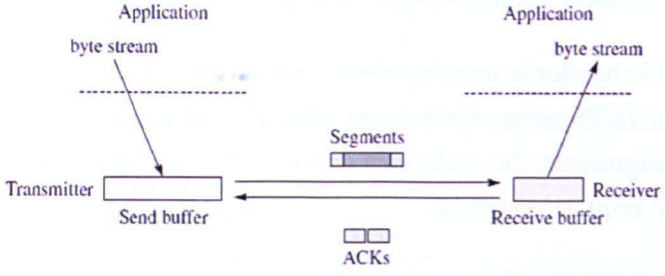


Fig. 2.3 TCP high-level view. Extracted from [1].

The TCP segments sequence numbers correspond to the numbers of the first byte in the segment. If a received segment is error free and not duplicated, it is stored in the receive buffer, even if it is out of order in the sequence. For every received segment containing byte corresponding to R_{next}, the receiver window is moved to the next byte which is not received and an acknowledgment is sent to the sender that confirms that bytes up to $R_{next} - 1$ are correctly received. Hence, the sender updates S_{last} to R_{next}, thus the sender window is moved forward. If a received segment is lost, then it is retransmitted. The retransmission procedure is based on setting timers for every segment. Retransmissions are performed only when the timer expires and none of the bytes in the segment are acknowledged. TCP uses an adaptive technique for setting the retransmission timeout value due to the variable delays in the network layer.

The TCP protocol provides flow control through an advertised window field in the segment header. The ACK packets that are send from the receiver to the sender contain the advertised window size, hence the sender is informed about the state of the receive buffer. The advertised window size is given by:

$$W_A = W_R - (R_{new} - R_{last}). \tag{2.1}$$

where W_R is the maximum number of bytes that can be buffered at any given time. The sender keeps the number of outstanding bytes below the advertised window size ($S_{recent} - S_{last} < W_A$) in order to prevent buffer overflows at the receiver.

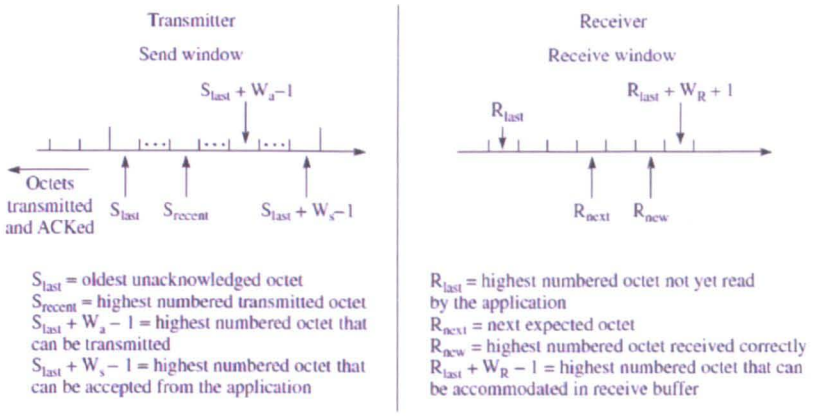


Fig. 2.4 TCP sliding window at the sender and at the receiver. Extracted from [1].

The basic implementation of the TCP protocol has been upgraded and nowadays the TCP protocol has many flavors, including TCP Reno [22], TCP New Reno [23], TCP Tahoe [24], TCP SACK [25], TCP Vegas [26]. There is an extensive literature on TCP, where analytical models of different TCP flavors are compared [27] and simulation based comparisons are presented [28]. The most widely implemented version is the TCP Reno, which contains specific algorithms for slow-start, congestion avoidance, fast retransmit and fast recovery.

2.1.3 Video signals, compression and quality

2.1.3.1 Video signals

Video is a representation of real-world visual scene which is composed of multiple objects with different characteristics, including shape, depth, texture, and illumination. In order to produce the video signal, the scene is sampled spatially, thus capturing the spatial characteristics like texture variation, number and shape of objects and colour, and temporally, thus capturing the temporal characteristics like object motion, changes in illumination, and camera movement [29, 30].

The spatial sampling of the scene results in a representation of the scene on a rectangular plane (frame), whereas the temporal sampling results in a series of still frames sampled at regular intervals in time. The visual quality of the video is affected by the number of sampling points (pixels) obtained from the spatial sampling which defines the spatial resolution of the video and by the number of frames obtained from the temporal sampling which defines the frame rate. There are two sampling approaches: progressive sampling where the video signal is sampled as series of complete frames and interlaced sampling where only half of the data in a frame is sampled at each spatial and temporal sampling interval. The odd or

even number of lines within one frame are labeled as fields. The progressive sampling results in a series of frames, whereas the interlaced sampling results in a series of fields. Interlaced sampling provides sending of twice as many fields compared to frames by using the same data rate which gives an appearance of a smoother motion.

Every pixel in the frames represents a particular colour information related to the captured scene. The colour information in a pixel is expressed as three numbers associated to the brightness (luminance or luma) and colour which form colour space. An example of colour space is *RGB* where the three numbers indicate the relative proportions of Red, Green and Blue. Any colour can be obtained by using a proportional combination of the primary colours. Further, in the *RGB* space, the three colours are equally important and stored at the same resolution. However, the Human Visual System (HVS) is less sensitive to colour than to luminance, therefore it is more efficient to use higher resolution for the luma and lower resolution for the colour. This approach is adopted for the YC_rC_r space which is commonly referred as *YUV* colour space. The luma component Y is calculated as a weighted average of the R, G and B, whereas each colour difference component (C_b, C_r, C_r) is calculated as the difference between R, G, B, and the luma component. A complete description of the frames is obtained by the luma component and the three chroma components. However, since the sum of the chroma components is always constant, only two components are sufficient and required for transmission. In addition, the *YUV* space provides the additional benefit of using different resolution for the components when compared to *RGB*. The different resolutions used for the *YUV* colour space are expressed as sampling patterns, i.e., the sampling pattern 4 : 4 : 4 indicates that the luma and chorma components have the same resolution and a sample of the three components is associated to every pixel in the frames. Sampling patter 4 : 2 : 2 indicates that for every four luma samples there are two samples for each of the two chroma components in the horizontal direction of the frame (the number of luma and chroma samples in the vertical direction of the frames is the same), whereas the sampling pattern 4 : 2 : 0 indicates that for every four luma samples there are two samples for each of the two chroma components in both the horizontal and vertical directions.

The number of pixels in the horizontal and vertical direction determines the spatial resolution of the frame, whereas the number of frames captured every second determines the video frame rate. A frame can be divided into units called macroblocks that are composed of a certain number of pixels. Macroblocks are defined as $M \times N$, where M, N represent the number of pixels in the horizontal and vertical directions, respectively. If many changes occur within few pixels in the horizontal and vertical direction then that image has high spatial complexity (high spatial information), whereas if many changes occur between adjacent images then the video has high temporal complexity (high temporal information).

2.1.3.2 Video compression principles

Video contains large amount of information, hence it is not feasible to store due to the memory constrains of the devices and to transmit over finite channel capacity without initially compressing the video signals. Compression refers to removing the redundancy of the information-carrying video signal and there are two general approaches to achieve this: lossless and lossy compression. In a lossless approach, only statistical redundancy is removed from the signal, thus the original signal can be perfectly reconstructed. However, lossless approaches provide modest compression of the video signal. In contrast, lossy approaches provide greater compression at the cost of not having identical decoded and original signal at the receiver side.

Video compression algorithms are based on removing redundancy in spatial, temporal and/or frequency domain. A simplest approach to compress the video signal is to compress each frame in the video sequence independently by using an image compression algorithm, i.e., by using only the horizontal and vertical direction. This approach uses the similarity between adjacent pixels in the frame (spatial redundancy) and it is called intra-frame compression. However the compression rate in this case is low and hence not suitable for modern video services. Improved results are obtained by using the similarity between adjacent frames (temporal redundancy) in the video sequence. This approach is called inter-frame compression. Further, using the higher sensitivity of the HVS to lower frequency components, additionally improves compression.

These fundamental principles are the basis of video encoding standards like the ones from Moving Picture Experts Group (MPEG) [31] and modern video services like video telephony, video conferencing, DVB-T, and a large number of multimedia services, including HTTP adaptive video streaming over Internet. In this dissertation, the video encoding is based on the H.264/AVC standard [32].

2.1.3.3 Video quality

Video quality is a characteristic of the video signal that represents a quantitative measure of the perceived video degradation due to video compression or transmission [33]. Evaluation of the video quality is performed to capture the quality of a set of video sequences under study. These sequences can be evaluated subjectively (by asking users for their rating) or objectively (by mathematical models).

Subjective evaluation of video quality depends on the clarity and the motion smoothness of the visual scene. In addition, the viewing environment, the subject's state of mind and its interaction with the visual scene, the visual attention and the recency (the opinion of a

sequence is more heavily influenced by the recently viewed parts of the sequence) [34] affect the perceived video quality. The subject's perception of video quality is expressed through the Mean Opinion Score (MOS) measure which can assume different values depending on the adopted scale. Different test procedures for subjective quality evaluation are defined in recommendations from ITU [35, 36]. In the dissertation, the five-grade MOS scale is mainly used.

The complexity and cost of subjective quality evaluation is mitigated by using objective quality evaluation which is based on mathematical models that capture the video quality. A main goal of the objective quality evaluation is to develop a metric that can automatically predict the video quality [37, 38]. Depending on the availability of the undistorted reference video, the objective video quality metrics are categorized as full-reference (the reference video is fully available), reduced-reference (parts and features of the reference video are available) and no-reference (the reference video is not available). The full-reference metrics like PSNR and Mean Squared Error (MSE) are widely used due to their simplicity [39]. A survey of objective quality metrics for video is presented in [40]. However, in many practical systems including the DASH systems considered in the dissertation it is not possible to have a fully available reference video. Therefore, additional metrics tailored to the functionality of the system are required.

Subjective evaluation is the most reliable way of assessing the video quality through the MOS. Therefore, the objective evaluation based on a quality metric should provide a high correlation with the MOS obtained from the subjective evaluation.

2.2 Adaptive video streaming: state of the art

The state of the art initially presents a brief protocol viewpoint on traditional, progressive and adaptive video streaming with more detailed focus on adaptive video streaming. Further, an overview of the emerging MPEG-DASH standard is presented. Deployments and ongoing research related to the standard specifications are gaining momentum in recent years, thus DASH is becoming the de-facto platform for adaptive HTTP video streaming over the Internet. In addition to these, several relevant areas related to DASH are highlighted and notable works in the literature are discussed related to DASH video encoding approaches, adaptation strategies, QoS and QoE for DASH and navigation experience.

2.2.1 Video streaming: protocol viewpoint

The traditional streaming approach uses stateful application layer protocols, i.e., Real-time Transport Protocol (RTP) and RTP Control Protocol (RTCP), or Real Time Streaming Protocol (RTSP) [41, 42]. The RTP/RTCP protocols use the User Datagram Protocol (UDP) connectionless transport service, whereas the RTSP can use both the TCP connection-oriented and the UDP connectionless transport services. The RTP protocol does not provide timely in-order delivery and Quality of Service (QoS) guarantee, hence it must be used with RTSP or different session description/signaling protocols, i.e., H.323 [43] and Session Initiation Protocol (SIP) [44] that would provide QoS and the required control information for synchronization. The RTCP is used for establishing and controlling the media session between the client and the server. The actual media data transmission is based on RTP delivery mechanisms. Regardless of the utilized protocols, the traditional streaming approach provides true streaming since the video arrives in real time and is displayed immediately (e.g., 2 minutes video takes 2 minutes to deliver). The video is sent as a series of packets and the main requirement is to provide the client with larger bandwidth compared to the date rate of the video stream. This streaming approach was introduced to emulate the broadcast TV, satellite TV and CATV system. The main application of this approach is for live streaming.

The progressive streaming approach uses stateless application layer protocol, i.e., HTTP on top of TCP. The video is progressively downloaded into the client's buffer and when there is sufficient buffered data, the player starts to play the video. When playing starts, download still continues. If the rate of the playout is larger than the download rate, the playout is stopped until sufficient data is buffered. In this approach, there is no bit-rate adaptation and every client is equal in terms of the available bandwidth utilization. This approach does not provide support for live sources and most common application is watching a movie preview.

The adaptive streaming approach uses stateless application layer protocol, i.e., HTTP on top of TCP. This approach differs from the progressive streaming since adaptive streaming assumes user detection of the available bandwidth and adjusting the quality of the streamed video over time. The video is encoded at multiple bit rates and is deployed within Content Delivery Network (CDN) in order to improve scalability [45]. A CDN is an interconnected system of cache servers that use geographical proximity as a criterion for delivering web content. CDNs significantly improve network performance and provide reliability for services by distribution of the content to cache servers near to the users, which is particularly useful for adaptive HTTP video streaming in order to provide the best possible quality in the variable network. A high-level view of an adaptive video streaming model is depicted in Figure 2.5.

Adaptive video streaming is based on a client-server communication, i.e., the HTTP server stores the video content and the client sends requests to obtain the content. At the

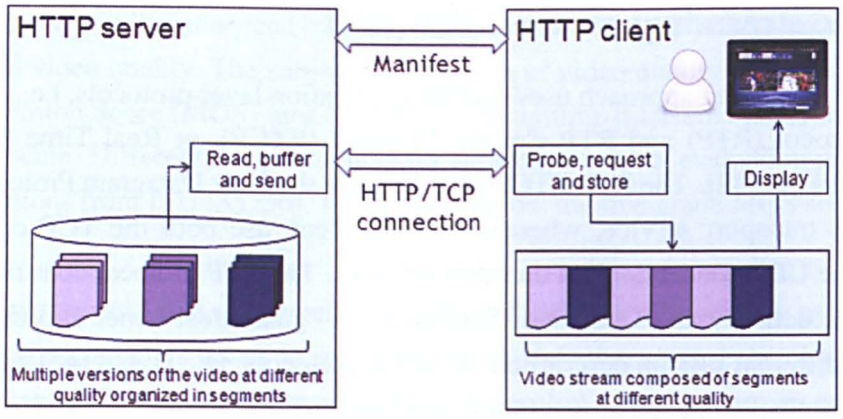


Fig. 2.5 High-level view of adaptive video streaming over HTTP.

server side, the video content is cut into short segments in order to mitigate past/future segment dependencies. These segments are encoded at variable rates in the desired format. The client learns about the video content availability/location through metadata information, i.e., manifest file, which is delivered prior to the streaming process. After receiving the manifest file, the client initiates and guides the streaming process by obtaining knowledge about the network condition, sending HTTP requests for particular segments, and storing the retrieved video segments. The sending of the requests is based on a prescribed adaptation logic at the client, which can consider different aspects, including the video content quality and availability, the network condition and the device capabilities. When a request for a segment is received, the server reads, buffers, and sends the segment in the network towards the client. Hence, the adaptive streaming model enables content/link/device aware streaming, which provides the highest possible quality of the displayed video at the client.

Several proprietary solutions, including Apple's HTTP Live Streaming (HLS), Microsoft's Live Smooth Streaming (LSS), and Adobe's HTTP Dynamic Streaming (HDS) were initially deployed for delivering adaptive streaming to the users. However, these solutions are platform dependent and do not provide interoperability support across different platforms, servers and devices. Therefore, there is a growing interest in the industry and academia regarding the specifications in the Dynamic Adaptive Streaming over HTTP (DASH) standard from the Moving Picture Experts Group (MPEG) group. The following subsection presents an overview of the standard.

2.2.2 MPEG-DASH overview

The MPEG-DASH standard [46] is a promising prospect solution that aims to provide a general framework for interoperable integrated ecosystem for adaptive video streaming in a heterogeneous environment of devices, networks, and servers such as the Internet network.

The MPEG-DASH standard adopts as baseline the specifications introduced in the 3GPP Release 9 PSS standard [47] and the Open IPTV Forum (OIPF) specifications [48]. MPEG-DASH provides definitions for a Media Presentation that is comprised of two parts: Media Presentation Description (MPD) and media segments. The first part defines the MPD, which is a manifest file in XML format used by the client to learn about content availability, location and additional content characteristics. The second part defines the content format in terms of media segments as extensions to 3GP and MP4 media file formats. Segments are independent units of media data composed of encoded video frames with a unique URL, an index, and explicit or implicit start time and size, that corresponds to the time duration of the segment playout. In DASH, the client selects and downloads segments from several media representations according to an implemented adaptation logic, where the media representations provide different quality of the same video content. Hence, the media stream played out at the client is composed of consecutive media segments that are downloaded from a server using the HTTP/TCP protocols. A general system model for MPEG-DASH client server is given in Figure 2.6.

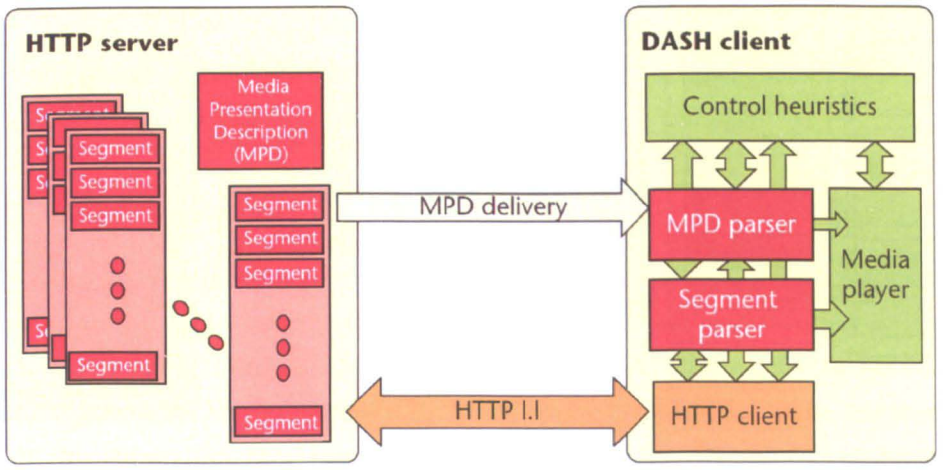


Fig. 2.6 MPEG-DASH client-server model. Extracted from [2].

A key DASH feature is the shared common timeline between the media representations; each segment in the media representation has a particular presentation time, which is mapped to the global common timeline, hence synchronization and seamless switching between the representations is enabled. Therefore, the quality of the media stream varies in time as

consecutive segments from different media representations are played out at the client. The HTTP/TCP protocol stack used for the video streaming enables firewall transparency and utilization of the current HTTP infrastructure (servers, proxies, caches, etc.) [2].

2.2.2.1 Media Presentation Description

The MPD file format consists of one or more periods. A period is a program interval along the temporal axis which has start time, duration, and contains one or multiple adaptation sets. The adaptation set provides information about one or more media components and its encoded alternatives (e.g., same media component but different bitrates of the video/audio component). Adaptation sets include multiple representations. A representation contains alternatives of the same multimedia content, but encoded with different bitrate or resolution, which are organized in multiple segments. The segments contain the actual media information and are located on specific URLs in the network. The hierarchical MPD model with the elements is presented in Figure 2.7.

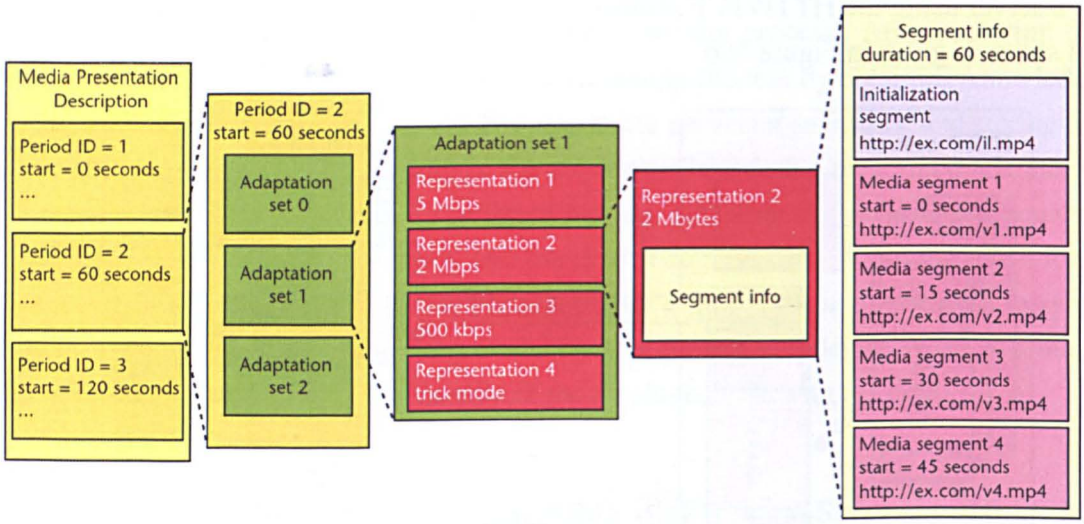


Fig. 2.7 MPD hierarchical model. Extracted from [2].

In the dissertation, one period and one adaptation set composed of several representations encoded at different bitrates and with different resolutions is assumed unless it is otherwise stated. Each representation contains the required number of segments to present the entire video content, where the number of segments depends on the segment size.

2.2.2.2 Segment formats and types

The segment format specifies the syntax and semantics of the resources associated with the URL identified by the MPD file. The HTTP GET request to a resource in MPD is responded with an HTTP response that contains information conforming to a segment format. The standard focuses on segment formats based on MPEG container formats and describes the usage of two segment formats: Media Segments based on the ISO Base Media File Format (ISO/IEC 14496-12) and Media Segments based on the MPEG-2 Transport Stream (TS) (ISO/IEC 13818-1). Both segment formats are defined such that the Media Segment format complies with the respective container formats. The standard defines the following segment types:

- Initialization. Segments containing initialization information for access to different Representation;
- Media. Segments containing encoded media content components;
- Index. Segments primarily containing indexing information for Media Segments;
- Bitstream Switching. Segments containing essential data to switch to the Representation to which it is assigned.

In this dissertation, an initialization segment to start the streaming sessions is used and the resulting stream at the client is obtained by concatenating the requested media segments unless it is otherwise stated.

2.2.2.3 Media Presentation

An MPD and the referenced segments comprise a Media Presentation. A complete Media Presentation is obtained when the MPD and the segments are combined according to the general rules provided in the standard specification. In addition to the general rules, there are specific rules for ISO based media file, MPEG-2 TS, and guidelines for supporting other formats. In this subsection, the general rules are discussed, i.e., the shared common timeline and the segment indexing.

The shared common timeline between the encoded version of different media components, i.e., the representations is essential for switching between different representations. An access unit in each segment in the media component has a particular presentation time, which is mapped to the global common timeline, hence synchronization and seamless switching between the components is enabled. Besides the presentation time, the access unit is defined

with a byte range and presence of representation access positions (IDR frames) in the segment. These are called Stream Access Points (SAPs) and are independently encoded frames, i.e., I frames.

Segment indexing provides binary information in ISO box structure and it is used for hierarchical structuring and efficient access to access units. If the segment index is present in a media segment of one representation of the media component within an adaptation set, then this index should be present in all segments in every representation of the media component within the adaptation set. There are three ways of combining the media segments and the segment index: simple, hierarchical and daisy-chain.

2.2.2.4 Profiles

DASH profiles are defined to enable interoperability and the signaling of the use of features. In each profile, a set of restrictions is imposed on the MPD and the segment formats, or on the delivered content with the segments (codecs, protection formats, segment sizes, etc.). Three DASH profiles are defined on the ISO base media format of the segments, i.e., ISO Base media file format On Demand profile and ISO Base medial file format live profile, which are subsets of ISO Base media file format main profile. Additional two profiles are defined for MPEG-2 TS based segments, i.e., the MPEG-2 TS simple profile, which is a subset of the MPEG-2 TS main profile. All of these profiles are subset of the full profile.

2.2.3 Relevant areas

The research works in the literature relevant to this dissertation are organized in four groups. In the first group, works that mainly focus on encoding and segmentation of the video content are surveyed. In the second group, works that investigate adaptation approaches and different QoS and QoE parameters are presented. The third group presents works which are mainly focused on QoE, in particular works that propose a metric or model for QoE. Finally, in the fourth group works related to streaming navigation are presented.

2.2.3.1 Content encoding and segmentation

The encoding and segmentation of the content for adaptive HTTP video streaming is controlled by two main parameters: the segment size and the GOP size. The former determines the time duration of the segment as defined in the MPEG-DASH standard, and the latter determines the time duration of the GOP. In most of the works in the literature, these parameters are equal, however in the dissertation the benefits and drawbacks of using unequal segment size and GOP size are discussed. In DASH terminology, if the GOP size is smaller

than the segment size, the considered GOP corresponds to DASH subsegment or fragment. In all the works in the dissertation, the same structure of the GOP is used, i.e., the IPPP..P structure as defined in the H.264/AVC video coding standard [32]. Recently, few works in the literature consider encoding according to the High Efficiency Video Coding (HEVC) standard [49] for DASH [50–52], however in the dissertation only H.264/AVC video coding is considered.

The authors in [53] propose an approach for optimal video encoding, which adopts content-based segmentation that provides a balance between the delay requirements in the network and the rate-distortion performance. The segmentation is performed by considering fixed segment lengths, segments derived from scenecuts, and a novel strategy that selects optimal candidates among scene changes with an additional goal of producing close to constant segment length.

An approach for generating content (i.e., HTTP datasets) according to the MPEG-Dynamic Adaptive Streaming over HTTP (DASH) standard is presented in [54], which elaborates on the design of the DASHEncoder tool [12]. In [54], a fixed GOP size equal to the segment size in the H.264/AVC encoder is used and it is shown that the PSNR increases when the segment size is increased. This is due to the smaller number of I frames in the entire video and the fixed bitrate set during the encoding.

The work in [55] presents a method for selecting the segment duration, which estimates the segment duration based on relevant TCP parameters (number of rounds for the congestion window, number of acknowledged packets) and network conditions (packet loss probability, round trip time). The method estimates the minimum duration in which the average rate can represent the end-to-end capacity, and the segment size should be selected to be equal to this duration.

Another study in [56] investigates the impact of the access technology (DSL, WLAN, UMTS) on the HTTP streaming performances based on measurements, showing a frequency of late segments up to 7.8% for the cellular case and 17.72 % for the LAN cases. Four segment sizes equal to 1s, 2s, 5s, and 10s are used, and the buffer occupancy level for each session is reported. Similar approach is used in [57] where OPNET modeler is used to implement a real-time test platform for mobile adaptive HTTP video streaming with different segment sizes (4s, 8s, and 12s). Slightly different approach is presented in [58] where the QoE unfairness for multiple DASH users is investigated. The unfairness occurs due to the TCP congestion avoidance and the adaptation of the client and it results in some users receiving high-quality video, whereas others receiving low-quality video. This work shows contrasting observations compared to the works mentioned above in terms of the segment size and this is due to the considered multi-user setup. Another reason is the adopted QoE

metric which provides the quality by considering only the quality profiles of the received segments (representations) and it does not consider the rebuffering events, the amount of switches and the QoE level distance between the quality profiles.

In the end, Muller et al. have tested in their works the implementation of both the H.264/Advanced Video Coding (AVC) and the H.264/SVC standard over MPEG-DASH [59, 60]. These results show that DASH performs better compared to commercial adaptive streaming solutions for both AVC and SVC implementations.

In the dissertation, the impact of using different segment sizes and different GOP sizes with respect to the adopted segment size is investigated in specific system models. The encoding and segmentation are controlled via the segment size and GOP size parameters for which different parameters are investigated like the PSNR of the streamed frames, the average video throughput, the rebuffering frequency, and the MOS of the streaming session. The DASHEncoder is used for encoding and segmentation. Similar trends as in the literature regarding the impact of encoding and segmentation on the average PSNR of the frames are observed: smaller number of I frames in the entire video and fixed bitrate setting during the encoding results in PSNR reduction. In large parts of the dissertation, the segment size and the GOP size are equal, however in some cases using smaller GOP compared to the segment size might be preferable. Few insights along these lines are presented via results in the dissertation showing that despite the encoding benefits for the larger segments (in terms of PSNR), smaller segments are preferable to facilitate dynamic adaptation in the highly variable access technologies. This is additionally showed by considering the actual MOS of the DASH video streaming and a broader range of performance metrics of the streaming system compared to the aforementioned works.

2.2.3.2 Adaptation and QoS

The MPEG-DASH standard does not prescribe a precise logic for adapting/switching between the available representations. Since this part of the system is not standardized, there is extensive research and tests of adaptation approaches in different scenarios. As a measure of the effectiveness of the adaptation approach several QoS and QoE parameters are observed, including the throughput, end-to-end delay, PSNR, number of switches, rebuffering events, etc.

An extensive study on understanding the video quality impact on user engagement is presented in [61], where a dataset of different content types and services is used, and the following parameters at the client side are observed: join time, buffering ratio, average bitrate, rendering quality and rate of buffering events. Furthermore, the effect of these parameters on

the QoE is investigated and the analysis shows that the buffering ratio has the largest impact on the QoE regardless of the service and the content type.

Four different segment request strategies are proposed in [62] and metrics related to packet loss, sessions goodput, smoothness of playback and end-to-end delay are observed, whereas the authors in [63] analyze the delay in a DASH live streaming scenario, which occurs due to the preparation of segments, fetching of the segments, HTTP download time and buffering at the client.

An approach for optimizing HTTP adaptive video streaming over wireless access networks is presented in [64]. An Intelligent Bitrate Switching-based Adaptive Video Streaming (ISAVS) algorithm that minimizes the replay interruptions and selects best video bitrate is proposed.

A parallel adaptive HTTP streaming method where the client can request multiple segments in parallel is presented in [65]. This method enables adaptation of the selected media bitrates while previously requested segments are received, which improves the performance compared to the traditional streaming methods. The following metrics are observed: frequency of interruption and media playback quality, which is represented as an average of the representation levels, expressed in kbps.

The client-oriented adaptation algorithm proposed in [66] is implemented on a prototype and tested in real-world scenarios. This algorithm considers the information about the dynamics of the available throughput in the past and the buffer level in order to produce the selected representation level and the minimum buffer level when the download must start as output arguments.

A buffer-based adaptation method that performs estimation of the buffer in the future is proposed in [67]. The main aim of the method is to reduce the negative impact caused by bitrate variations and to prevent buffer underflow. Based on future estimates, the method selects a sequence of bitrates (paths), and the tests conducted over different bandwidths show improvements compared to common adaptation approaches regarding the average bitrate and the number of switches between the representations.

The works [68] and [69] elaborate adaptation approaches for large scale setups. The authors in [68] propose a probe-and-adapt (PANDA) adaptation algorithm that outperforms the conventional adaptation approaches in terms of best stability-responsiveness tradeoff, whereas the rate adaptation approach in [69] is designed for multiple servers, where multiple fragments from different servers are grouped in a block, which is then downloaded by using parallel connections.

The FESTIVE [70] algorithm is compared to several commercial players and provides better efficiency, fairness and stability for bottleneck connections shared by multiple users.

The average video bitrate, the bandwidth utilization and the average buffered time are considered.

Two content-aware adaptation schemes for optimizing the QoE for DASH are presented in [71]: Content-aware Probe and Adapt (C-PANDA) and Content-aware Dynamic Programming (C-DP). Similar to the previous adaptation approaches, the available bandwidth and the existing buffer capacity are considered, however C-PANDA and C-DP additionally use the video content type. The video content type is classified based on extracted content features, including scene boundary detection and motion intensity classification at the level of a segment. These approaches outperform the PANDA approach in terms of buffer level, selected representation and MOS as captured in the model proposed in [72].

A reinforcement learning adaptation is proposed and evaluated in [73] which uses the average segment quality, the switching and the rebuffering as input for the quality estimation. The quality is evaluated with network-based simulation and it outperforms the traditional clients, however no content-related parameters are considered in the adaptation approach.

ELASTIC [74] is a client-side controller that mitigates the on-off pattern which typically occurs in a bottleneck connection shared by multiple clients that use typical client-side adaptations as elaborated in previous works. The ELASTIC adaptation provides full utilization of the bottleneck and fair sharing of the connection. It outperforms the FESTIVE and PANDA algorithms in terms of the fairness and provides better performance with respect to the bitrate and the switching.

The work presented in [75] proposes a QoE-enhanced adaptation algorithm over DASH (QAAD) that preserves minimal buffer length, avoids rebuffering and minimizes the changes in video quality. The algorithm is compared to QDASH [75] and provides better performances.

Another adaptation algorithm is presented in [76], which is one of the first works in the area that considers QoE in the context of the adaptation approach. A quality based optimization is presented in the algorithm [77] which uses stochastic dynamic programming to find the optimal request policies, thus guaranteeing a certain level of QoE expressed as in [78]. Further, this algorithm is compared to another early work in the area, referenced above in [66] and shows better results.

In [79] a study on the required buffer size for prescribed video quality by targeting memory constrained applications is presented. This approach considers video streaming over TCP and provides an analytical model for the buffer size as a function of the TCP transmission parameters, network characteristics and probability of buffer underrun at the playback.

The work in [80] focuses on the introduction of a video adaptation proxy at the edge between Internet and wireless core, in order to improve the DASH QoE in cellular broadband access networks. The proxy performs global optimization over multiple DASH flows by splitting the original TCP connection to a wired and multiple wireless TCP connection towards the client (hence, increasing average throughput), deploying video quality aware dynamic prioritization (low rate streams have high priority to preserve minimal QoE and fairness is introduced), and an adaptive controller is proposed in order to minimize a cost function which depends on the video distortion, bit rate variations and playback jitter.

The adaptation algorithms in the dissertation are either throughput aware or throughput and buffer aware, however most of the works focus on selecting representation for the following segment when different encoding and segmentation parameters are used, rather than investigating the effect of using different adaptation strategies at the client. Parallel connections at the client are not considered, however similar QoS and QoE parameters are observed and in the dissertation additional parameters are considered compared to the aforementioned works. In addition, proxy and network-based control is not considered in the dissertation and all the presented works consider adaptation control which is strictly deployed at the client side.

2.2.3.3 Quality of Experience

As a result of the encoding and segmentation of the video sequence in multiple bit rates (expressed in bps) by using predefined GOP sizes (expressed in seconds) and segmented in independent units (segments) by using predefined segment sizes (expressed in seconds). The bit rates define the DASH representations, and during the video streaming the client uses an adaptation strategy to select a segment from the DASH representations. Hence, segments with variable quality are played out at the client side and the quality as perceived by the users varies. QoE models/metrics capture the quality as perceived by the users by including different system parameters.

The QoE model derived in [81] considers the initial buffering time, the mean duration of re-buffering events, and the re-buffering frequency. In this model, the most significant input is the frequency of re-buffering, outlined as the main factor for the variations of the users QoE. Therefore, throughout the dissertation, the rebuffering frequency impact on MOS is constantly considered, however models and methods proposed in the dissertation also consider the average bitrate of the streaming, initial buffering and switching.

The authors in [82] propose a no-reference QoE model based on Random Neural Networks (RNN) for two main problems regarding adaptive video streaming: playout inter-

ruptions and video quality variations due to lossy compression at different video quality levels.

A subjective quality study of adaptive streaming is presented in [83], where the authors investigate the impact of the adaptation techniques on the visual quality for monoscopic and stereoscopic video. Four quality levels corresponding to 2, 4, 6, and 8 Mbps, and adaptations based on increasing quality, decreasing quality, and quality oscillations are considered.

An approach that considers the switching impact on the QoE is elaborated in [84], where the estimation of the MOS is performed based on buffer underflow metrics (rebuffering frequency and the average duration of rebuffering), frequency and amplitude of quality switches (depending on the implemented adaptation and the available DASH representations), and objective video quality related parameters (bitrate, frame rate, quantization parameter). This work outlines the importance of considering content related parameters when modelling the QoE; where it is highlighted that the impact of the content encoding parameters on the QoE is significant in intervals measured in seconds, whereas the remaining parameters have an impact on broader interval ranges.

The extensive work presented in [85] provides a modeling approach of the time varying subjective quality of adaptive HTTP video streaming. The model targets real time MOS evaluation by considering several factors related to the impact of quality fluctuations on the MOS (sensitivity to switches, recency, etc.). The conclusions indicate that the users' response is more sensitive in the low video quality regions, and the recency effect of quality fluctuations spans across interval of 15s.

Few works on TCP video streaming services propose quality metrics and methodologies for quality evaluation at the end of the streaming session [86, 87]. The authors in [86] propose a quality monitoring mechanism in a dedicated measurement point in the network that considers the number and duration of the stalling (rebuffering) events, and the time elapsed since the last stall. This network-based solution merges these three parameters into a single parameter called Negative Impact (NI) that is linearly dependent on the MOS. In the end, a general discussion on applying the proposed mechanisms and the quality metric in the DASH context is presented; however evaluations in a general DASH setup and DASH related parameters that impact the quality (PSNR of the streamed frames for different representations during quality switching, frame rate, etc.) are not investigated.

In [87] a Video Streaming Quality Metric (VsQM) metric is proposed to assess the TCP streaming service by considering the number, duration and the temporal location of the pauses. The pauses occur at locations characterized with different spatial and temporal information (three sequences are used) and several application scenarios are considered. The same authors extended the previous work in [88] by proposing a no-reference video quality metric for

DASH based on two parameters: pauses and changes in the video resolution (temporal resolution). In addition, several weighting factors are introduced and an exponential function is used to map the metric values in the 5-point MOS scale.

The works in [78, 89] investigate different adaptation approaches based on the metric derived in [78]. This metric considers the average segment quality penalized by the frequency and length of the video freezes (rebuffering) and the quality switching, summarized in three factors that further depend on the number of segments, the segment bitrate, the average bitrate between the switches and the minimum and maximum representations in the setup.

A perceptual quality assessment is presented in [90], using the absolute category scale method for adaptive HTTP video streaming under predefined adaptation logic. The findings show that the MOS scores are similar when the adaptation aims at keeping the transmission throughput constant and when the aim is to keep the quality index close to the previously used quality index.

An approach to model the user experience at the end of the DASH session are presented in [91] where three main factors: initial delay, stall and bitrate, and multiple dimensions for each factor are considered, including the total stall duration, number of stalls, timing, average bitrate, number of switches and average switch magnitude. Therefore, this work is the first notable work in the literature to study both the spatial and temporal quality of the video streaming through deriving separate impairment functions. The proposed metric is derived based on real mobile network condition for short videos and the impairment functions are derived for each of the influencing factors; however, a full metric that considers all factors is not proposed in the work.

Different models for estimating the QoE are presented in [92], however these models are tested only for DASH over LTE network. Two analytical methods [93] and analytical model [94] for DASH are discussed in these works. The methods are based on linear approximation and power approximation and consider the average video quality and the standard deviation of the video quality, whereas the analytical model presented is one of the first works in the literature that presents analytical framework to compute the starvation probability of the playout buffer, the continuous playback time and the mean video quality under the assumption that the adaptation logic is known in advance. The models can be parametrized to investigate the impact of various parameters on quality, for example the impact of buffer-aware switching, patience of the users and receiver side flow control.

In the dissertation, a novel method and model for QoE are proposed as an extension of a conducted QoS study. The method expresses the MOS for DASH by assuming that model for MOS for constant bitrate video streaming is known in advance. In the proposed method and model, the average bitrate, the switching variance, and the rebuffering frequency are used

as key parameters. However, the proposed method and model are derived under a specific setup. Therefore, a more general setup is considered in which conditions that occur typically for DASH streaming are used, and a QoE metric is derived. This metric captures a broader range of parameters quality related and rebuffering related parameters and it additionally considers content-based parameters. As such, it is compared with state of the art metric [81] and provides higher correlation with the actual MOS. The method, model and the metric are obtained by using analytical models and are supported by results from subjective results.

2.2.3.4 Navigation experience

The authors in [95] propose a segment-based proxy caching algorithm that supports interactive functionalities for video-on-demand (VoD) services. Streaming users tend to pay most attention to the initial delay and accept shift of the requested starting point during seeking. This observation is investigated via subjective VoD tests, and the algorithm is tailored to maximize the user satisfaction by trading off between the initial delay and the deviation of the requested starting point.

The QoE assessment of web-based audio/video presented in [96] focuses on a comparison between adaptive streaming and progressive download. The subjective tests are conducted under various network conditions and load traffic, where users perform seeking to pre-specified scenes during the video streaming. It is outlined that adaptive bitrate streaming is not necessarily effective for QoE enhancement, and the main effectiveness of the scheme depends on the usage of the system and the network conditions.

A system for interactive editing of video clips, ChunkStream is proposed in [97]. The clips are represented using linked lists and search trees, which allows clients to retrieve required portions of the video in forms of fixed size chunks. The authors compare the proposed system with the HTTP Live Streaming technology, where 1s segment size is used, and the experiments are conducted over a wireless network with limited 3G-like speeds of 2 Mbps. The ChunkStream system uses less bandwidth and provides better support for video editing. In addition, the seek time with the proposed solution is significantly decreased.

Another work in [98] presents an approach for accurate and low-delay seeking across mash-ups, which are composed of highly compressed video sequences.

A novel system that supports playout of personalized multi-path non-linear video is presented in [99]. The proposed system is based on multiple nonlinear video sequences of the underlying (linear) video to be played in any personalized order. This work combines non-linear video streaming and DASH.

Another work from the same authors presented in [100] focuses on non-linear streaming where uses select paths through the video and the idea is to mitigate the playback interruptions

by careful buffer management and prefetching content in advance by using parallel TCP connections.

In the dissertation, the term navigation experience is introduced to capture the users' perception of quality for non-linear video streaming guided by the users, similar to the conventional usage of QoE in the case of linear video streaming as in conventional DASH systems. In particular, a novel analytical model that shows the dependency of the expected navigation delay during non-linear video streaming is proposed. The delay is expressed by using several parameters related to the DASH system.

2.3 Chapter summary and conclusions

The chapter on literature overview presents a section on background information and a section on state of the art in adaptive video streaming. In the background section, the important aspects related to protocols and layer services and the video coding fundamentals and quality aspects are presented. The section on the state of the art discusses video streaming models from protocol viewpoint, presents an overview of the MPEG-DASH standard, surveys notable works in the literature from the areas relevant to the research, and overviews the commercial and open-source solutions for DASH deployment. The background information and the state of the art sections summarize the required knowledge and present the tools used throughout the research. These are useful in particular at the stages of designing the system models and interpreting the obtained results and conclusions relevant to adaptive video streaming systems. As a result of the work presented in this chapter, the theoretical background knowledge is summarized and state of the art in adaptive video streaming is presented.

Chapter 3

Case studies

MPEG-DASH enables users to maintain video quality in varying transmission conditions by careful selection of system parameters depending on the considered use case. In this chapter, three specific MPEG-DAS case studies are presented. In the first case study, a novel system model for DASH-based video streaming over LTE networks is proposed. The model is used to emulate end-to-end streaming from a server located in a CDN to multiple LTE users, and the study focuses on how the scheduling affects the video streaming quality depending on the number of users served at the eNodeB. The second case study focuses on the segment size impact on QoS for DASH live streaming services. Based on the conducted measurements, the study shows that the end-to-end QoS is affected by the segment size selection. The third case study focuses on using DASH for medical video streaming since the HTTP/TCP provides lossless packet transmissions and the video quality is investigated for two adaptation approaches and different segment sizes. Different system modeling approaches and evaluation methodologies related to DASH are used in these studies, however the outline of the studies is organized in a similar manner. Initially, related works are discussed. Further, the system model for the respective study is described and the obtained results from the system model are presented. In the end of each study, a brief summary is presented.

This chapter is organized as follows. The study on DASH video streaming over LTE is presented in Section 3.1. Section 3.2 presents the study on DASH live video streaming. The study on utilizing DASH for medical videos is presented in Section 3.3, whereas Section 3.4 concludes the chapter.

3.1 DASH-based video streaming over LTE

This study proposes a novel system model for DASH-based video streaming over LTE networks and presents an approach for the evaluation of the streaming from a server located in a CDN to multiple LTE users. In the study, an analytical model and simulations are used to emulate DASH-based end-to-end video delivery and to investigate how the packet scheduling affects the adaptive video streaming quality depending on the number of users in the LTE network. The analytical model is used for the HTTP/TCP transmission in the wired/core link, and the simulation traces for the packet transmission in the the wireless/access link. Further, the core and access link are separately considered by using the Wireless Transmission Control Protocol (WTCP) at the eNodeB, and three scheduling approaches are used for delivering the video packets in the access link. As an output of the proposed model, the streaming quality is captured via:

- Rebuffering probability. This indicator shows the probability that a rebuffering event occurs during the whole duration of the video streaming;
- Average rebuffering duration. This indicator shows the average time duration of the occurred rebuffering event;
- Average video representation. This indicator shows the average video bitrate of the video streaming.

3.1.1 Related Work

The standardization works of the MPEG-DASH standard followed by extensive performance evaluation in several works such as [101], [102], [103], [66] have demonstrated better performance of MPEG-DASH with respect to other streaming standards. The evaluation of the QoE provided with the MPEG-DASH standard has attracted significant research attention. For instance, the works in [104], [105] focused on evaluating the QoE for HTTP adaptive streaming simulated over 3GPP LTE networks. They showed that the rebuffering can be used as a quality evaluation metric for adaptive streaming. Furthermore, in [106] the authors presented their work in optimizing adaptive streaming protocols with simulation results indicating the factors which influence the QoE of streaming applications, and also recommended both subjective and objective quality evaluation methods to evaluate the streaming efficiency.

Recent works have investigated the functionality of the MPEG-DASH standard and presented several implementations of HTTP-based adaptive streaming over different wireless

networks. For instance, Pu et al. [107] designed an architecture for DASH streaming in which a video is placed in multiple servers and their proposed collaborative multi-scale scheduling algorithm (CMSS) is tailored for adaptive video streaming. A more detailed work published in [108] proposes a wireless DASH (wiDASH) proxy, which uses three novel methodologies to obtain an improved overall streaming Quality of Experience (QoE). First, it uses split TCP and parallel TCP to provide one wired TCP as well as multiple wireless TCP connections from the DASH server to the user. Second, it uses video quality aware dynamic prioritization. Third, an adaptive optimal controller is proposed, which controls the multiple input multiple output DASH streams aimed at improving the overall streaming QoE.

In particular, a few works have focused on the implementation of various resource allocation and scheduling strategies over LTE networks. Some of these works are QoE driven. In [109], a QoE driven resource optimization approach for multiple users using DASH over different wireless networks was presented. The presented approach implements re-writing of client HTTP requests at a proxy in the mobile networks without the need for adaptation of the media content. In [110], the QoE driven resource allocation also focused on optimizing the energy consumption in the mobile devices. A QoE-aware rate conservative DASH streaming over mobile networks was presented in [111], which uses utility function to find the optimal stream switching solution in DASH. This function captures the QoE for the users and it is used as an input. Further, in [112], a framework for scheduling HTTP-based adaptive video flows over cellular networks was presented. In [113], a gradient based algorithm chooses an optimal resource allocation by integrating DASH-based media delivery and radio level adaptation in a cross layer manner. In order to provide fair resource allocation for HTTP adaptive streaming over LTE networks, the authors in [114] proposed a solution based on the video content characteristics and the channel condition.

The works in the literature have focused on various aspects, including adaptation strategy evaluations, QoE models based on subjective tests, and applying different LTE resource allocation and scheduling strategies in the context of adaptive video streaming standards. However, to the best of the author's knowledge none of these works provide an analysis on how the trade-off between the packet delay and the packet loss of the scheduling procedure affects the adaptive streaming quality. In this study, these aspects are considered within a system model that combines analytical and simulation approaches for DASH-based end-to-end video delivery from a server located in a CDN to multiple LTE users. Further, the usage of the WTCP protocol enables separate observations for the core and access link, and the implemented packet-based retransmission at the eNodeB provides reliable packet delivery in the wireless link.

3.1.2 System Model

Fig. 3.1 depicts the system model from a topology and protocol viewpoint. The video content, i.e., the DASH dataset is stored at the DASH server as segments, which are encoded at different representation (quality) levels and segmented with fix length T (in seconds) as described in the MPD hierarchical model in the MPEG-DASH standard. A DASH client, i.e., an LTE user, obtains the MPD file (e.g. through email, signaling messages, or during network association) before the streaming starts and learns about the content availability and location. The adaptation at the DASH client is throughput-based, i.e., the client requests segments from a representation that matches the network throughput.

The segments are requested by sending HTTP GET requests from the client to the eNodeB (through the access link), which forwards the requests through the core link to the server. The server responds with the actual video data that corresponds to the requested segment. This video data is sent based on the TCP/IP protocol stack through the core link (exchange of IP packets), and received at the eNodeB, which stores the packets and schedules transmissions to the corresponding users. Depending on the scheduling, the video packets are delivered to the users with a certain packet delay and packet loss probability. In addition, a packet-based retransmission mechanism is used to mitigate the losses in the wireless link since DASH streaming is based on lossless packet delivery. The transmission of the segments from a particular video representation to the LTE clients depends on the packet delay, the packet loss ratio and the used scheduling procedure.

The eNodeB is used as a relay that splits the connection in two parts by using the WTCP protocol, from the server to the eNodeB (core link) and from the eNodeB to the LTE users (access link). In the core link an analytical TCP model for the packet delivery is used, whereas for the access link simulation traces are used. Three different scheduling approaches are considered at the eNodeB, resulting in different packet delays and packet losses. Hence, in the proposed system model, end-to-end DASH-based video streaming is emulated, and the effect of the scheduling at the eNodeB on the adaptive streaming quality is evaluated through the rebuffering probability, the rebuffering duration and the selected video representation for different number of users.

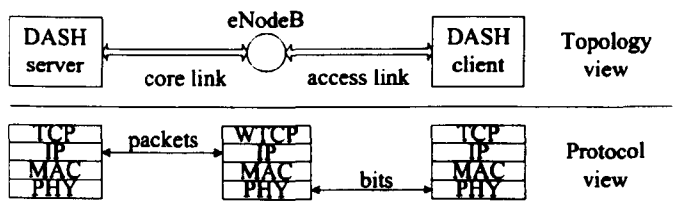


Fig. 3.1 System model: topology and protocol view.

3.1.2.1 Server & clients

The DASH dataset is encoded by using the implementation of the H.264/AVC standard in JM reference software [13] at four different representations, expressed in kbps {100, 200, 300, 400}, and the segmentation is performed with fixed segment size ($T = 1s$). During encoding the IPPPP... sequence format is chosen wherein the Intra (I) frame repeats at the beginning of every segment with video frame rate set at 30 frames per second (fps). Therefore, this is used to generate the DASH manifest with the relevant information regarding the encoding. The first $60s$ of the *Big Buck Bunny* sequence were used for the tests with a resolution of 640×360 pixels. All the clients in the LTE network obtain the same video sequence, however with different quality variations depending on a throughput-based adaptation, i.e., by selecting the highest representation the data rate of which data is smaller than the network capacity. At the time of the study, the bitrate and resolution selection was reasonable for DASH, especially since the intention was to investigate performances for high number of users. Although, current bitrates for DASH are significantly higher, the same procedure presented in the study can be applied for emerging DASH services.

3.1.2.2 eNodeB scheduling

The eNodeB receives the packets for the requested video segments, and serves these packets by using the following scheduling procedures: Proportional Fair (PF), Modified Largest Weighted Delay First (M-LWDF), and Queue-HoL-M-LWDF.

The proportional fair (PF) scheduler [115] assigns the Physical Resource Blocks (PRBs) by taking into account both the experienced channel quality and the estimated user average throughput. This scheduler is suitable to support non-Real Time (NRT) traffic. The goal of this scheduler is to improve the total network throughput and to guarantee fairness among the flows. Hence, the utility metric $W_{i,j}(t)$ in this scheduler is defined as the ratio between the instantaneous data rate of the i-th flow in the j-th sub-channel (i.e., $r_{i,j}(t)$) and the average data rate of the i-th flow (i.e., $\bar{R}_i(t)$). The following equation illustrates the metric used to represent the PF scheduler:

$$W_{i,j}(t) = \frac{r_{i,j}(t)}{\bar{R}_i(t)}, \tag{3.1}$$

where $r_{i,j}(t)$ is computed based on the Adaptive Modulation Coding (AMC) module, which is selected according to the Channel Quality Information (CQI) feedback received from the User Equipment (UE). This feedback represents the channel quality (i.e., Signal-To-Interference-plus-Noise (SINR)) of the j-th sub-channel associated to the i-th flow. $\bar{R}_i(t)$ is the estimated average data rate.

The M-LWDF scheduler is developed to support multiple Real Time (RT) data users [116]. The scheduler assigns PRBs to different RT flows by considering the properties of the classical PF rule and the Head of Line (HoL) packet delay. On the other hand, the scheduler switches to use the PF rule to assign PRBs for NRT data users. In contrast to the PF scheduler, which is more suitable for NRT flows and fairly distributing the resources in the network, the M-LWDF can support RT and NRT services. In addition, $\mathscr{R}$ is defined as the set of flows with real-time requirement,whereas $\mathscr{N}$ as the set of flows with non-real time requirement; hence the following equation illustrates the metric for the M-LWDF scheduler:

$$W_{i,j}(t) = \begin{cases} \alpha_i D_i(t) \frac{r_{i,j}(t)}{\bar{R}_i(t)}, & \text{if } i \in \mathscr{R} \\ \frac{r_{i,j}(t)}{\bar{R}_i(t)}, & \text{if } i \in \mathscr{N} \end{cases} \tag{3.2}$$

where α_i is the maximum probability that the HoL packet delay $D_i(t)$ (i.e., the first packet to be transmitted in the queue) exceeds the target delay, therefore packets belonging to a RT service will be discarded if they violate the target delay bound while waiting at the Medium Access Control (MAC) buffer. The definitions of $r_{i,j}(t)$ and $\bar{R}_i(t)$ parameters are the same as for the PF scheduler.

The Queue-HoL-M-LWDF scheduler in [117] is a modification of the M-LWDF and Virtual Token Modified Largest Weighted Delay First (VT-M-LWDF) [116] [118]. This scheduler combines the effective components and the core principle of the aforementioned schedulers. It takes into account the queue size of the buffer and the packets delay parameters in the VT-M-LWDF and M-LWDF rules, in order to improve the performance of the proposed scheduler when serving both RT and NRT services. The following equation illustrates the metric used to represent the Queue-HoL-M-LWDF scheduler:

$$W_{i,j}(t) = \alpha_i D_i(t) Q_i(t) \frac{r_{i,j}(t)}{\bar{R}_i(t)}, \forall i \tag{3.3}$$

where $Q_i(t)$ is the queue size of the i-th flow at a particular scheduling epoch (t). The definitions of α_i, $D_i(t)$, $r_{i,j}(t)$, and $\bar{R}_i(t)$ parameters are the same as the ones for the PF and M-LWDF schedulers.

3.1.2.3 Access and core links

The aforementioned scheduling approaches are tested through simulations, and the results for the average packet loss and the average packet delay in the access link are obtained for different video representations depending on the number of LTE users. These simulations are used to model the access link, hence, for a particular set of requested bitrates and depending on the number of users, the video packets are transmitted by considering a predefined average

packet loss and average packet delay from the eNodeB to the users. In addition, a simple retransmission procedure is implemented to enable lossless, reliable and in order packet delivery in the wireless link as provided with the TCP protocol. The used packet size in the simulations of the access link transmission corresponds to the packet size used in the core link, which is set to MSS=1400 bytes.

The video segments which are stored on the DASH server are delivered via the core link as a response to the HTTP-GET requests sent by the client. The segments are delivered by using the HTTP/TCP protocols as packets. The TCP protocol provides reliable transmission of these packets, hence each segment is mapped into a corresponding number of TCP packets. The TCP throughput in the core link is modeled according to [119], and it is expressed (in packets/s), as follows:

$$\sigma(p) = \frac{1}{RTT\sqrt{\frac{2bp}{3}} + T_0 min(1, 3\sqrt{\frac{3bp}{8}})p(1+32p^2)}, \tag{3.4}$$

where p is the packet loss probability, RTT is the server-eNodeB round trip time, T_0 is the waiting time for re-transmission of delayed or lost TCP packet, b is the number of acknowledged packets (TCP implementations usually send one ACK for two received packets, hence $b = 2$), W_{max} is the advertised maximum TCP window size, which represents the number of packets that can be accepted without acknowledging the server. The server uses a congestion avoidance mechanism while incrementing the window size until W_{max}, at which the window size remains fixed until occurrence of triple duplicate or time-out event.

3.1.3 Performance results

In this subsection, the performance results are presented. Further, the subsection provides relevant insights regarding the impact of the used scheduling approach on the adaptive video streaming quality. The quality is captured with the rebuffering probability, the rebuffering duration, and the selected video representation, which are all averaged across the number of users in the system.

Rebuffering occurs when the time needed to download the packets from the server to the user for the requested segment exceeds the playout time of that segment. The additional elapsed time during the rebuffering is used to calculate the rebuffering duration, whereas based on the requested segments, the video representation level is calculated.

Previous studies [115], [116], [117] showed that the PF scheduler provides the lowest packet delay, whereas the Queue-HOL-MLWDF scheduler provides the lowest packet loss. This study shows how the streaming quality is affected by the scheduling approaches which

provide different packet delay and packet loss in the wireless LTE link during the adaptive video streaming sessions.

The results that capture the streaming quality are shown in Figs. 3.2-3.4. It can be observed that for up to 30 users in the system the rebuffering probability is insignificantly affected by the scheduler, however when the number of users increases to 40 users, the rebuffering significantly increases, and in this case the PF scheduler provides the lowest rebuffering probability. The rebuffering duration is lowest when the PF scheduler is used, except in the case of 40 users, where the rebuffering duration is insignificantly affected by the scheduler. In addition, in the case of 40 users, a decrease in the rebuffering duration that occurs as a joint effect of the high number of rebuffering events and the used adaptation. Further, the PF scheduler provides highest video representation (expressed in kbps) regardless of the number of users in the system. Therefore, in the proposed system model, the scheduler that provides the lowest packet delay is the most suitable for delivering high-quality adaptive video streaming. In order to further investigate the background rationale of the aforementioned remarks, the trade-off between the packet loss and the packet delay for the schedulers in the case of 40 users in the system is investigated. In this case the difference in the rebuffering probability is most notable, and the rebuffering probability is the most important QoE factor of the video streaming session [81].

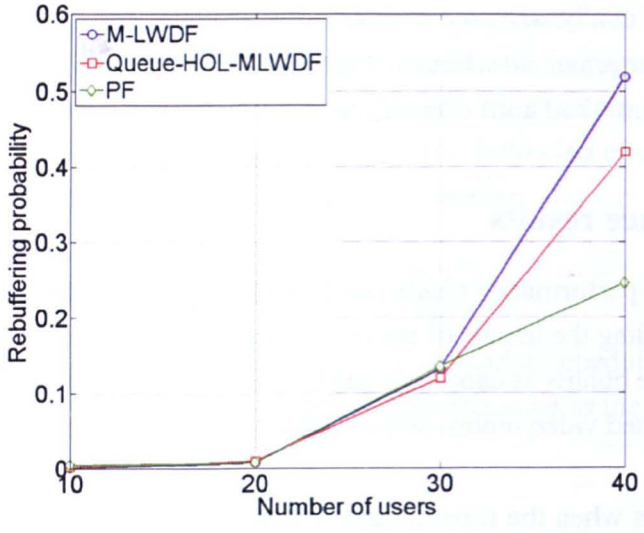


Fig. 3.2 Rebuffering probability for different scheduling approaches.

Table 3.1 shows the average packet loss and the average packet delay in the case of 40 users obtained through the simulations for the considered video representations. The packets from a higher video representation experience higher packet loss and packet delay. Considering the lowest representation, the PF provides 11.4% and 41.3% lower packet delay

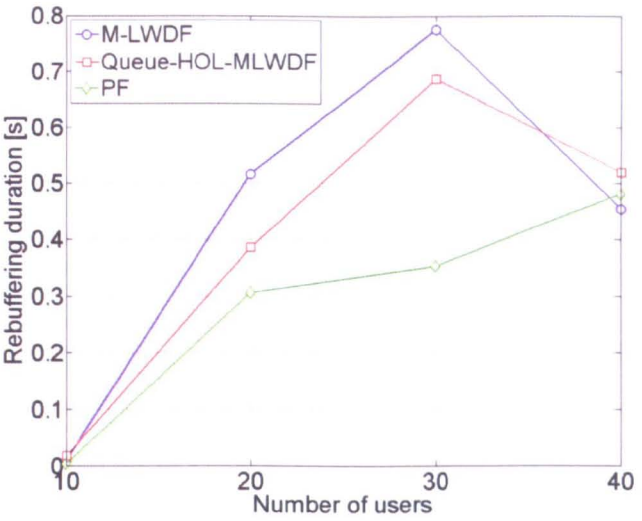


Fig. 3.3 Rebuffering duration for different scheduling approaches.

with respect to the Queue-HOL-MLWDF and M-LWDF, whereas the PF packet loss is 5.3 and 8.6 percentage points higher with respect to the M-LWDF and Queue-HOL-MLWDF. Further, for the highest representation, it can be observed that the PF provides 28.5% and 25.3% lower packet delay with respect to the M-LWDF and Queue-HOL-MLWDF, whereas the PF packet loss is 4.2 percentage points higher with respect to the M-LWDF, and 2.1 percentage points lower with respect to the Queue-HOL-MLWDF. There is significant packet delay difference and similar packet loss when comparing the schedulers. The scheduler with the lower packet delay provides better quality of the adaptive video streaming.

Table 3.1 Scheduler performances for 40 users.

Scheduler	Bitrate [kbps]	Packet loss [%]	Packet delay [ms]
PF	100	9.2	17.4
	200	21.5	27.4
	300	35.0	42.7
	400	46.4	55.3
M-LWDF	100	3.9	24.6
	200	12.4	39.0
	300	27.0	58.2
	400	42.2	71.1
Queue-HOL-MLWDF	100	0.6	19.4
	200	7.2	34.7
	300	31.3	60.1
	400	48.5	69.3

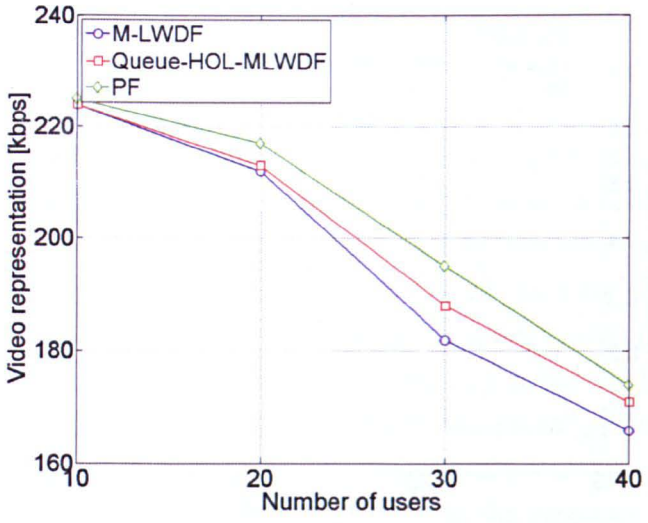


Fig. 3.4 Video representation for different scheduling approaches.

3.1.4 Study summary

The work presented in this study proposes a system model for emulating DASH-based end-to-end video delivery from a server located in a CDN to multiple LTE users. The focus in this study is on analyzing how the scheduling affects the adaptive streaming quality depending on the number of users in the system by considering the rebuffering probability, the rebuffering duration and the video representation. Three scheduling approaches with different packet delay and packet loss performances were considered, and it is observed that up to 30 users the rebuffering probability is unaffected by the scheduler, whereas the rebuffering duration and the video representation are improved when the scheduler with lower delay is used. Further, in the case of 40 users, the scheduler with the lowest packet delay provides significantly improved rebuffering probability, whereas the rebuffering duration and video representation are slightly affected in terms of percentages. This is due to the packet delay and packet loss difference of the considered schedulers in the case of 40 users, where the difference in packet delay is significantly higher compared to the packet loss difference.

The results show very high packet delay and packet loss values for the considered schedulers even for low DASH bitrates, especially when the number of users in the system is high. Therefore, significant improvements in the scheduling procedures for LTE are required in order to facilitate DASH services with satisfactory QoE, in particular for serving large number of users.

3.2 DASH live video streaming over cloud

In this study, an operational streaming server setup is used, where the Wowza streaming engine is deployed over a commercial cloud service (Rackspace). The Wowza streaming engine is composed of transcoder and streamer for configuring DASH live streaming sessions with different segment sizes between a source side and DASH client that receives the video content via WLAN network. The main focus in the study is on the segment size impact on QoS for DASH live streaming over cloud. The QoS of the DASH live streaming sessions is captured in uplink (source to cloud server) and downlink (cloud server to DASH client) via throughput and packet inter-arrival time measurements for different segment sizes. The uplink measurements show that the segment size selection results in a statistically significant difference in the QoS, whereas the downlink measurements show that the segment size selection affects the QoS. As a result, the end-to-end QoS is affected by the segment size selection, hence selecting appropriate segment size improves the QoS for DASH live streaming over cloud.

3.2.1 Related work

DASH streaming over cloud is an emerging topic related to adaptive video streaming systems. To the best of the author's knowledge, the work presented in this study is the first to investigate how the segment size selection impacts the QoS for DASH live streaming sessions over cloud. Several related studies are discussed in this subsection.

The work in [120] presents a general framework for Adaptive Mobile Video Streaming in the Cloud (AMVSC) that is based on adapting the streaming flow by using private agents and scalable video coding (SVC) technique with link quality feedback. The contributions in this work are summarized as follows: the AMVSC platform adapts the streaming bitrate per user by using SVC and private agents that are dynamically initiated in the cloud, thus providing the best possible streaming experience; AMVSC supports distributing video streaming using a 2-tier structure composed of the Content Delivery Network (CDN) and data center. The evaluation results target different SVC resolution schemes, and include the encoding delay and overhead in the cloud, and the watching delay.

A comprehensive study on the feasibility of DASH in the cloud is presented in [121] with focus on Video on Demand (VoD) services. The main findings in this study show the following: the geographical location is not the dominant factor impacting the streaming performance; there is a cloud instance dependent threshold with respect to the parallel streaming sessions, and above this threshold the video quality dramatically drops; the Amazon cloud instances provide 70% higher throughput compared to Rackspace at the cost

of supporting a smaller number of parallel client sessions. In the experiments, the Big Buck Bunny video is segmented with segment size set at $2s$ and encoded at 17 representations in the range between 100 kbps and 10 Mbps. The bitrate, the video quality, and the video initiation time are considered as relevant metrics in this study.

Another work on DASH streaming over cloud [122] additionally considers Device to Device (D2D) cooperation. This work focuses on live streaming sessions, where a group of adjacent mobile devices obtains live stream that is transcoded in the cloud according to the accumulated bandwidth of the group, i.e., a high quality stream. The streaming to the devices is coordinated by a Captain Device via Wi-Fi network, and the D2D coordination assumes exchanging video fragments (parts of the segment) among the devices via Bluetooth that results in improved video quality. The results are presented only in terms of bitrate of the streamed segments.

The work in [123] proposes a dynamic scheduling on the video transcoding for DASH in cloud environment. The following contributions are provided by this work: estimation of the video transcoding time depending on the segment duration and the target bitrate based on measured statistics and probabilistic theory; job distribution mechanism for load balancing between the processors (the scheduler assigns the fastest processor for high priority jobs, i.e., pending viewing requests and dynamically optimizes the transcoding for the requested bitrates and in the following step the remaining bitrates are transcoded). Several evaluation metrics are considered, including the MOS according to the model in [81] and the load balance factor related to the processors.

A method for high resolution video streaming by cloud and DASH is presented in [124]. The large video size file (panorama, 8K UHD, etc.) is divided into many patches that are distributed to the client by using the cloud servers and DASH transmission, hence the client receives streaming at high rate regardless of the particular cloud server transmission capabilities.

The work in [125] shows optimal video transcoding in CDNs networks through several online transcoding policies that transcode the segments only to those bitrates requested by the user rather than transcoding to all possible bitrates. The proposed approach uses a Markov prediction model that predicts the bitrate for the next requested segment with high accuracy, therefore the quality of the users experience is not affected. As main conclusion, the authors show that online transcoding schemes reduce transcoding resources by over 95% without a major impact on the QoE. In addition, the authors show that increasing the size of the segment and transcoding the segment to a higher bitrate increases the required transcoding time.

In contrast to these works, the work presented in this study focuses on the findings related to the segment size impact on the throughput and the packet inter-arrival time for uplink (source to cloud server) and downlink (cloud server to client) transmissions during DASH live streaming session. In particular, it is shown that the QoS is improved when appropriate segment size selection is performed.

3.2.2 Experimental setup

A high-level diagram of the streaming setup over cloud is shown in Fig. 3.5. Streaming video over the cloud is experimented using the following apparatus: capture device, streamer at the source, cloud server, streaming engine (transcoder + streamer) and DASH-based client. In order to analyze the impact of the segment size selection, the streams are segmented into $T = 2s$, $T = 4s$, and $T = 8s$, corresponding to small, medium and large segment size. The study is based on the measurements obtained in the uplink and downlink. The major components of the setup are discussed as follows.

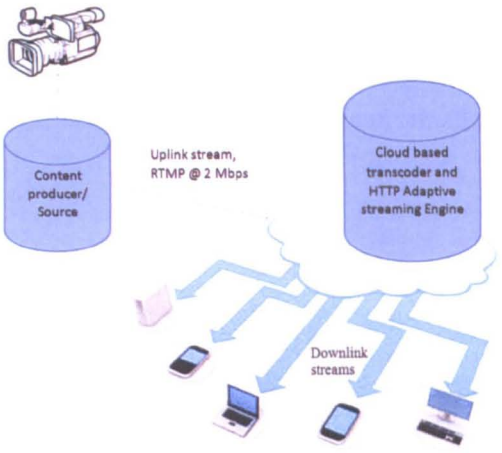


Fig. 3.5 Experimental setup for HTTP streaming over cloud.

3.2.2.1 Source side

A Logitec 500 webcam (1.3 Mega pixel) and Wirecast Version 5.0 are used to capture a dynamic scene at 720p resolution and to upstream the captured video using Real Time Messaging Protocol (RTMP) stream at 2 Mbps to the cloud. The sessions last for 600 seconds, where high dynamic content from the Frozen movie is streamed after being captured by the webcam. The intra frame interval is set to 2s, 4s, and 8s in order to facilitate segmentation at the cloud server. The video and audio compression are based on the H.264/AVC and AAC

codecs, respectively. A 100 Mbps LAN shared connection is used to up-stream the captured content to the cloud server.

3.2.2.2 Cloud server

A Rackspace cloud instance is used to run the HTTP adaptive bitrate streaming using the incoming stream from the source. The used cloud server configuration is described as follows: Ubuntu 12.04 LTS (Precise Pangolin) OS, 8 GB RAM, 8 core CPU, and 160 GB SSD disk. In addition, the *pcap* in the *libpcap* library was used to capture the incoming packets to the cloud server from the source. These traces were later analyzed to obtain the statistics related to the uplink transmissions. Furthermore, the Wowza streaming engine (version 4.1.0) is used to transcode and downstream the incoming stream in real-time. The incoming 2 Mbps stream is transrated using H.264/AVC into three bitrate streams, i.e., three video representations: 200 kbps and 284×160 pixels, 850 kbps and 640×360 pixels, and 2 Mbps and 1280×720 pixels.

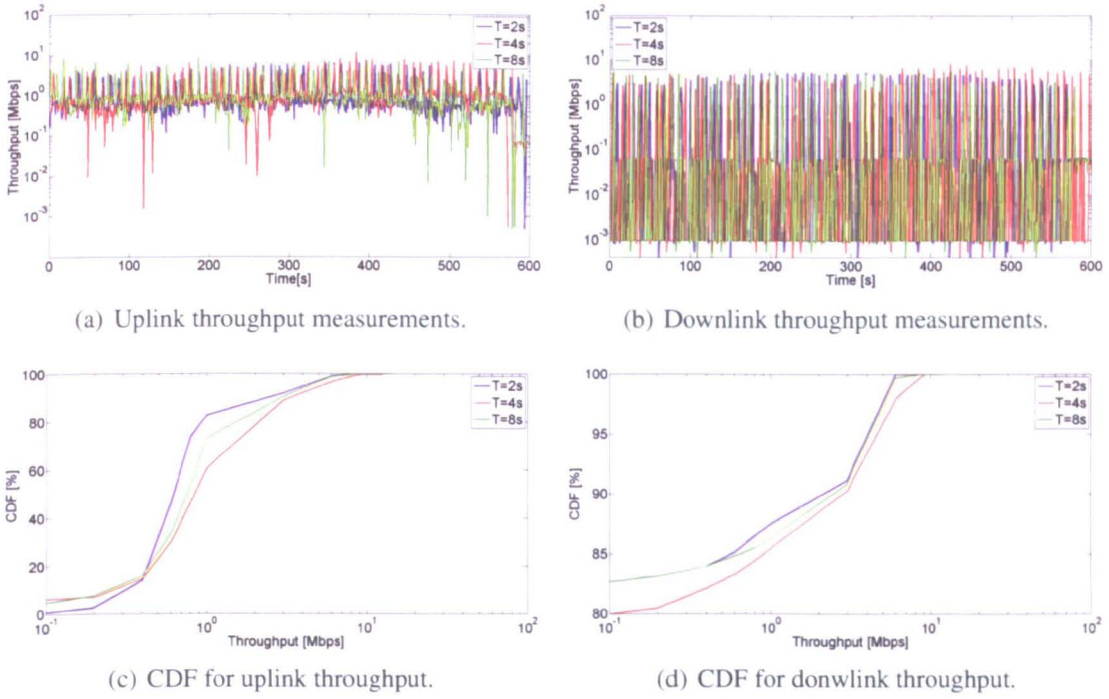
These streams were then segmented into different segment sizes based on the key frame intervals set at the source (i.e., 2s, 4s, 8s). The segments from each stream were then listed in a manifest file in order to enable HTTP adaptive streaming to the DASH client. The Wowza streaming engine makes the segments and the manifest file available and accessible for connecting clients.

3.2.2.3 Client side

DASH client is used to receive the DASH content from the cloud where the client selects segments from the most appropriate video representation depending on the network condition. The availability and the location of the segments is known to the client through the manifest file. The connection between the cloud and the client is an IP core network and a WLAN connection in the last hop. Wireshark is used at the client to capture the incoming packets in real time. These files are later analyzed to obtain statistics related to downlink transmission.

3.2.3 Evaluation results

This subsection presents the evaluation results from the conducted tests. Results are presented for three segment sizes corresponding to the key frame interval of the source, i.e. $T = 2s$, $T = 4s$, and $T = 8s$, referred to as small, medium and large segment size. The QoS of these sessions in uplink and downlink is observed via throughput and packet inter-arrival time measurements. In the end, the end-to-end QoS is discussed from a DASH viewpoint.



(a) Uplink throughput measurements.

(b) Downlink throughput measurements.

(c) CDF for uplink throughput.

(d) CDF for donwlink throughput.

Fig. 3.6 Measurements and CDFs for uplink/downlink throughput for different segment sizes.

3.2.3.1 Throughput

The uplink and downlink throughput measurements for the considered segment sizes are shown in Fig. 3.6(a) and Fig. 3.6(b). The samples are plotted for every second of the session. As expected, the uplink throughput is higher compared to the downlink throughput due to the larger network capacity of the LAN network compared to the capacity of the WLAN network. The throughput peaks correspond to the data transmission of the I frame as burst of packets. These peaks are higher for larger segment sizes since within one segment there is a higher number of P frames encoded with respect to the I frame.

The segment size impact on the throughput in uplink and downlink is shown in Fig. 3.6(c) and Fig. 3.6(d) through Cumulative Distribution Functions (CDFs). These functions are plotted based on the measured samples. In uplink, a small segment size provides the lowest throughput, whereas medium and large segment sizes increase the throughput. The highest uplink throughput is obtained with the medium segment size $T = 4s$. In downlink, the same trend as in uplink with respect to the segment size is observed, however the difference in the throughput is less dependent on the segment size.

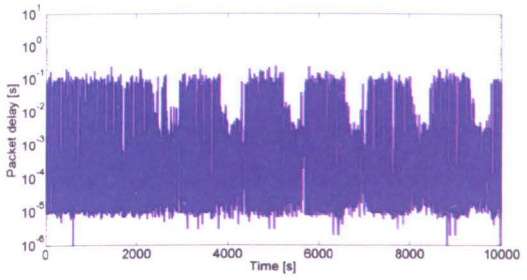
3.2.3.2 Packet inter-arrival time

The packet-inter arrival time shows the elapsed time between the reception of two successive packets. The uplink and downlink packet inter-arrival time measurements for the considered segment sizes are shown in Fig. 5.11. The samples are calculated by considering the timestamps of the successive packets. In the plots, 10.000 samples are shown in order to observe the segment size dependency. As expected, the packet inter-arrival time in uplink is less variable compared to the packet inter-arrival time in downlink due to the larger network capacity of the LAN compared to the capacity of the WLAN. Additional reason for this effect is the application layer protocol. In uplink, the RTMP protocol produces a CBR-like traffic transported via the TCP protocol, whereas in downlink, the HTTP protocol that is used to obtain the video segments from the server uses sequential TCP connections for each segment. This produces peaks in the packet inter-arrival time which are proportional to the number of requests for video segments (using smaller segment sizes results in a higher number of segment requests to the server).

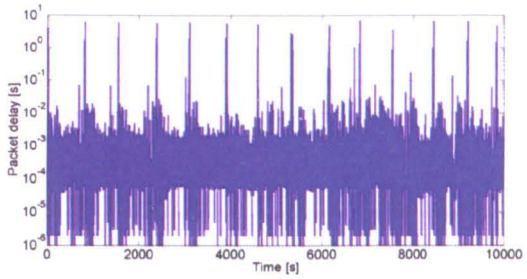
The segment size impact on the packet inter-arrival time in uplink and downlink is shown in Fig. 3.7(g) and Fig. 3.7(h) through CDFs. These functions are plotted based on the measured samples. Similar trends in the uplink and downlink are observed; increasing the segment size produces better results with respect to the packet inter-arrival time. Small segment size results in more frequent changes due to the aforementioned remarks related to the application protocols.

3.2.3.3 Confidence intervals

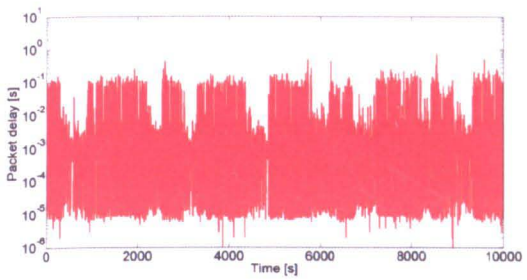
The confidence intervals (CI) of the throughput and the packet inter-arrival time measurements for the uplink and downlink measurements are shown in Fig. 3.8 and Fig. 5.2. A statistically significant difference in the measurements is observed via non-overlapping confidence intervals in the uplink. Hence, in the proposed setup, it is shown that appropriate segment size selection provides statistically significant difference in the throughput and the packet inter-arrival time for the uplink transmission, however there is no statistically significant difference in the downlink transmission, but the segment size selection affects the downlink throughput and packet inter-arrival time. In the setup, the medium segment size ($T = 4s$) provides highest throughput in uplink and downlink, whereas the large segment size ($T = 8s$) provides lowest packet inter-arrival time in uplink and downlink.



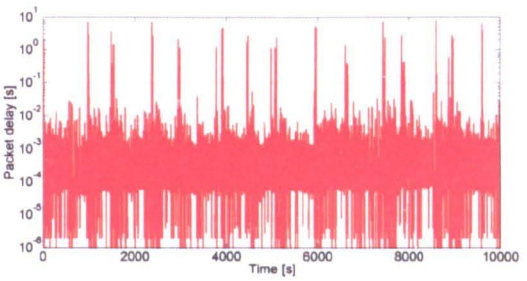
(a) Uplink packet inter-arrival time for T=2s.



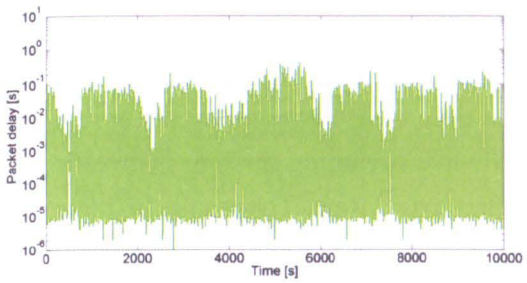
(b) Downlink packet inter-arrival time for T=2s.



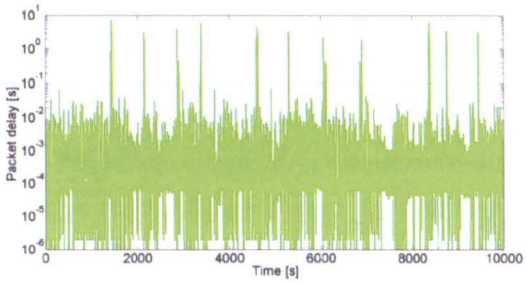
(c) Uplink packet inter-arrival time for T=4s.



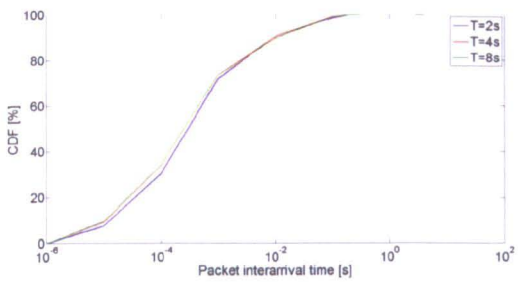
(d) Downlink packet inter-arrival time for T=4s.



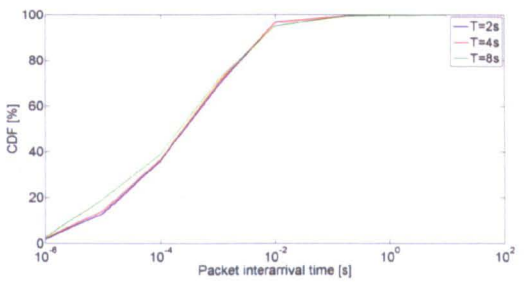
(e) Uplink packet inter-arrival time for T=8s.



(f) Downlink packet inter-arrival time for T=8s.



(g) CDF for uplink packet inter-arrival time.



(h) CDF for downlink packet inter-arrival time.

Fig. 3.7 Measurements and CDFs for uplink/downlink packet inter-arrival time for different segment sizes.

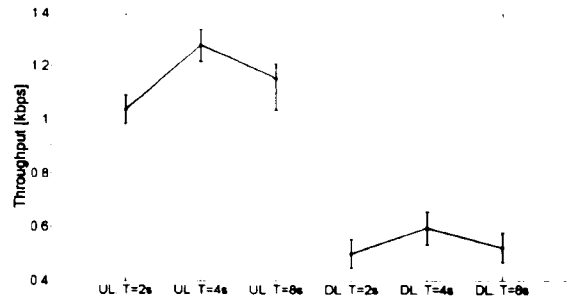


Fig. 3.8 CI for uplink/downlink throughput with different segment sizes.

Fig. 3.9 CI for uplink/downlink packet inter-arrival time with different segment sizes.

3.2.3.4 End-to-end QoS

Based on the throughput and the packet inter-arrival time measurements for the uplink and downlink transmission, the end-to-end QoS for DASH live streaming sessions is investigated in terms of average end-to-end throughput/packet inter-arrival time. The results in Table 3.2 show the average values of the measurements. Highest average throughput is achieved with medium segment size ($T = 4s$) and lowest average packet inter-arrival time is achieved with large segment size ($T = 8s$).

Table 3.2 End-to-end QoS performance

Parameter	T=2s	T=4s	T=8s
Throughput [Mbps]	0.77	0.94	0.84
Packet inter-arrival time [s]	0.010	0.01	0.008

In the context of DASH, the uplink measurements show the transmission and the availability of the video data at the cloud server; further, this data is transcoded and streamed according to the requests from the DASH client, and this transmission is captured via the downlink measurements. The uplink and downlink measurements show the end-to-end QoS for DASH live video streaming over the cloud, where the throughput is proportional

to the quality of the streamed video and the packet inter-arrival time affects the streaming continuity. High end-to-end throughput enables the client to select segments from a higher video representation, whereas low packet inter-arrival time facilitates smooth video playout.

3.2.4 Study summary

This study investigates the segment size impact on the QoS for DASH live streaming over the cloud. The QoS is captured via throughput and packet inter-arrival time measurements for uplink transmission (source to cloud server) and downlink transmission (cloud server to DASH client). The transcoding of the RTMP/TCP uplink stream in multiple streams with different quality (video representations) is performed in the cloud. These streams are organized in segments with predefined segment size. The DASH client selects segments from the most appropriate video representation depending on the network condition and it obtains the video data as HTTP/TCP downlink stream.

This study shows that the segment selection results in a statistically significant difference in the QoS in uplink and it additionally affects the QoS in downlink. As a result, the end-to-end QoS is affected, hence selecting appropriate segment size improves the QoS for DASH live streaming over the cloud. In the setup, using medium segment size $T = 4s$ provides highest average throughput, whereas large segment size $T = 8s$ provides lowest average packet inter-arrival time. The presented observation in this study highlight the need for carefully tuning the segment size parameter during session setup in order to improve the end-to-end QoS of live DASH video streaming service.

3.3 Medical video streaming utilizing DASH

This study presents an approach for medical video streaming based on the MPEG-DASH standard. Nowadays, medical video streaming over the Internet has various applications like university medical education, virtual diagnosis, sharing of knowledge within the medical faculty, etc. [126], [127]. Furthermore, different transmission platforms/networks have been utilized for medical video transmission [128], and a variety of architectures have been proposed for telemedicine services [129],[130]. However, to the best of the author's knowledge, this is the first study that analyses MPEG-DASH-based medical video streaming over the Internet in the context of medical applications. In this study, two adaptation approaches at the client are tested and parameters of the QoE are evaluated, i.e., the number and duration of re-buffering events, the added time in the playout, and the PSNR of the medical video frames are captured depending on the used adaptation strategy and the predefined segment size.

The results show that the selection of MPEG-DASH-related parameters and the adaptation approach at the client result in improved quality of the medical video streaming.

3.3.1 Evaluation setup

This subsection elaborates the evaluation setup, i.e., the video encoding and segmentation, the streaming model, and the proposed client adaptation approaches.

3.3.1.1 Video encoding and segmentation

The DASHEncoder tool [12] is used for encoding and segmentation of the video. A 636×420 resolution ultrasound video sequence with $T_v = 3s$ duration is encoded at four different representations corresponding to 400 kbps, 600 kbps, 800 kbps and 1 Mbps. Video streaming is tested for two different segment sizes, set at $T = 1s$ and $T = 3s$, where the sequence is looped in order to provide sufficient media frames for streaming analysis. The Group of Pictures (GOP) size is set to correspond with the used segment size, hence the intra (I) frame repeats at the beginning of every segment.

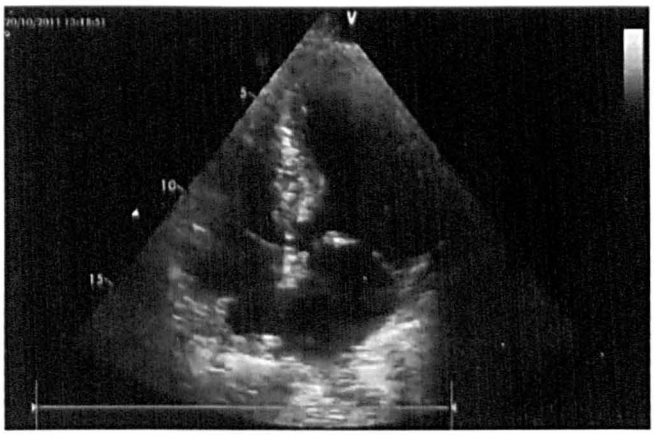
Table 3.3 shows the average PSNR value of each segment for different representations and the average PSNR values (Seg_{av}) over the whole representation. It can be noted that the PSNR variation range is approximately 2.5 dB between the lowest and the highest representation across all segment sizes. The clinically acceptable PSNR threshold depends on the particular medical application. The authors in [131] adopt 30 dB threshold for capsule endoscopy based on objective criterion of medical doctors; the authors in [132] provide medical Quality of Service (QoS) provision for ultrasound streaming, whereas the authors in [133] adopt a threshold of 35 dB for ultrasound video communication in the proposed telemedicine platform. Furthermore, according to Panayides et al. [134], a value of 35 dB provides sufficient quality for atherosclerotic plaque ultrasound video encoding based on H.264/AVC. Therefore, in this study, the 35 dB threshold is adopted. The results in Table 3.3 show that in the setup, the threshold is only reached with the highest representation 1000 kbps (the average PSNR values are approximately 35 dB). The reasoning behind this is to investigate how the client's adaptation approach (Subsection 3.3.1.3) and the segmentation can be jointly used to reach the threshold. An illustration of the difference in the image quality for the lowest representation and the highest representation is shown in Fig. 5.1.

The effect of video segmentation is less evident (similar PSNR values for $T = 1s$ and $T = 3s$) due to the very low spatio-temporal changes in the ultrasound sequence. Reference [54] elaborates the drawback of small segment sizes, i.e., frequent usage of I frames that occur at the beginning of the segment. Besides the effect on the PSNR, the segmentation has

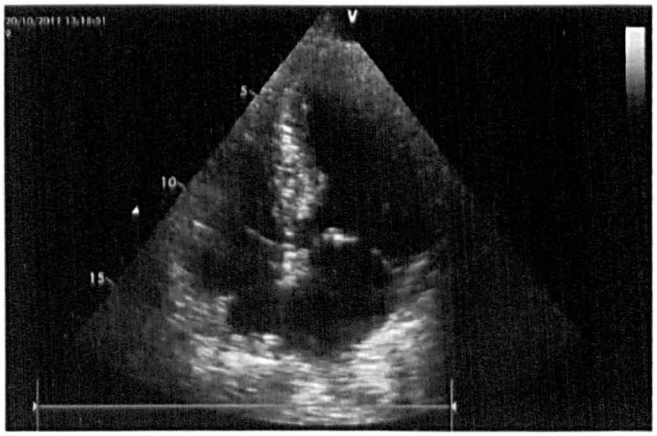
significant effect on the overall streaming quality (re-buffering events, duration of re-buffering events, etc.).

Table 3.3 Average PSNR for different segmentation/representations

	T=1s				T=3s
Repr.	$Seg1$	$Seg2$	$Seg3$	Seg_{av}	Seg_{av}
400Kbps	32.82	32.11	32.08	32.33	32.39
600Kbps	33.97	33.16	33.13	33.42	33.50
800Kbps	34.80	34.05	33.98	34.27	34.35
1000Kbps	35.47	34.77	34.68	34.97	35.03



(a) Representation at 400 kbps.



(b) Representation at 1000 kbps.

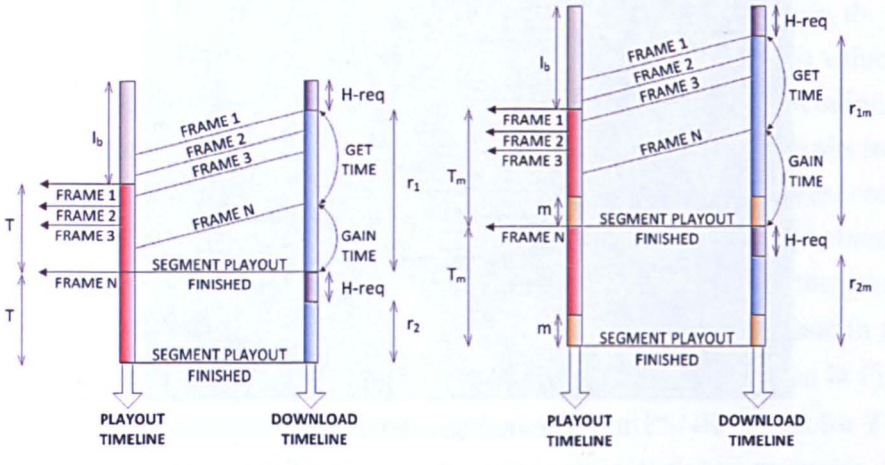
Fig. 3.10 Comparison between a frame from different representations.

3.3.1.2 Streaming model

The encoded video segments are sent to the client based on the proposed adaptation approach (Subsection 3.3.1.3) as consecutive Protocol Data Units (PDUs) that are reliably transmitted utilizing the HTTP/TCP protocols in the form of TCP packets. The video content from each segment is mapped into a corresponding number of TCP packets depending on the segment's representation and duration T. The adopted TCP packet size is MSS=1400 bytes, whereas the TCP throughput (packets/s) is modeled according to [119], as used in the first case study in this chapter 3.1, expression 4.18. The RTT round trip time is set between 100 and 200 ms, the timeout T_0 is set to 50 ms, $b = 2$ for jointly acknowledging packets and the advertised maximum TCP window size W_{max} is 32 packets.

3.3.1.3 Client adaptation approaches

The client downloads the segments from a particular representation, which is decided based on the used adaptation approach. The adaptation approach depends on several technical parameters (network TCP throughput, signaling requests (HTTP-GET), segmentation/encoding of the content, etc.) and on the client's supported capabilities (resolution, codecs, etc.). The following two segment selection approaches are proposed: *No margin* and *Margin-based*. These approaches are depicted in Fig. 4.9, and are based on the true streaming condition [15], i.e., the time needed to download video is less or equal than the time needed to playout the streamed video.



(a) No margin approach (b) Margin (m) based approach

Fig. 3.11 Different approaches at the client.

The approaches are elaborated by using the download and playout timeline for every segment at the client. The download of the $i-th$ segment starts (assuming that the MPD is known to the client) by sending an HTTP-GET request $H-req$ followed by *reserved time* r_i for the segment download. The actual segment download time depends on the TCP throughput $\sigma(p)$, the segment size T and its representation, and it is labeled GET TIME in Fig. 4.9. If the GET TIME is less then the reserved time, the remaining time difference is labeled GAIN TIME in Fig. 4.9, which is used for download of the following segment. The playout of the segment depends on the initial buffering I_b, the segment size T, and the frame rate at which the frames are played on the playout timeline.

The no margin approach provides playout at a frame rate used in the video encoding and segmentation, whereas the margin-based approach (the notations for this approach are indexed with m) provides additional time budget for selecting higher representation levels (corresponding to higher image quality) at the cost of decreasing the frame rate at the playout timeline. Hence, the playout frame rate is smaller compared to the rate used in the video encoding. The margin size m is defined as a fraction of the segment size T, i.e., $m = wT$, where w is the margin index. The margin index can be derived based on the acceptable decrease in the playout rate in the considered medical application. In this work, w is set in the $(0, 0.25)$ range.

3.3.2 Performance evaluation

The effect of the MPEG-DASH segmentation and of the client's adaptation approaches (no margin and margin-based) on the video quality is captured through the following performance metrics:

- Number of re-buffering events presented as percentage of segments affected by re-buffering events;
- Average duration of re-buffering events;
- Added time in the playout at the client;
- PSNR values.

The number of re-buffering events and the duration of re-buffering events are averaged across all video segments, where a re-buffering event occurs when the reserved time is not sufficient for the segment download. Since averages are considered, if the number of re-buffering events is equal to 1, that implies that re-buffering event occurs in every video segment. Therefore, lower re-buffering values are preferable and if the number of

re-buffering events is 0 then the true streaming condition is preserved. When the reserved time is not sufficient for the download of the segment, the playout of the segment stops until all the frames from the segment are downloaded. The elapsed time while these frames are downloaded is used to calculate the duration of re-buffering. As a result of the re-buffering and the duration of re-buffering, the actual viewing time of the streamed video is extended. This effect is captured through the added time in the playout at the client. In the end, the PSNR of the streamed video frames is observed, where the frames are encoded as elaborated in subsection 3.3.1.1. The results for the performance metrics are shown in Figs. 3.12-3.15.

The results for the re-buffering events, reported in Fig. 3.12, show that the margin-based approach outperforms the no margin approach ($w = 0$) only when the segment sizes are large ($T = 3s$). This effect is more evident for higher values of the margin index. In contrast, when the segment size is small ($T = 1s$), the no margin approach decreases the number of re-buffering events.

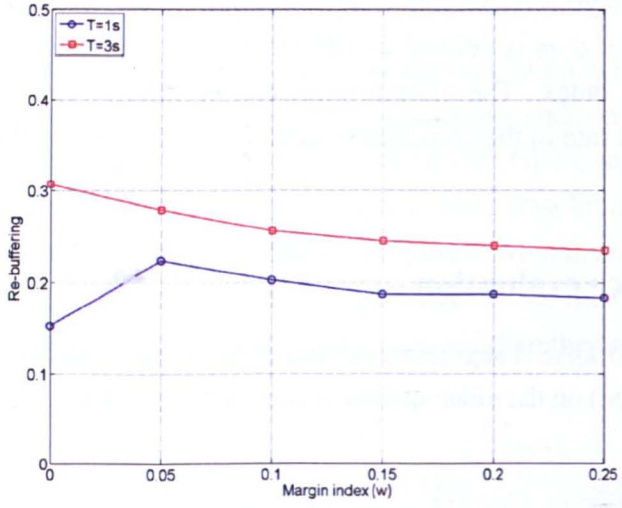


Fig. 3.12 Re-buffering frequency[for different segment sizes and margin indices.

The results for the duration of re-buffering events, reported in Fig. 3.13, show that the no margin approach outperforms the margin-based approach, since increasing the margin index results in higher durations of re-buffering events.

Based on the results for the re-buffering events and the duration of re-buffering events, the added time in the playout is calculated by averaging across 10.000 simulations and varying the duration of the medical video sequence (T_v). The results, reported in Fig. 3.14, show that there is a linear increase in the added time in the playout when the video duration is increased. The margin-based approach ($w = 0.25$) reduces the added time in playout only for large segment size ($T = 3s$), whereas the no-margin approach produces lower added time for small segment size ($T = 1s$).

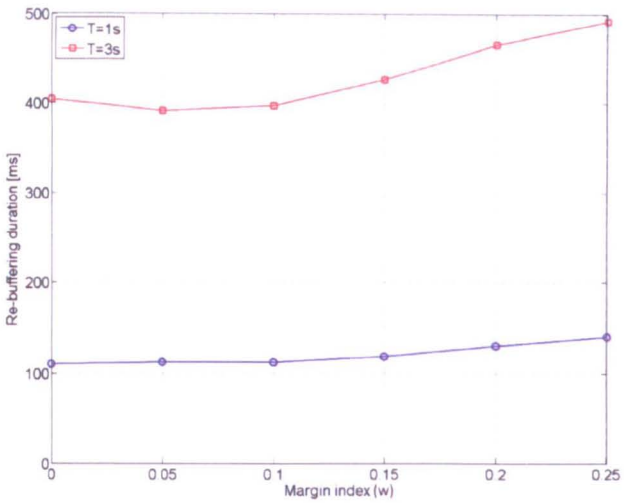


Fig. 3.13 Duration of re-buffering for different segment sizes and margin indices.

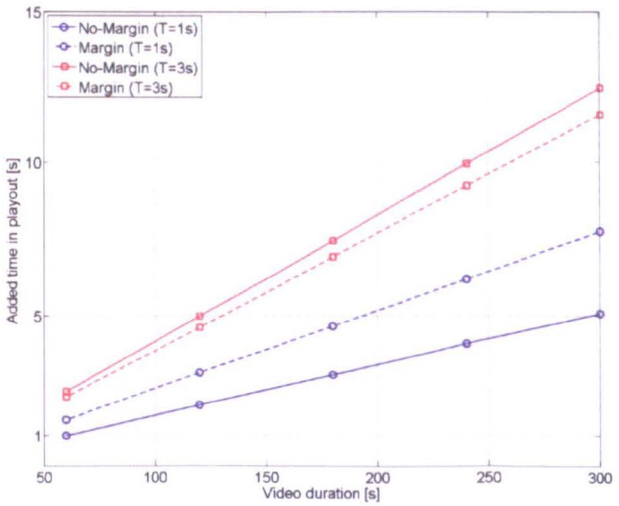
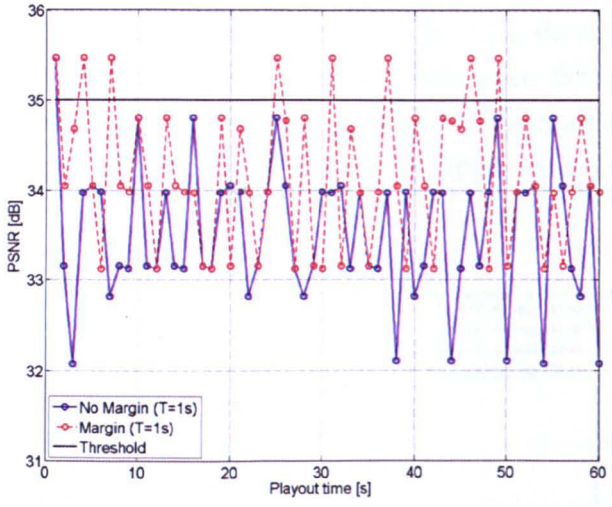


Fig. 3.14 Added time in playout.

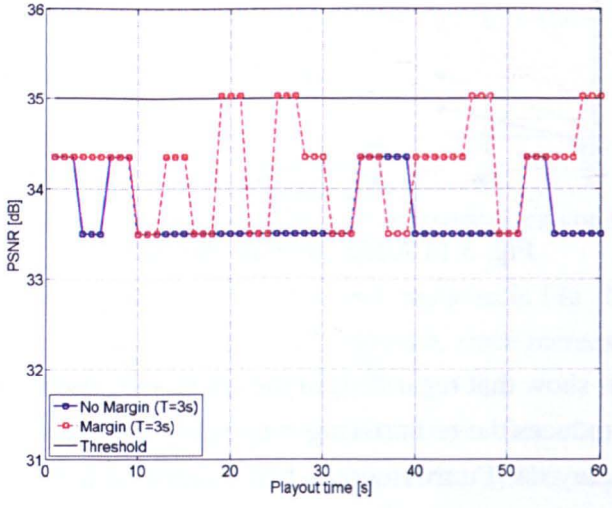
The previous results show that regardless of the adaptation approach, the utilization of smaller segment sizes reduces the re-buffering events, the duration of re-buffering events, and the added time in playout. Furthermore, when utilized with the no-margin approach, small segment sizes further improve these streaming performances. However, the drawbacks of small segment sizes [54] imply that the lossless PSNR of the frames would be reduced and possibly not suitable for the medical application. In order to increase the PSNR of the source, large segment sizes should be used. The streaming with large segment sizes can be improved when the margin-based approach is used, hence the percentage of segments

affected by re-buffering and the added playout time are decreased at the cost of higher duration of re-buffering.

The PSNR results, reported in Fig. 3.15, show that, regardless of the segment size, the margin-based approach provides better PSNR and frequent usage of the highest representation (35 dB threshold).



(a) PSNR performances for the two approaches and $T = 1s$.



(b) PSNR performances for the two approaches and $T = 3s$.

Fig. 3.15 PSNR results for different segment sizes and adaptation approaches at the client.

3.3.3 Study summary

This study presents an analysis of medical video streaming utilizing the emerging MPEG-DASH standard. To the best of the author's knowledge, this is the first work which considers adaptive video streaming based on MPEG-DASH for medical video content. The presented results show that the selection of suitable MPEG-DASH parameters (representation, segment size) and adaptation approach improves the video streaming quality. Furthermore, small segment sizes and the adoption of the no-margin approach improve the streaming performances (number of re-buffering events, duration of re-buffering, and added time in playout), however due to video encoding and segmentation the resulting PSNR for medical video could be insufficient. Therefore, larger segment sizes (hence, higher PSNR) should be used with the margin-based approach in order to improve the ultrasound video streaming performance. Using DASH in medical use cases is especially interesting and emerging topic due to lossless delivery of information via DASH, however in order to provide sufficient quality for diagnostic purposes adequate adaptation approach and segment size need to be selected.

3.4 Chapter summary and contributions

The chapter on DASH case studies presents three different cases where the core principles outlined in the MPEG-DASH standard are applied in the proposed system models. The models in these studies are designed by using existing analytical models, traces from simulators, real experiments in operational setup, and standalone numerical simulations. As a results of the work presented in this chapter, the following presents a summary of the contributions and works:

- DASH-based video streaming over LTE network. This is a novel model in the literature for the evaluation of end-to-end DASH video streaming to multiple LTE users that combines analytical and simulation approaches. Hence, the impact of the deployed scheduling on the adaptive video streaming quality is investigated where the quality is captured via the rebuffering probability, the average rebuffering duration, and the video representation depending on the number of users. The results show that the impact of the scheduling is significant for the considered metrics; in particular, the impact on the rebuffering probability is most notable. Further, the rationale behind this observation is the difference between the combination of the packet loss and the packet delay provided by the used scheduling.

- QoS for live DASH streaming. This case study investigates the segment size impact on the uplink and downlink QoS throughput and packet inter-arrival time measurements for DASH live streaming. The results show that the segment size selection has a statistically significant impact on the uplink QoS and it affects the downlink QoS.

- Utilizing the MPEG-DASH framework for medical video streaming. This case study focuses on utilizing the MPEG-DASH framework for medical video streaming. DASH is suitable for medical videos since the HTTP/TCP protocols provide no packet loss during the transmissions. An analysis of the quality requirements for the medical video is presented and two client adaptation techniques are considered for different segment sizes. The results show the streaming performance dependency on the selected segment size and the adaptation approach at the client.

Chapter 4

Teletraffic and statistical analysis of DASH systems

Mathematical tools and concepts are essential for acquiring quantifiable knowledge. This chapter presents a teletraffic and statistical analysis of DASH systems where the former proposes a novel segment-based teletraffic model and the later proposes a general methodology for statistical evaluation. The teletraffic analysis is based on teletraffic engineering which is the application of traffic engineering theory to telecommunication systems based on concepts from statistics and queuing theory. The model proposed in this study is used to investigate the functionality of DASH from teletraffic viewpoint by using probability-based metrics which are derived to evaluate the effect of the segment duration/size of the encoded video at the server and of the buffer size at the client. The statistical analysis proposes a general framework to evaluate the video quality experienced by DASH users. In general, the quality is tightly related to the bit-rate of the video representation selected and to the frequency and duration of the rebuffering events. Therefore, these aspects are investigated from a statistical viewpoint. In the end, a transvariation-based approach is used in the statistical analysis in order to conduct a comparative system analysis with respect to the segment size and to formulate guidelines relevant to the design of DASH systems. A shared feature in the two analyses is the effect of using different segment sizes under single predefined adaptation logic. The difference is twofold. First, in the teletraffic analysis different buffer sizes at the client under the same network conditions are considered, whereas in the statistical analysis different network conditions under no buffer size restrictions are considered. Second, the observed metrics in the two analysis provide different insights regarding DASH systems, i.e., the teletraffic analysis provides probabilistic assessment and conclusions relevant to the protocol stack, whereas the statistical analysis provides relevant statistics for parameters that influence the QoE.

This chapter is organized as follows. The teletraffic analysis is presented in Section 4.1, whereas the statistical analysis is presented in Section 4.2. Section 4.3 concludes the chapter.

4.1 Teletraffic analysis

This analysis is based on a novel segment-based teletraffic model for DASH. The general principles of MPEG-DASH are mapped on the proposed model, which is based on the application of probability and queuing theory. This model is utilized for probabilistic assessment of the DASH system performances through the probabilities of overflow, empty buffer and active buffer which are adopted as relevant performance metrics. The metrics are observed for different segment sizes of the encoded video at the server and different buffer sizes at the client. The authors in [54] investigate how different segment sizes affect objective video quality metrics. Similar insight are provided in this analysis, however the observed metrics provide insights from a teletraffic viewpoint and their impact is discussed in the context of the protocol stack.

The model is elaborated in Subsection 4.1.1, which includes the model's boundaries, the adopted segment selection approach, and the proposed performance metrics. Subsection 4.1.2 presents the obtained results and the study summary is resented in Subsection 4.1.3.

4.1.1 Segment-based teletraffic model for DASH

The mapping of the DASH streaming model on the teletraffic model is depicted in Fig. 4.1, where network and teletraffic views are presented to capture the video streaming in both domains.

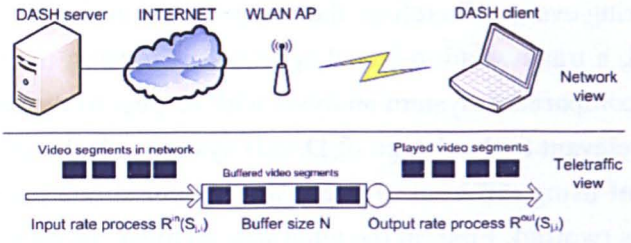


Fig. 4.1 System model: network and teletraffic view.

From the network side, the DASH encoded video segments are transfered from the server through the Internet, then the segments are received through the access network and played at the DASH client. It should be noted that the WLAN AP shown in Fig. 4.1 is selected as an example, since the model is more general and can refer to different types of access networks.

From the teletraffic viewpoint, the arrival of the video segments from the network and their playout at the client is modeled by the input and output processes $R^{\text{in}}(S_{\text{j,i}})$ and $R^{\text{out}}(S_{\text{j,i}})$, expressed in bits/second. These processes correspond to the average arrival/serving rates, which are used to calculate the throughput $\gamma(S_{\text{j,i}})$ for the observed segment $S_{\text{j,i}}$. The model used for the teletraffic analysis is based on the Schwartz diagram [135] and it is depicted in Fig. 4.2.

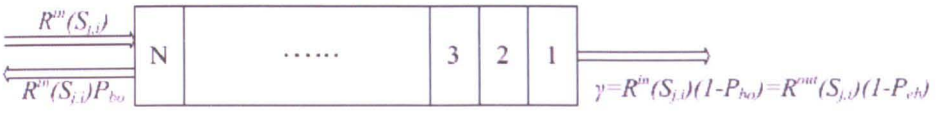


Fig. 4.2 Arrivals and serving rates in the queue.

A finite buffer (queue) size is adopted for which the arrival and serving rate vary for every segment. Furthermore, a simple M/M/1 queuing system is assumed as a reasonable approximation for the arrival and serving of the segment's packets. The buffer size N shows the number of packets that can be buffered at the client. Regarding the client's buffer the following three cases are observed:

- Buffer is full, hence the arriving packet is dropped;
- Buffer is active, hence the arriving packet is stored in the buffer and played out at its respective display time;
- Buffer is empty, there is no arriving packet and playout is stopped.

The effects of the three states are captured through the following proposed performance metrics for the teletraffic model:

- Probability of buffer overflow denoted as P_{bo};
- Probability of active buffer denoted as P_{ab};
- Probability of empty buffer denoted as P_{eb}.

The following notations are adopted for the teletraffic model:

- $S_{\text{j,i}}$ refers to the segment from representation j in the MPD file indexed with i;
- $B_{\text{j,i}}$ refers to the number of bytes used for the encoding of the segment $S_{\text{j,i}}$;
- $T_{\text{j,i}}$ refers to the time duration of the segment $S_{\text{j,i}}$;

- $t(S_{j,i})$ refers to the variable time needed for downloading segment $S_{j,i}$;
- r_j is the encoded rate of the jth representation, where $j = 1,2,3,..J$;
- M is the total number of segments that compose the video $i = 1,2,3,..M$;
- $R^{in}(S_{j,i})$ refers to the rate of the input process for segment $S_{j,i}$;
- $R^{out}(S_{j,i})$ refers to the rate of the output process for segment $S_{j,i}$;
- $\gamma(S_{j,i})$ refers to the throughput in the model.

The following assumption for the model is adopted: the number of bytes in segments with fixed duration from a particular representation is equal ($B_{j,1} = B_{j,2} = ... = B_{j,M}$, for all j) and depends only on the representation level (j) and the segment duration/size T. Henceforth, B_j is used to denote the number of bytes in a segment depending on the representation j.

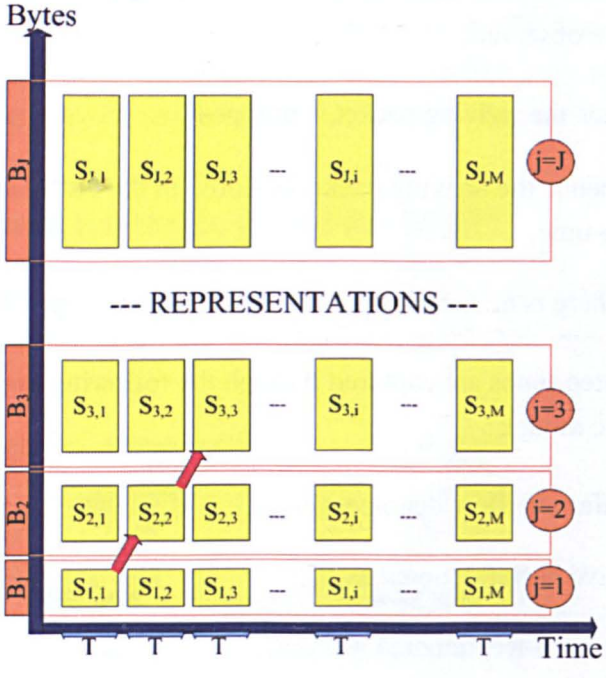


Fig. 4.3 DASH representations.

In Fig. 4.3, a simple diagram of the DASH representations over time is shown, with fixed segment duration T. The segments from each representation j, contain B_j bytes, which are streamed to the client. The switching among different representations is performed at segment level, i.e., by every T seconds.

4.1.1.1 Model boundaries

The boundaries of the input and output process in the teletraffic model are directly dependent on the MPD encoding of the video (representation levels, segment sizes) and on the network conditions/protocol efficiency (available throughput, round trip time, re-transmissions, etc.). The average arrival rate of the input process for the segment $S_{j,i}$ is calculated according to (4.1), whereas the average serving rate of the output process for the obtained segment $S_{j,i}$ is calculated according to (4.2):

$$R^{in}(S_{j,i}) = \frac{B_j}{t(S_{j,i})}; \tag{4.1}$$

$$R^{out}(S_{j,i}) = \frac{B_j}{T}. \tag{4.2}$$

Each segment from the MPD is delivered utilizing the HTTP/TCP protocols, where the packet size L of the Protocol Data Unit (PDU) is 1430 bytes. The segment duration within one MPD is fixed regardless of the representation and the segment index ($T_{j,i} = T$). The following set of integers $\{K_1, K_2, K_3, .., K_J\}$ is introduced to represent the number of packets that are needed for the delivery of a particular segment that corresponds to representation r_j, where $B_j = K_j L$, regardless of the segment's index (i).

The maximum arrival rate for the input process is obtained when a segment i from the highest representation level r_J is downloaded for the minimal time supported by the network $t_{min}(S_{J,i})$, whereas the minimum arrival rate for the input process is obtained when a segment i from the lowest representation level r_1 is downloaded for the maximum time possible in the network $t_{max}(S_{1,i})$. The t_{max} and t_{min} parameters are the maximum and minimum time durations needed for delivery of a packet in the network. Hence, the arrival rate in the input process is bound according to (4.3) and (4.4):

$$R^{in}_{max}(S_{J,i}) = \frac{B_J}{t_{min}(S_{J,i})} = \frac{K_J L}{\sum_{k=1}^{K_J} t_{min}}; \tag{4.3}$$

$$R^{in}_{min}(S_{1,i}) = \frac{B_1}{t_{max}(S_{1,i})} = \frac{K_1 L}{\sum_{k=1}^{K_1} t_{max}}. \tag{4.4}$$

Similarly, for the maximum/minimum serving rate of the output process the boundaries are dependent on the segment duration and are calculated according to (4.5) and (4.6):

$$R^{out}_{max}(S_{J,i}) = \frac{B_J}{T} = \frac{K_J L}{T}; \tag{4.5}$$

$$R_{\min}^{\text{out}}(S_{1,i}) = \frac{B_1}{T} = \frac{K_1 L}{T}. \tag{4.6}$$

4.1.1.2 Segment selection

The rate of the input process depends on the selected representation for the considered segment and the throughput in the network during the download of the segment. The next segment is selected depending on the averaged throughput experienced in a time interval equal to the segment duration (T). The averaging is done based on the experienced throughput for the packets of the segment. Therefore, assuming average throughput G in previous T seconds, then the next segment is selected from representation j such that $G - r_j$ is minimal for all available j. The selection of the segment determines the rate of the output process since the downloaded segment is played out at the client side. Additionally, the segment selection is tightly related to the buffer size since segments from high representations require larger buffer size at the client.

4.1.1.3 Model metrics

The metrics for the teletraffic model are derived based on the approach in [7], which provides a description of a general streaming model. This model proposes a consumption function $A(t)$ at the player, a transmission schedule function $S(t)$, and a buffer function $b(t)$, which are used to observe the accumulated data as function of time. These functions correspond to the segment-based functions in the teletraffic model, i.e., the output process $R^{\text{out}}(S_{j,i})$, the input process $R^{\text{in}}(S_{j,i})$, and the buffer for the segments, which is modeled with its size N. Furthermore, the general streaming model proposes an overflow limit function $B(t)$ as a sum of the consumption and the buffer function. This function determines the upper bound of the model for which the streaming process is preserved.

The performance metrics for the teletraffic model are derived based on [136]. The probability of buffer overflow $P_{\text{bo}}(S_{j,i})$ corresponds to the overflow limit function in the general streaming model and it is calculated as a blocking probability for a finite queue, depending on the queue size, according to (4.7):

$$P_{\text{bo}}(S_{j,i}) = \left(\frac{R^{\text{in}}(S_{j,i})}{R^{\text{out}}(S_{j,i})}\right)^N \frac{1 - \dfrac{R^{\text{in}}(S_{j,i})}{R^{\text{out}}(S_{j,i})}}{1 - \left(\dfrac{R^{\text{in}}(S_{j,i})}{R^{\text{out}}(S_{j,i})}\right)^{N+1}}. \tag{4.7}$$

The throughput $\gamma(S_{j,i})$ in the teletraffic model can be calculated considering the average arrival rate and the probability of buffer overflow according to equation (4.8), or considering the average serving rate and the probability of empty buffer according to equation (4.9):

$$\gamma(S_{j,i}) = R^{in}(S_{j,i})(1 - P_{bo}(S_{j,i})); \tag{4.8}$$

$$\gamma(S_{j,i}) = R^{out}(S_{j,i})(1 - P_{eb}(S_{j,i})). \tag{4.9}$$

The probability of empty buffer is given in equation (4.10), and is derived from equations (4.8) and (4.9).

$$P_{eb}(S_{j,i}) = 1 - \left(\frac{R^{in}(S_{j,i})(1 - P_{bo}(S_{j,i}))}{R^{out}(S_{j,i})}\right). \tag{4.10}$$

In the end, the probability of active buffer is calculated according to equation (4.11):

$$P_{ab}(S_{j,i}) = 1 - P_{eb}(S_{j,i}) - P_{bo}(S_{j,i}). \tag{4.11}$$

In order to investigate the effect of different segment durations of the encoded video, an averaging approach for these probabilities is introduced. The averaged probabilities of buffer overflow, empty buffer, and active buffer are calculated according to equations (4.12)-(4.14):

$$P_{bo}^{av} = \frac{1}{M}\sum_{i=0}^{M} P_{bo}(S_{j,i}); \tag{4.12}$$

$$P_{eb}^{av} = \frac{1}{M}\sum_{i=0}^{M} P_{eb}(S_{j,i}); \tag{4.13}$$

$$P_{ab}^{av} = 1 - P_{bo}^{av} - P_{eb}^{av}. \tag{4.14}$$

4.1.2 Performance results

The performance analysis is derived utilizing files obtained from tracing real MPEG-DASH streaming over a 54 Mbps WLAN network with the open-source platforms provided in [137]. From this trace the available data rate in the network during the download of the video segments with the considered segment durations is obtained. These values are used to determine the average arrival/serving rate for the input and output processes, which are used as an input to the teletraffic model implemented in Matlab. The serving rate of the output process is determined by the segment selection approach described in Subsection 4.1.1.2.

Fig. 4.4 shows the arrival and serving rates for the input and output process as a function of the number of video segments. The representations of the *Big Buck Bunny* sequence available in [137] are considered for the rate of the output process. It should be noted that different rates for the input and output process would change the observed values, however the trends would be preserved.

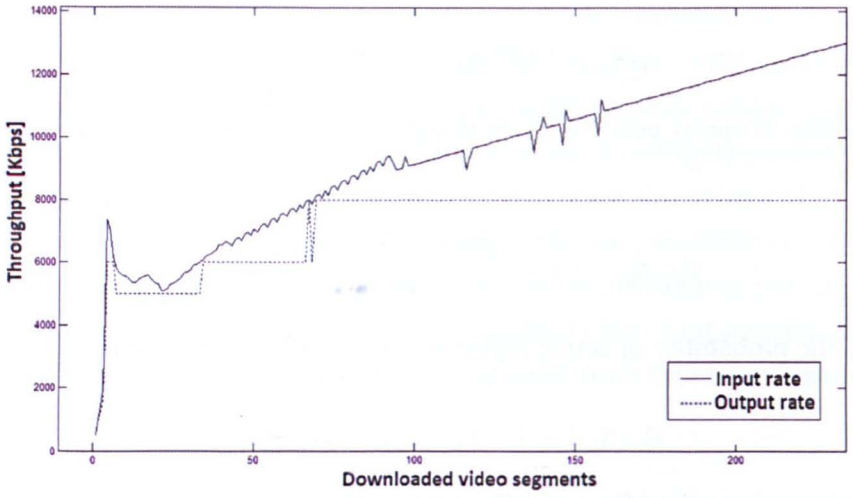


Fig. 4.4 Snapshot of the model's input trace.

The average probabilities of buffer overflow, empty buffer and active buffer for different segment durations and buffer sizes are plotted in Figs. 4.5-4.7.

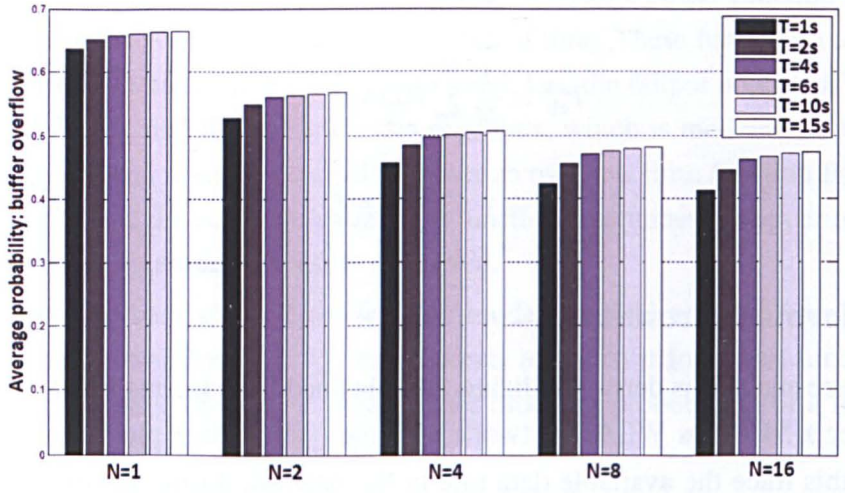


Fig. 4.5 Average probability of buffer overflow.

The configurations with higher buffer size N result as expected in decreased probabilities of buffer overflow and empty buffer. As a result, the probability of active buffer is increased,

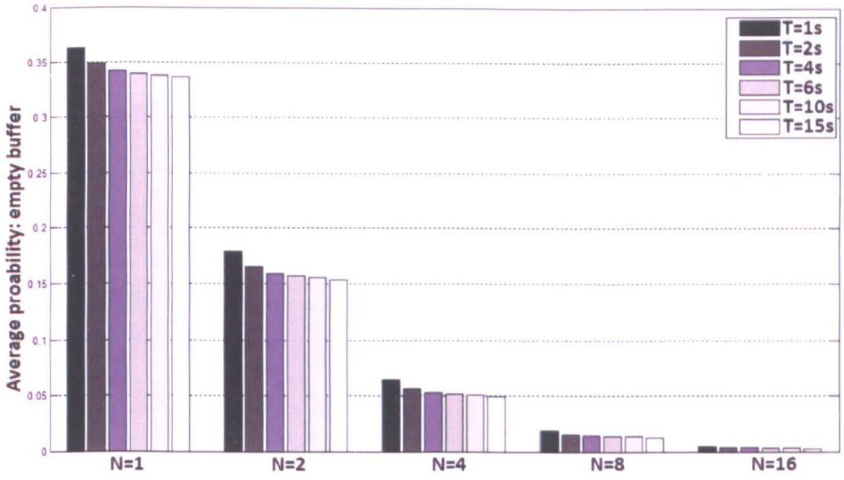


Fig. 4.6 Average probability of empty buffer.

hence a larger client buffer produces better overall performances since it improves the proposed performance metrics. When the buffer size is fixed, the effect of the video segment duration is observed. The results show that small segment durations of the encoded video are preferable for decreasing the probability of buffer overflow at the cost of increased probability of empty buffer. In addition, in this case the probability of active buffer is increased which would result in smooth playout of the buffered segments. However, this provides limitation in the adaptation since the already buffered segments can be with lower quality that the network can support at a particular moment in time. Furthermore, the performance analysis shows that large segment durations are preferable for decreasing the probabilities of empty buffer and active buffer at the cost of increased probability of buffer overflow.

The segmentation of the video has significant effect on the streaming performances [53]. In the analysis, the I-frame position is fixed by restricting the GOP-size of the encoder to the segment size. Therefore, the smaller segment sizes contain high number of I-frames, thus more overall bits within the stream. In order to keep the bitrate of the stream corresponding to the particular representation, the average PSNR for the small segments decreases as reported in [54].

4.1.3 Study summary

This analysis proposes an approach for mapping the functionality of the standard MPEG-DASH model into a segment-based teletraffic model, for which appropriate probability metrics (i.e., probabilities of buffer overflow, empty buffer, and active buffer) are proposed and investigated for different video segment durations at the DASH server and different

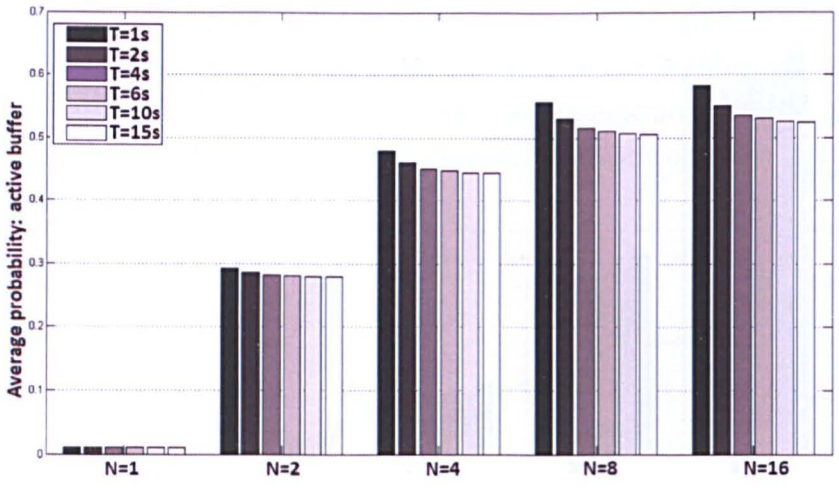


Fig. 4.7 Average probability of active buffer.

buffer sizes at the DASH client. Small segment durations of the encoded video increase the probabilities of empty buffer and active buffer, while decreasing the probability of buffer overflow, whereas large segment durations are suitable for decreasing the probabilities of empty buffer and active buffer, while increasing the probability of buffer overflow. Furthermore, as expected, the performance metrics are improved when the buffer size at the client is increased.

DASH streaming in the context of protocol utilization is based on HTTP/TCP, where the impairments in the video streaming occur during large round trip times of the TCP packets and due to retransmissions when packets are lost, which influence the network throughput and hence the selected video representation from the DASH server. This analysis shows that by increasing the size of the advertised TCP window (i.e., larger buffer size) at the client and selecting appropriate segment durations the overall performances of the streaming can be improved. In the end, it is outlined that different content preparation/segmentation techniques should be adopted in parallel with streaming adaptation in order to further enhance the quality of the video stream.

4.2 Statistical analysis

This analysis is based on a methodology for statistical investigation of DASH systems and it is used to analyse the rebuffering events and the quality through rebuffering and video quality parameters which are relevant for the overall QoE [104]. In particular, the rebuffering events are investigated through the rebuffering frequency and the average rebuffering duration. The rebuffering frequency shows the percentage of DASH segments affected by a rebuffering

event, whereas the average rebuffering duration shows the expected duration of the occurred rebuffering event. The total rebuffering duration during the streaming is derived from the parameters above. The investigated video quality parameters are the video bitrate and the PSNR of the streamed frames at different video resolutions. These are investigated under fixed adaptation logic at the client during lossless/lossy TCP transmissions with different segment sizes. In the end, a transvariation-based approach is also used for comparing the streaming systems with respect to the segment size.

The proposed methodology is elaborated in Subsection 4.2.1. In Subsection 4.2.2, the system model and the performance metrics are presented. The metrics variability and dependencies are then investigated via extensive simulations. The statistical analysis of the simulation data (simulation dataset) is presented in Subsection 4.2.3, where the statistics of the performance metrics (statistical dataset) is shown. Subsection 4.2.4 presents the transvariation-based approach for comparison of the streaming systems. The study summary is presented in Subsection 4.2.5.

4.2.1 Methodology

The methodology workflow is depicted in Fig. 4.8. A simulation study of an adaptive video streaming system is used to obtain a simulation dataset. Further, the simulation dataset is statistically analysed and a statistical dataset is obtained. In the end, the statistical dataset is used to conduct transvariation-based system analysis.

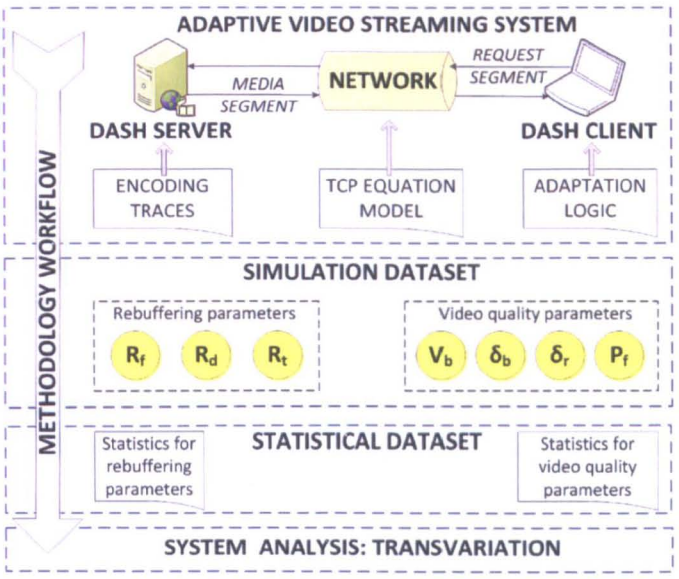


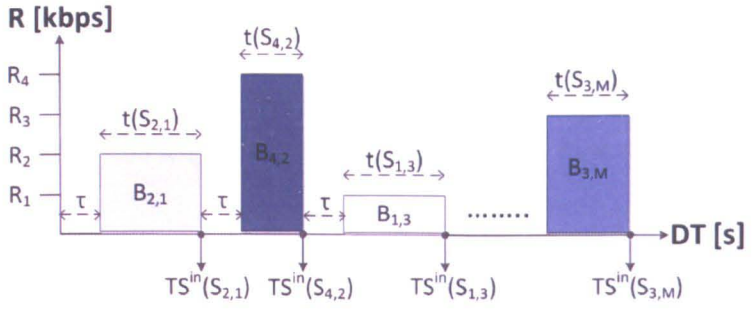
Fig. 4.8 Methodology workflow.

The adaptive video streaming system model is based on the MPEG-DASH standard, which includes a server for storing the encoded media segments, a network model based on TCP transmissions, and a client that controls the adaptation based on a specific adaptation logic. The client requests segments and the server responds by sending the required media segments via the network. Information from the encoding traces is used at the server, an analytical model for the lossless/lossy TCP transmissions in the network, and a predefined adaptation logic at the client. The implementation of the adaptive streaming system model and the simulations are performed in Matlab. As a result of the simulations, a simulation dataset that contains rebuffering parameters and video quality parameters is obtained. The following rebuffering parameters are considered: rebuffering frequency R_f, average rebuffering duration R_d, total rebuffering duration R_t; and the following video quality parameters are considered: video bitrate V_b, percentage of time during which the video is shown at a particular bitrate δ_b, percentage of time during which the video is shown at a particular video resolution δ_r, and PSNR of the frames streamed at the client P_f. The simulation dataset is statistically analysed in order to show how the statistics of the rebuffering and video quality parameters depend on the segment size in lossless and lossy networks. This statistical data comprises a statistical dataset that is further used to provide transvariation-based system analysis. The statistical data analysis and the system analysis are obtained from the statistical package R [11].

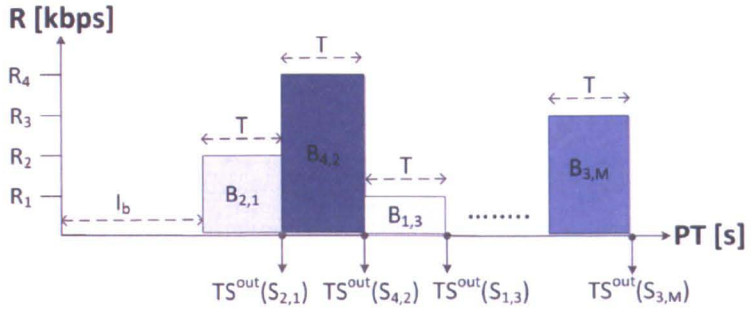
4.2.2 System model and simulations

The adaptive streaming system is based on a client-server model, where the client downloads and plays a video sequence with duration T_v as a composition of M consecutive segments with size T. The modeling of the download and the playout of the video at the client is depicted in Fig. 4.9. In the download timeline (DT), the transmission time of the HTTP GET requests to the server is fixed at τ, whereas the download time of the segments is denoted as $t(S_{j,i})$, where $S_{j,i}$ denotes a segment with $B_{j,i}$ bytes of media data corresponding to the j-th representation (in the depicted example $j = 1,2,..,J; J = 4$) and with ordinal number i (in the depicted example $i = 1,2,3,..,M$). The events in the DT are observed through timestamps for each downloaded segment, denoted as $\{T^{\text{in}}(S_{j,i}), i = 1,2,..,M\}$. In the playout timeline (PT) there is an initial buffering period denoted as I_b followed by a playout of the downloaded segments with size T. The playout is interrupted when a rebuffering event occurs. The events in the PT are observed through timestamps for each segment playout denoted as $\{T^{\text{out}}(S_{j,i}), i = 1,2,..,M\}$.

Prior to the start of the streaming, the timestamps for the PT are constructed based on the metadata information in the MPD file. These timestamps indicate the time instants where the playout of each segment is completed. The timestamps for the DT are obtained as a result



(a) Download timeline.



(b) Playout timeline.

Fig. 4.9 Download and playout timelines at the client side.

of the segment downloads, where the selection of the representation, i.e., the adaptation logic by the client, is tailored to preserve the following condition for every video segment: $\{T^{\text{out}}(S_{j,i}), i = 1,2,..,M\} \geq \{T^{\text{in}}(S_{j,i}), i = 1,2,..,M\}$. Hence, the client selects segments from the highest representation j that complies with:

$$\begin{cases} I_b + T \geq t(S_{j,1}) + \tau; \\ I_b + 2T \geq t(S_{j,1}) + t(S_{j,2}) + 2\tau; \\ . \\ . \\ I_b + MT \geq \sum_{k=1}^{M} t(S_{j,k}) + M\tau. \end{cases} \tag{4.15}$$

When a representation is selected for a particular video segment, then the segment data which is located on a URL resource in the network is delivered as a response to the previously sent HTTP-GET request. This response is organized as consecutive Protocol Data Units (PDUs) that are reliably transmitted utilizing the HTTP/TCP protocols in forms of TCP packets. The content from each video segment $S_{j,i}$ with size T is mapped into a number $X_{j,i}$

of TCP packets containing $B_{j,i}$ bytes. The used packet size for a TCP packet is MSS=1400 bytes, hence the number of TCP packets for the delivery of a video segment is $X_{j,i} = \lceil \frac{B_{j,i}}{MSS} \rceil$.

The download time $t(S_{j,i})$ for segment $S_{j,i}$ depends on the number of TCP packets $X_{j,i}$ of TCP packets and the experienced TCP network throughput $\sigma(p)$, according to:

$$t(S_{j,i}) = \frac{X_{j,i}}{\sigma(p)}. \tag{4.16}$$

The TCP throughput $\sigma(p)$, expressed in packets/s, is modeled as a function of the packet loss probability p in the network [119] as used in the previous chapter, where for $p \neq 0$:

$$\sigma(p) = \frac{1}{RTT\sqrt{\frac{2bp}{3}} + T_0 min(1, 3\sqrt{\frac{3bp}{8}})p(1+32p^2)}, \tag{4.17}$$

where RTT is the round trip time in the network, T_0 is the waiting time for the re-transmission of delayed or lost TCP packets, b is the number of acknowledged packets (TCP implementations usually send one ACK for two received packets, hence $b = 2$), W_{max} is the advertised maximum TCP window size, which represents the number of packets that can be accepted without acknowledging the server. The server uses a congestion avoidance mechanism while incrementing the window size until W_{max}, at which the window size remains fixed until occurrence of three duplicate packets or a time-out event. In the case of no packet loss in the network ($p = 0$), the TCP throughput is determined only by the maximum window size and the RTT in the network:

$$\sigma(p = 0) = \frac{W_{max}}{RTT}. \tag{4.18}$$

The throughput variations are simulated by setting an average round trip time $RTT_{av} =$ 150 ms, and uniformly distributed packet delay variations in the interval (100 ms - 200 ms). In addition, the packet loss occurrence in the network is captured by setting the packet loss probability $p = 0.1\%$. Hence, the throughput is variable according to (4.17), and the flow control is performed by advertising the maximum window size $W_{max} = 100$ packets.

The adaptive video streaming system is implemented in Matlab, where the streaming simulations are performed with a sufficiently long video sequence (*Big Buck Bunny*) and three segment sizes ($T = 1s$, $T = 4s$, $T = 10s$). The number of simulations was set to 500 in order to provide sufficient data for the statistical analysis. As a results of the simulations, a simulation dataset is obtained, composed of the following rebuffering parameters:

- Rebuffering frequency (R_f),
- Average rebuffering duration (R_d),

- Total rebuffering duration (R_t),

and the following video quality parameters:

- Video bitrate (V_b),
- Percentage of time during which the video is shown at a particular bitrate (δ_b),
- Percentage of time during which the video is shown at a particular video resolution (δ_r),
- PSNR of the frames streamed at the client frames (P_f).

The parameters are observed for the considered segment sizes in lossless and lossy TCP networks and are defined in the following subsections.

4.2.2.1 Rebuffering parameters

The rebuffering frequency is calculated by tracking the occurrence of rebuffering events within the streamed segments by using the following function:

$$r_f(S_{j,i}) = \begin{cases} 1, & \text{if } T^{\text{out}}(S_{j,i}) < T^{\text{in}}(S_{j,i}) \\ 0, & \text{if } T^{\text{out}}(S_{j,i}) > T^{\text{in}}(S_{j,i}) \end{cases} \tag{4.19}$$

which tracks the difference between the timestamp sequences in the download and the playout timeline, and it is used to calculate the rebuffering frequency according to:

$$R_f = \frac{1}{M} \sum_{i=1}^{M} r_f(S_{j,i}). \tag{4.20}$$

By definition, the rebuffering frequency is in the interval [0,1], where 0 indicates that there is no rebuffering during the streaming of all video segments, whereas 1 indicates that rebuffering occurred in every video segment. Therefore, the rebuffering frequency shows the percentage of segments affected by rebuffering during the video streaming.

The average rebuffering duration R_d is calculated by using the function $r_d(S_{j,i})$, which is expressed as:

$$r_d(S_{j,i}) = \begin{cases} \delta(S_{j,i}), & \text{if } T^{\text{out}}(S_{j,i}) < T^{\text{in}}(S_{j,i}) \\ 0, & \text{if } T^{\text{out}}(S_{j,i}) > T^{\text{in}}(S_{j,i}) \end{cases} \tag{4.21}$$

where $\delta(S_{j,i}) = T^{\text{in}}(S_{j,i}) - T^{\text{out}}(S_{j,i})$ is the time difference between the timestamp sequences for the segments. By averaging over the segments affected by the rebuffering, the average duration of the rebuffering is obtained:

$$R_d = \frac{1}{M_{fr}} \sum_{i=1}^{M} r_d(S_{j,i}). \tag{4.22}$$

where M_{fr} is the number of segments affected by rebuffering.

The defined metrics show the percentage of segments affected by rebuffering and the average duration of this rebuffering at segment level. Therefore, if during the streaming of 100 video segments, the values $R_f = 0.1$ and $R_d = 50$ ms are obtained, on average, 10 segments are affected by rebuffering, and the average duration of the event is 50 ms, which results in additional 500 ms viewing time with respect to the duration of the video T_v. Hence, the total rebuffering duration R_t is calculated as a product of the rebuffering frequency and the average rebuffering duration:

$$R_t = R_f R_d. \tag{4.23}$$

4.2.2.2 Video quality parameters

The video quality parameters are related to the representation selection. In the simulations, the set of representations available at [54] is used and the percentage of streaming time during which the video is shown at a particular bitrate (δ_b) and at a particular resolution (δ_r) is observed, since multiple representations encoded at different bitrates might have the same video resolution. At the end of the simulations, the video bitrate (V_b) is obtained as the average of the bitrates for the downloaded segments, and the PSNR of the streamed frames (P_f) is obtained as an average of the PSNR of all frames for the downloaded segments.

4.2.3 Statistical data analysis

The statistical analysis of the simulation dataset is used to obtain a statistical dataset. The statistical dataset contains the summary statistics of the parameters in the simulation dataset. Since three segment sizes in a lossless and lossy network are considered, a total of 6 cases are obtained indexed with $K1, K2, .., K6$ for the statistical dataset. For the lossless network, $K1, K2$, and $K3$ are used to refer to the streaming cases with segment sizes $T = 1s, T = 4s$, and $T = 10s$, respectively. For the lossy network $K4, K5$, and $K6$ are used to refer to the streaming cases with segment sizes $T = 1s, T = 4s$, and $T = 10s$, respectively. The summary statistics of the parameters are presented in the following subsections.

4.2.3.1 Rebuffering frequency

Table 4.1 shows the summary statistics for the rebuffering frequency. The column of the averages represents the estimated probability that a rebuffering event occurs in a segment. It should be noted that this probability decreases for smaller segment sizes since adaptation is performed more efficiently when using smaller segment sizes. Furthermore, a significant effect on this probability of the network is observed when the segment size is fixed. In the lossy network, a rebuffering event is 6.5%, 8.0% and 8.2% more likely (likelyhood increases in terms of percentage points) to occur with segment size $T = 1s, T = 4s$, and $T = 10s$, compared to the lossless network, respectively. Since the probability of packet loss in the lossy network is $p = 0.1\%$, hence even small packet losses significantly affect the rebuffering frequency during the video streaming.

For a visual comparison of the empirical distributions simulated in the six cases, the boxplots are shown in Fig. 4.10. The boxplot, or "box-and-whisker" diagram, depicts the data through quartiles. The box is delimited by the first and third quartile. The second quartile (the median) is shown by the horizontal line inside the box. While the box represents the 50% central observations, the whiskers are designed for describing the tails of the distribution and visualizing the most extreme data as individual points. Thus, the end of the whiskers is the more extreme observation that is still less than 1.5 times the interquartile distance from the box, while the outlying data points are represented by the individual circles.

The $T = 1s$ case has most of the times performed better in terms of the rebuffering frequency in both the lossless network and the lossy network case. On the other hand, there is more overlapping between the boxplots of the $T = 4s$ and $T = 10s$ cases, and a reduced variability of the $T = 1s$ case compared to the other two. Finally, the lossy network clearly deteriorates the performances by producing a shifting effect on the relative boxplots (a 50%-60% increase in the average rebuffering frequency).

4.2.3.2 Average rebuffering duration

Table 4.2 shows the summary statistics for the rebuffering duration. It can be observed that both the average and the variability of the rebuffering duration increase as the segment size is increased in both lossy and lossless networks. On average, for the lossless network, the rebuffering duration for $T = 1s$ is 310 ms shorter compared to the $T = 4s$ case, which in turn results in a 630 ms shorter rebuffering duration compared to the $T = 10s$ case. In the lossy network, the differences are higher, i.e., the $T = 1s$ case results in a 421 ms shorter rebuffering duration compared to the $T = 4s$ case, which results in a 819 ms shorter rebuffering duration compared to the $T = 10s$ case.

The boxplots of the simulated rebuffering durations are shown in Fig. 4.11. In the lossless network, the range of rebuffering duration values for $T = 1s$ does not overlap with the one of $T = 4s$, therefore selecting $T = 1s$ always (over 500 simulations) produces lower rebuffering duration compared to $T = 4s$. However, only in a single case (over 500 simulations), the rebuffering duration for the $T = 10s$ case falls in the range of the $T = 1s$ case. Further, the overlapping between $T = 4s$ and $T = 10s$ cases is minimal. These differences are slightly more notable in the lossy network, where $T = 1s$ always produces lower rebuffering duration compared to $T = 4s$ and $T = 10s$, and the overlapping between $T = 4s$ and $T = 10s$ cases is reduced compared to the lossless network. Therefore, the impact of the segment size is more notable in the lossy network and the small packet loss has also a significant impact on the rebuffering duration metric.

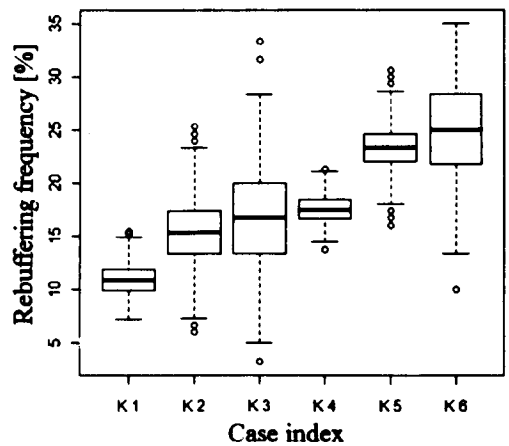


Fig. 4.10 Boxplots for the rebuffering frequency.

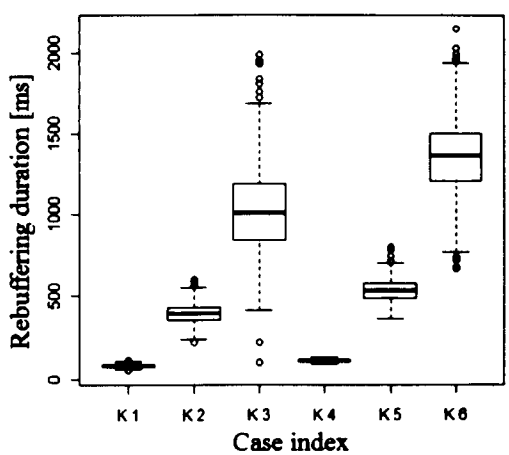


Fig. 4.11 Boxplots for the average rebuffering durations where averaging is done over time.

Table 4.1 Rebuffering frequency: summary statistics

Network	Segment size [s]	Index	Min [%]	1st Q. [%]	Median [%]	Mean [%]	3rd Q. [%]	Max [%]	# segments
Lossless	T=1	K1	7.2	9.9	10.9	10.9	11.9	15.4	597
	T=4	K2	6.0	13.3	15.3	15.4	17.3	25.3	150
	T=10	K3	3.3	13.3	16.7	16.6	20.0	33.3	60
Lossy	T=1	K4	13.7	16.6	17.4	17.5	18.4	21.3	597
	T=4	K5	16.0	22.0	23.3	23.4	24.7	30.7	150
	T=10	K6	10.0	21.7	25.0	24.8	28.3	35.0	60

Table 4.2 Average rebuffering duration: summary statistics

Network	Segment size [s]	Index	Min [ms]	1st Q. [ms]	Median [ms]	Mean [ms]	3rd Q. [ms]	Max [ms]	# segments
Lossless	T=1	K1	58	76	82	82	88	108	597
	T=4	K2	218	348	390	392	430	595	150
	T=10	K3	98	838	1003	1022	1189	1985	60
Lossy	T=1	K4	90	105	111	111	116	131	597
	T=4	K5	355	485	531	532	574	797	150
	T=10	K6	665	1207	1360	1351	1501	2147	60

4.2.3.3 Total rebuffering duration

In this subsection, the total rebuffering duration is investigated. The summary statistics of the total rebuffering duration is presented in Table 4.3, and the relative boxplots are depicted in Fig. 4.12. The results in Table 4.3 show that, on average, the total rebuffering duration increases for larger segment sizes in both lossless and lossy networks. In the lossless network, selecting $T = 4s$ instead of $T = 1s$, and selecting $T = 10s$ instead of $T = 4s$, results in 3730 ms and 1025 ms larger total rebuffering duration. Therefore, the difference between selecting $T = 4s$ instead of $T = 1s$ has a more significant impact on the streaming compared to the difference between selecting $T = 10s$ instead of $T = 4s$. In the lossy network, these differences are significantly more notable, i.e., selecting $T = 4s$ instead of $T = 1s$, and selecting $T = 10s$ instead of $T = 4s$, results in 7150 ms and 1430 ms larger total rebuffering duration.

The comparison between the relative boxplots for the rebuffering frequency, average rebuffering duration, and the total rebuffering duration show that the impact of the rebuffering frequency on the total rebuffering duration is more significant compared to the impact of the rebuffering duration. This is due to the similar trends observed between the rebuffering frequency and total rebuffering duration boxplots in terms of overlapping between the different cases. The total rebuffering duration metric is useful to describe the continuity of the streaming process. The observations show that this metric is highly dependent on the selected segment sizes and the network.

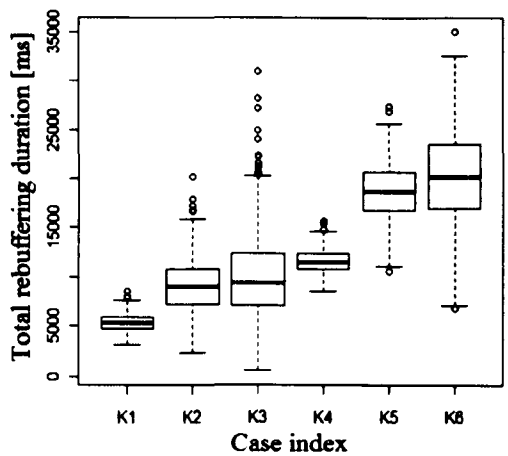


Fig. 4.12 Boxplots for the total rebuffering duration.

4.2.3.4 Video quality parameters

The summary statistics of the video quality parameters for the considered cases is shown in Table 4.4. In the case of the lossless network two representations are used during adaptation, whereas in the lossy network four representations are used. Furthermore, in the case of lossless network, the video is shown in a higher resolution for higher percentage of the streaming time. As expected, both the video bitrate and the PSNR are significantly higher in the case of lossless network. The significant difference in these parameters is due to the fact that in the lossless network the throughput is constrained only by the round trip time and the congestion window size, whereas in the lossy network the throughput is additionally constrained by the packet loss which results in retransmissions. It should be noted that in both lossless and lossy networks, the segment size has an insignificant effect on the video bitrate and the resolution. The reason for this is twofold. First, the used equation-based throughput model produces slow variations in the considered setup, and second, the previously described adaptation logic only considers the throughput and it is buffer unaware. However, the observation shows that the impact of the segment size on the PSNR, where using larger segment sizes is preferable since it slightly improves the PSNR in both lossless and lossy network. This benefit is obtained only due to the encoding of the segments.

4.2.4 Comparative system analysis

In this section, a comparative system analysis is presented by investigating how the segment size affects the system performance by considering transvariation. The statistical data analysis showed that the rebuffering frequency and the average rebuffering duration, hence the total rebuffering duration are significantly affected by the segment size, whereas the video quality metrics remain as reported earlier. Therefore, the focus is on system analysis only on the total rebuffering duration during the streaming. An approach based on probability graphs is used as a generalized version of the concept of transvariation.

The transvariation probability was introduced by Gini [138] as a similarity measure between distributions. A pair of observations of the random variables X and Y are said to transvary when the sign of the difference $X - Y$ is opposite to the sign of their mean difference $E(X) - E(Y)$. For example, in the case of lossless network, selecting $T = 4s$ results, on average, in a longer rebuffering duration compared to the $T = 1s$ case, therefore the transvariation probability is expressed as $P(X - Y < 0)$, where X, Y correspond to the $K1, K2$ cases, respectively. Since the transvariation measures the similarity between the distributions, the complementary probability $P(Y - X > 0)$ is a measure of separation (cis-variation). Henceforth, the random variables $\{D1, .., D6\}$ are used to denote the total

Table 4.3 Total rebuffering duration: summary statistics

Network	Segment size [s]	Index	Min [ms]	1st Q. [ms]	Median [ms]	Mean [ms]	3rd Q. [ms]	Max [ms]	# segments
Lossless	T=1	K1	3128	4762	5354	5345	5933	8505	597
	T=4	K2	2314	7221	8996	9075	10810	20210	150
	T=10	K3	587	7076	9394	10100	12360	31040	60
Lossy	T=1	K4	8491	10770	11490	11540	12340	15670	597
	T=4	K5	10580	16770	18770	18690	20670	27410	150
	T=10	K6	6842	16950	20210	20120	23620	35090	60

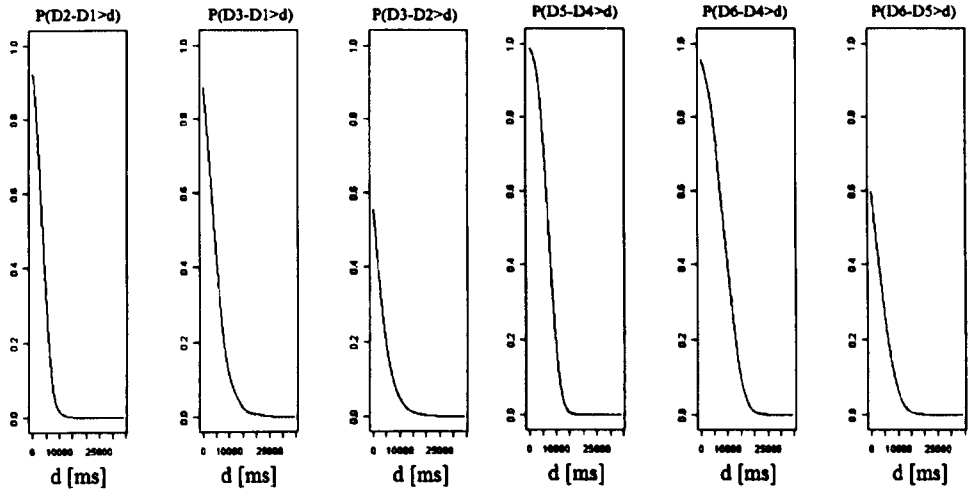
Table 4.4 Content quality: summary statistics

Network	Segment size [s]	Index	Representation				V_b [Mbps]	P_f [dB]
			Bitrates [Mbps]	δ_b [%]	Resolutions	δ_r [%]		
Lossless	T=1	K1	{6, 8}	{32, 68}	{1920x1080}	{100}	7.3	42.8
	T=4	K2	{6, 8}	{31, 69}	{1920x1080}	{100}	7.3	43.0
	T=10	K3	{6, 8}	{31, 69}	{1920x1080}	{100}	7.3	43.9
Lossy	T=1	K4	{1.5, 2.0, 2.5, 3}	{29, 39, 30, 2}	{1280x720, 1920x1080}	{68, 32}	2.0	38.10
	T=4	K5	{1.5, 2.0, 2.5, 3}	{27, 39, 32, 2}	{1280x720, 1920x1080}	{66, 34}	2.0	38.66
	T=4	K5	{1.5, 2.0, 2.5, 3}	{26, 39, 33, 2}	{1280x720, 1920x1080}	{65, 35}	2.0	38.88

rebuffering duration in the considered cases, which are obtained based on the probabilities related to the rebuffering frequency and average rebuffering duration.

In order to understand the practical impact of the separation, the generalized cis-variation probabilities $P_d = P(D2 - D1 > d)$ are estimated on regular intervals between 0 and 35 seconds, i.e., $d = 0, 100, .., 35000$ (ms). For example, if the $T = 1s$ and $T = 4s$ cases in the lossless network $(K1, K2)$ are considered, then the aforementioned expression for P_d is the probability that the total rebuffering duration of the video streaming for the $T = 4s$ case has an excess of d ms with respect to the $T = 1s$ case.

In Fig. 4.13, these probabilities are plotted against d in order to assess the total rebuffering duration for the considered segment sizes in both lossless and lossy networks. Based on the plots, the following remarks are summarized.



(a) Lossless network model (b) Lossy network model ($p = 0.1\%$)

Fig. 4.13 Comparison of segment sizes.

In the case of lossless network, comparing the $T = 1s$ and $T = 4s$ cases shows that there is a 31%, 2% and 0% probability that the total rebuffering duration of the video streaming is larger than 5 s, 10 s, 15 s, respectively, when selecting $T = 4s$ instead of $T = 1s$. Comparing the $T = 4s$ and $T = 10s$ cases, shows that there is a 19%, 5%, and 1% probability that the total rebuffering duration is larger than 5 s, 10 s, 15 s, respectively, when selecting $T = 10s$ instead of $T = 4s$. Comparing $T = 1s$ and $T = 10s$ results in even higher separation, i.e., the probabilities that the total rebuffering duration is larger than 5 s, 10 s, 15, are 42%, 11%, and 3%, when selecting $T = 10s$ instead $T = 1s$. In the lossless network, selecting smaller segment sizes improves streaming performances by combating the effect of rebuffering, and thus reducing the total rebuffering duration.

In the case of lossy network, the separations are more notable. Therefore, selecting smaller segment sizes in a lossy network results in more notable improvements. The observations show that there is a 75%, 18% and 1% probability that the total rebuffering duration of the video streaming is larger than 5 s, 10 s, 15 s, respectively, when selecting $T = 4s$ instead of $T = 1s$. By comparing the $T = 4s$ and $T = 10s$ cases, there is a 27%, 6%, and 1% probability that the total rebuffering duration is larger than 5 s, 10 s, 15 s, respectively, when selecting $T = 10s$ instead of $T = 4s$. In the end, the comparison between $T = 1s$ and $T = 10s$ shows the probabilities that the total rebuffering duration is larger than 5 s, 10 s, 15, are 77%, 39%, and 10%, when selecting $T = 10s$ instead of $T = 1s$.

4.2.5 Concluding remarks

The statistical analysis of rebuffering events and video quality for DASH-based video streaming systems shows how the segment size impacts the statistics of performance metrics related to the rebuffering events and the video quality. The rebuffering events are investigated through the rebuffering frequency and the average rebuffering duration, which result in total rebuffering duration, whereas the video quality is investigated through the bitrate and the PSNR of the streamed frames at different video resolutions. These performance metrics are investigated under fixed adaptation logic at the client during lossless/lossy TCP transmissions in the network.

In the proposed setup, the following conclusions are derived: 1) smaller segment sizes improve the rebuffering frequency and the average rebuffering duration regardless of the network, however these improvements are significantly more notable in the lossy network; 2) the rebuffering frequency has a higher impact on the total rebuffering duration compared to the average rebuffering duration regardless of the segment size and the network; 3) the video quality metrics are not impacted by the segment size (except the PSNR of the streamed frames due to the encoding of the segments), however these metrics are impacted by the network; and 4) the transvariation-based approach used for the system analysis shows the extent at which a smaller segment size improves the total rebuffering duration as the most significant metric impacted by the segment size.

4.3 Chapter summary and contributions

The chapter on teletraffic and statistical analysis of DASH systems presents probabilistic and statistical assessment of broad range of parameters related to the DASH protocol stack and to the QoE of DASH video streaming. These parameters are investigated with respect

to different segment sizes under single predefined adaptation logic. The models and the results in these analysis are obtained by using traces from real DASH streaming, packages for statistical analysis and standalone numerical simulations. Further, the analytical part of the analysis is based on fundamental principles from queuing and statistical theory. As a results of the work presented in this chapter, the following presents a summary of the contributions and works:

- A novel segment-based teletraffic model for the MPEG-DASH standard. This is a novel model that maps the MPEG-DASH functionality in teletraffic domain by using a mathematical approach based on queuing theory. Based on the model, a relevant probability metrics are derived and investigated for different MPEG-DASH technical parameters, i.e., the segment size of the encoded video at the server and the buffer size at the client. Furthermore, the results of this analysis are used to provide information on how to configure the HTTP/TCP communication protocol stack so that the streaming performance is improved.

- Methodology for statistical analysis of DASH systems. This work is based on a methodology for the evaluation of parameters related to the QoE of DASH video streaming by applying fundamental principles from statistics. Using the proposed methodology, relevant statistics for rebuffering and video quality parameters are obtained and a transvariation-based approach is proposed for comparative analysis with respect to the segment sizes. The results show how the considered parameters are affected by the segment size.

Chapter 5

Quality of Service and Quality of Experience for DASH

The QoS refers to the totality of characteristics of a service that bear on its ability to satisfy stated and implied needs of the user of the service and it comprises quantitatively measurable network performances such as error rates, bit rate, throughput, transmission delay, availability, jitter, etc [139]. In contrast to the QoS, the QoE is a strictly user-oriented paradigm that refers to the overall acceptability of a service or application as perceived subjectively by the end user [140]. This chapter addresses aspects related to the QoS and QoE for DASH. A study on the QoS and QoE for DASH video streaming presents two-fold contribution. First, an analysis on how the encoding and the segmentation of the DASH dataset affects the QoS and QoE; and second, a model and a novel method for estimating the MOS of DASH video streaming based on a MOS model for video streaming at a constant video bitrate and by considering the average bitrate, the variance of the switching between the DASH representations, and the rebuffering frequency. In addition, a novel content-aware quality metric is proposed in this chapter. This metric is derived in a more general setup compared to the one considered in the study on QoS and QoE. In contrast to the model proposed in the QoS and QoE study, the content-aware quality metric combines a broader range of QoS parameters and additionaly considers content related parameters.

This chapter is organized as follows. A study on the QoS and QoE for DASH video streaming is presented in Section 5.1, whereas a novel content-aware quality metric is proposed, formulated and evaluated in Section 5.2. Section 5.3 concludes the chapter.

5.1 A study on QoS and QoE for DASH

The contributions provided in the study on QoS and QoE for video streaming in DASH systems show how the encoding and segmentation of the DASH dataset affect the QoS and QoE of the video streaming. This includes the GOP size and segment size impact on the PSNR of the frames, the video throughput, the rebuffering frequency, and the MOS of the streaming session. Further, in this study, a model and a novel method for modeling the MOS of DASH video streaming are proposed. These are based on a MOS model for video streaming at a constant video bitrate and by considering the average bitrate, the variance of the representation switching, and the rebuffering frequency. The contributions in the study are provided for different GOP sizes and segment sizes used in the encoding and segmentation of the DASH dataset.

This study is organized as follows. Subsection 5.1.1 presents the related work. The system model is presented in Subsection 5.1.2, in which the encoding and segmentation of the DASH dataset, the adaptation strategy at the client and the details for the simulations and the subjective tests are presented. The results from the simulations and subjective tests are reported in Subsection 5.1.3. The QoE model and the novel method for estimating the MOS of the DASH streaming session are elaborated and validated in Subsection 5.1.4.

5.1.1 Related work

Related works address content encoding and segmentation of the DASH dataset and QoS/QoE studies. The DASH dataset is generated by encoding and segmentation of the video sequence, where the sequence is encoded in multiple bit rates by using predefined GOP sizes (expressed in seconds) and segmented in independent units (segments) by using predefined segment sizes (expressed in seconds). The bit rates are used to define the DASH representations, and during the video streaming the client uses an adaptation strategy to select a segment from the appropriate DASH representations. Hence, segments with variable quality are played out at the client side, and the main focus of the works on QoS and QoE is the dependence of quality on the system parameters.

Several content encoding and segmentation studies were discussed in Chapter 2, Section 2.2.3.1. In particular the works in [53–56] are tightly related to the content encoding and segmentation performed in this study. In contrast to [53], different segment sizes and different GOP sizes with respect to the adopted segment sizes are investigated, where the GOP size is fixed during the streaming. Further, these encoding and segmentation approaches are investigated not only with respect to the impact on the PSNR of the streamed frames, but also in terms of the average video throughput, rebuffering frequency of the frames, and the actual

MOS of the streaming session. Considering the work in [54], in this study the same encoding tools is used and similar trends regarding the impact of encoding and segmentation on the average PSNR of the frames that compose the segments are observed. However, in [54] a fixed GOP size equal to the segment size is used in the H.264/AVC encoder, whereas in this section the effect of smaller GOP sizes with respect to the segment size is investigated. This is particularly useful when fragmentation is used during the encoding of the segment, thus enabling the client to retrieve a GOP size without the segment being fully encoded/prepared. Unlike the work in [55], in this work the GOP size effects are considered and it is observed that the average throughput is not a sufficient indicator of the quality, since for instance the changes in quality with respect to the usage of different DASH representations impacts the quality. In the study, this effect is captured through the variance of the representation switching. In the end, similar remarks are emphasized as in [56], however in this study, the MOS is additionally considered, and it is shown that the MOS of DASH video streaming is improved with smaller segments due to the adaptation. Furthermore, a broader range of performance metrics of the streaming system are investigated and model and method for the QoE of the video streaming are provided.

Notable works on QoS and QoE were discussed in Chapter 2, Section 2.2.3.3. The works in [61, 81–85] are tightly related to the QoS and QoE aspects discussed in this study which considers relevant QoS and QoE parameters as discussed in [141]. The studies [61, 81, 82] emphasize the importance of the rebuffeing frequency as a major parameter that affects the QoE, therefore it is also included in the model and the method proposed in this study. In the study, a model and a method based on MOS model for constant video bitrate streaming and by additionally considering the average bitrate and the variance of the representation switching are proposed. The obtained model shows an exponential relationship similar to the one obtained in [84], and in contrast to the model proposed in [85] it considers the occurrence of rebuffering during the streaming.

5.1.2 System model

The system model used for the simulations comprises the encoding and segmentation of the video content and the adaptation strategy. Based on the video streaming simulations, impaired video sequences are generated and used for the subjective tests.

5.1.2.1 Content encoding and segmentation

Four different videos (a cardiac ultrasound video sequence, in the following "US", and the sequences HARBOUR, CREW, SOCCER) are considered for which DASH datasets with

the following segment sizes: $T = 1, T = 2, T = 4, T = 8$, expressed in seconds (s), and five representation levels (200, 400, 800, 1000, 1500), expressed in (kbps) are generated by using the DASHEncoder [12]. Additionally, the GOP sizes (G) are set to $G = T$, $G = T/2$, and $G = T/4$. The US sequence has resolution 640×416 and duration 32 seconds, whereas the HARBOUR, CREW, and SOCCER sequences are looped to provide a duration of 36 seconds, with the resolution of 704×576 pixels. The encoding frame rate for the sequences is set at 25 frames per second (fps).

5.1.2.2 Adaptation algorithm

Since MPEG-DASH standard does not prescribe a precise logic for adapting/switching between representations, there is a growing research interest in investigating the performance of different adaptation approaches. Several works on adaptation were discussed Chapter 2, Section 5.1.2.2, most notable [62, 65–69]. In this study, the impact of different adaptation strategies on the QoS and QoE is out of the study scope, however based on the aforementioned references, a throughput and buffer based adaptation is used, which is elaborated as follows.

At the client side, an adaptation strategy that sends requests in a sequential fetch and play manner is used by considering the network throughput τ (kbps) at decision instance, the segment size T (s), the buffer state B_{state} expressed as number of buffered frames, which depends on the frame rate N (fps), the current representation level ρ^n (kbps) before the decision, and the available representation set composed of the representation levels $\{\rho^i, i = 1, 2, .., M\}$ (kbps) , where $\rho^i < \rho^{i+1}$ is adopted. The output of the algorithm is the representation level ρ^* selected at the client for the following segment, therefore $\rho^{n+1} = \rho^*$.

The algorithm considers two types of triggers used to monitor the changes in the buffer and in the network, i.e., a buffer trigger (BT) and a throughput trigger (TT). The possible values for the triggers are 0 and 1, where 0 is associated to a decrease of the network throughput and of the number of buffered frames, and 1 is associated to an increase of the network throughput and of the number of buffered frames compared to the current condition. If both triggers have value 1, and the current representation level is $\rho^n = \rho^i$, then the algorithm chooses to switch to a higher representation level ($\rho^* = \rho^{i+1}$), i.e., $\rho^{n+1} = \rho^{i+1}$. If both triggers have value 0, then switching to a lower representation level is performed ($\rho^* = \rho^{i-1}$), i.e., $\rho^{n+1} = \rho^{i-1}$. In the end, if one trigger has value 1, and the other has value 0, then the same representation level is kept ($\rho^* = \rho^i$), i.e., $\rho^{n+1} = \rho^i$.

The triggers are calculated based on the following expressions:

$$BT = \begin{cases} 0, \text{if } B_{\text{state}} \leq NT\frac{\rho^n}{\rho^1} \\ 1, \text{if } B_{\text{state}} > NT\frac{\rho^n}{\rho^1}. \end{cases} \tag{5.1}$$

$$TT = \begin{cases} 0, \text{if } \tau \leq \rho^n \\ 1, \text{if } \tau > \rho^n. \end{cases} \tag{5.2}$$

The buffer trigger is used to monitor the depletion rate of the buffer and it is used to mitigate rebuffering, whereas the throughput trigger is used to adapt the switching according to the variable throughput in the network.

5.1.2.3 Simulation and subjective tests

The simulations were performed in Matlab, where encoding traces generated by the DASHEncoder are used and the described adaptation strategy is deployed. Further, to emulate the adaptive video streaming session, a channel model with a variable bandwidth based on [142] is used. Note that, since in the networking community the term *bandwidth* is frequently used also to indicate the channel capacity, this term is also used in parts of this section.

Simulation results are obtained for the following metrics:

- *PSNR of the streamed frames.* The PSNR of the streamed frames of the adaptive video is obtained at the client.
- *Video bitrate.* Based on the switching between the representation levels, the bitrate of the adaptive video streaming session is considered.
- *Rebuffering frequency.*The percentage of rebuffered frames at the client is considered, where a frame that is not downloaded fully by the time it needs to be played out at the client side is a rebuffered frame.

Based on the simulation results, impaired video sequences are generated and used for subjective tests with 15 subjects, except in the case of the US sequence, where the subjective tests were performed with 3 experts. Therefore, the MOS for the US sequence is shown, however the MOS modeling in the medical case is omitted.

5.1.3 Evaluation results

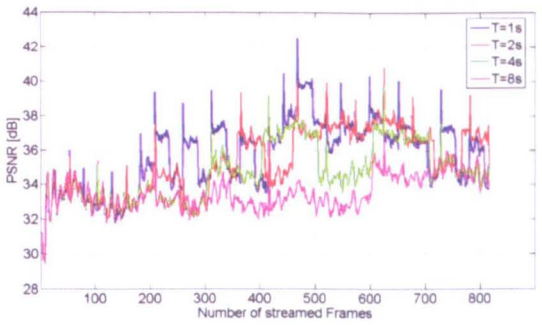
The simulation results and the results from the subjective tests show the impact of different GOP sizes and segment sizes on the QoS and QoE of the DASH video streaming and represent the basis for the QoE modeling.

5.1.3.1 Simulation results

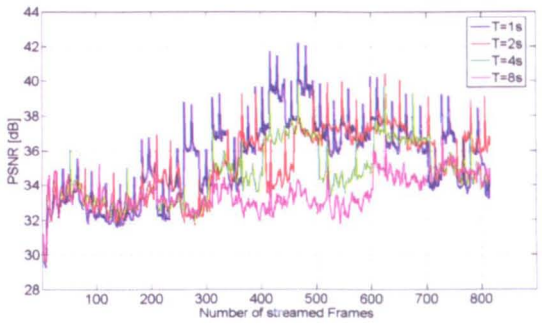
The GOP size and the segment size used in content encoding and segmentation have a significant effect on the PSNR. As shown in [54], smaller segment sizes decrease the PSNR values, and by using GOP sizes smaller than the segment size ($G = T/2, G = T/4$), the overall PSNR is further reduced. This is the case when fragmentation is used during the encoding of the segment since fragmentation enables the client to retrieve content (equal to the GOP size) without the segment being fully encoded/prepared. Fragments are particularly useful in the case of live DASH streaming services, and this is the rationale behind investigating the cases where the GOP size is smaller than the segment size.

The usage of smaller segment sizes increases the number of potential switching points between different representations, thus the adaptation can be performed more efficiently. This leads to improvements in the overall PSNR and bitrate, as shown in Fig. 5.1 for the extreme cases ($G = T$ and $G = T/4$) for the US sequence. Although from the encoding point of view smaller segment sizes decrease the average PSNR of the segment, the higher number of switching points provides better adaptation to the fast bandwidth changes. This allows the client to select representations with higher quality more frequently.

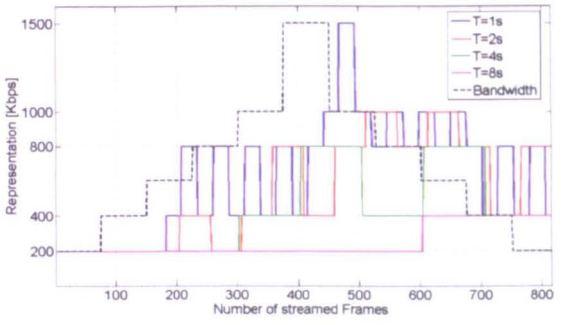
The averages of the PSNR and the video bitrate, as well as the rebuffering frequency of the DASH video streaming of the different video sequences, are shown in Tables 5.1-5.3. The more efficient adaptation with smaller segments provides better performances (higher PSNR and video bitrate and lower rebuffering frequency) when compared to the case with larger segments. Furthermore, it is observed that the ratio between the GOP size and the segment size has insignificant effect on the video bitrate, but it can impact the PSNR. This impact is especially noticeable for large segments (e.g., for the $T = 4s$ case and the SOCCER and CREW sequence, where $G = T/8$ produces 1dB gain compared to $G = T$). The results for the rebuffering frequency show that smaller segment sizes reduce the frequency of rebuffering due to more efficient adaptation to the bandwidth changes. Furthermore, it is observed that a smaller GOP size with respect to the segment size decreases the rebuffering frequency only for large segment sizes.



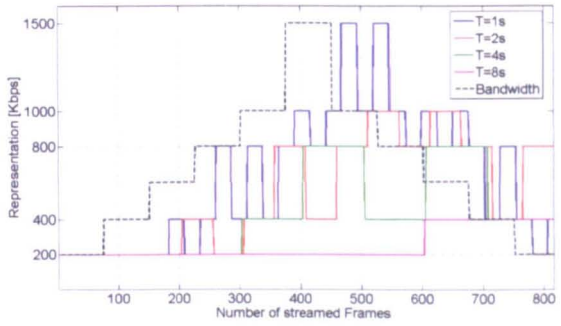
(a) PSNR for $G = T$.



(b) PSNR for $G = T/4$.



(c) Bitrate for $G = T$.



(d) Bitrate for $G = T/4$.

Fig. 5.1 PSNR and bitrate variations: US sequence.

Table 5.1 Average PSNR.

Video	Average PSNR [dB]											
	T=1			*T=2*			*T=4*			*T=8*		
	G=T	*G=T/2*	*G=T/4*	*G=T*	*G=T/2*	*G=T/4*	*G=T*	*G=T/2*	*G=T/4*	*G=T*	*G=T/2*	*G=T/4*
US	35.64	35.45	34.84	35.26	35.12	34.84	34.68	34.61	34.84	33.53	33.50	33.42
Harbour	27.60	26.99	26.22	27.38	27.19	26.22	26.69	26.61	26.22	25.52	25.47	25.38
Crew	31.64	31.45	31.17	31.55	31.23	31.17	**30.14**	30.08	**31.17**	28.142	28.11	28.07
Soccer	31.31	31.01	30.59	30.95	30.86	30.59	**29.55**	29.48	**30.59**	28.73	28.73	28.67

Table 5.2 Average video bitrate.

Video	Average representation level [kbps]											
	T=1			*T=2*			*T=4*			*T=8*		
	G=T	*G=T/2*	*G=T/4*	*G=T*	*G=T/2*	*G=T/4*	*G=T*	*G=T/2*	*G=T/4*	*G=T*	*G=T/2*	*G=T/4*
US	626.2	648.5	629.4	537.5	537.5	537.5	424.7	424.7	424.7	252.2	252.2	252.2
Harbour	616.0	598.6	610.2	524.6	517.3	517.3	402.0	402.0	402.0	244.6	244.6	244.6
Crew	598.6	598.6	592.8	517.3	517.3	517.3	402.0	402.0	402.0	244.6	244.6	244.6
Soccer	598.6	592.8	590.0	517.3	517.3	517.3	334.6	334.6	334.6	244.6	244.6	244.6

Table 5.3 Rebuffering frequency.

Video	Rebuffering frequency [%]											
	T=1			*T=2*			*T=4*			*T=8*		
	G=T	*G=T/2*	*G=T/4*	*G=T*	*G=T/2*	*G=T/4*	*G=T*	*G=T/2*	*G=T/4*	*G=T*	*G=T/2*	*G=T/4*
US	0.85	0.98	1.22	2.45	1.34	0.85	2.45	1.71	0.98	4.16	4.28	2.57
Harbour	0.33	0.55	1.00	0.11	0.22	0.33	1.22	0.44	0.22	2.11	0.77	0.33
Crew	1.66	1.11	1.55	0.33	0.55	0.44	4.33	2.11	1.66	7.44	7.22	4.55
Soccer	1.33	1.55	1.66	2.55	2.66	3.33	3.00	4.88	3.66	6.55	6.22	6.11

5.1.3.2 MOS scores

The results depicted in Fig. 5.2 show the MOS scores from the subjective tests, where similar observations as above can be done on the impact of different GOP sizes and segment sizes on the MOS score. Large segment sizes in the considered setup reduce the MOS due to the inefficient adaptation to the variable network conditions. Hence, higher MOS is obtained for $T = 1s$ and $T = 2s$ regardless of the video sequence, therefore these segment sizes are preferable in the considered setup. Further, it is observed that reducing the GOP size with respect to the segment size decreases the MOS regardless of the segment size and the video sequence. This is mainly due to the effect of reduced PSNR for $G < T$. It should also be noted, that the buffer condition is included in the adaptation approach and larger segment sizes requires larger buffer at the client size. In the considered setup, according to the MOS scores it is preferable to set the GOP size to be equal to the segment size ($G = T$).

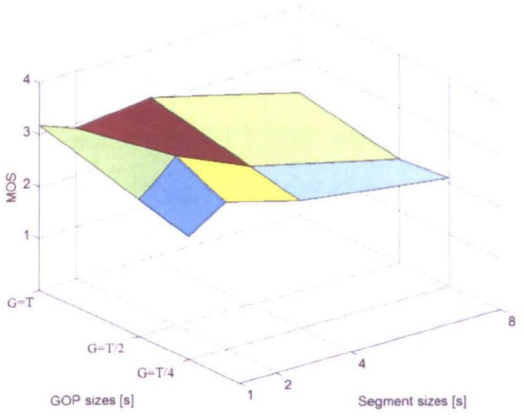
5.1.4 QoE model and proposed method

This section presents further results from subjective tests, a MOS model, and a novel method for obtaining the MOS of the DASH streaming session. Subjective tests were performed to collect MOS scores for video streaming at a constant bitrate and for DASH video streaming. In contrast to the previously reported MOS scores, where the sequences were impaired according to the parameters obtained from the system model, the MOS scores in this section are impaired based on predefined parameters in order to facilitate the modeling. Based on the MOS scores, post processing/fitting is used to obtain models for the MOS of the video streaming at a constant bitrate and for the MOS of the DASH video streaming. These models are then used to propose a novel method for MOS of DASH video streaming under the assumption that the MOS for video streaming at a constant bitrate is known in advance.

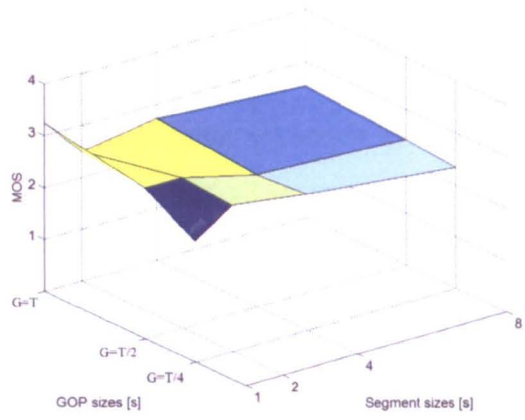
The following remarks are relevant for the models and the proposed method:

- The MOS model for video streaming at a constant bitrate expresses the MOS by considering the bitrate and the rebuffering frequency.

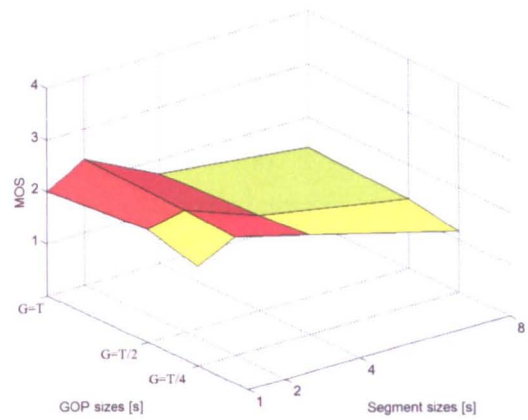
- The MOS model for DASH video streaming expresses the MOS by considering the average video bitrate, the variance of the switching, and the rebuffering frequency. In this case, the average bitrate and the variance of the switching depend on the usage of the DASH representations and are calculated through a switching pattern set which is denoted as $\{R_{sp} : \rho^i, i = 1, 2, .., M\}$. The number of elements in this set depends on the used segment size and the video duration (e.g., if $T = 8s$ and $T_v = 32s$, then the set has 4 elements that correspond to the used DASH representation for each segment).



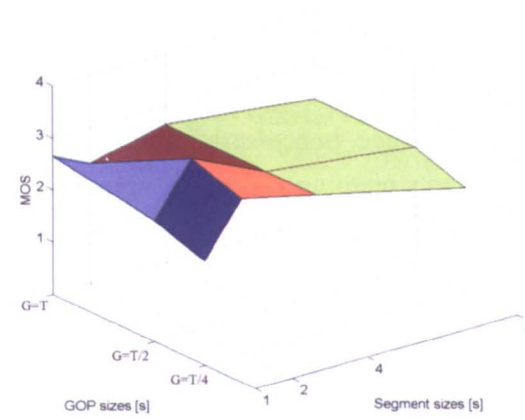
(a) Harbour sequence.



(b) Soccer sequence.



(c) Crew sequence.



(d) US sequence.

Fig. 5.2 MOS for different segment sizes and GOP sizes.

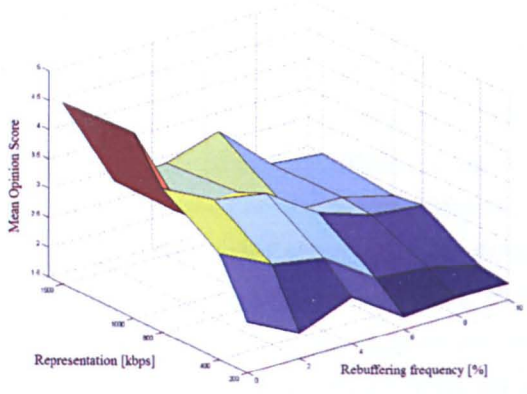


Fig. 5.3 MOS scores for video streaming at a constant bitrate, SOCCER sequence.

- The same rebuffering frequency impairments are used for the derivation of the two models.

5.1.4.1 MOS scores from subjective tests

The subjective tests were organized in two test sessions. In the first session, the MOS scores for video streaming at a constant bitrate were obtained, whereas in the second one the MOS scores for DASH video streaming under two bandwidth models (artificial bandwidth and 3G bandwidth trace, henceforth "AB" and "3G") were obtained so that a broader range of adaptive video streaming sessions is considered.

The bitrates in the video streaming at constant bitrate for the first test session were set to correspond to the available representations used in the DASH video streaming $\{\rho^i, i = 1,2,..,M\}$ for the second test session. Therefore, if $M = 5$, as in Subsection 5.1.2, then five bitrates for the video streaming at a constant bitrate are considered. The same rebuffering impairments are introduced for the video streaming at a constant bitrate, where the rebuffering frequency is denoted as $\{\phi_c^k, k = 0,1,..,5\}$, and for the DASH video streaming, where the rebuffering frequency is denoted as $\{\phi_{\text{dash}}^k, k = 0,1,..,5\}$. The index k corresponds to the rebuffering frequency in the range $[0,10]$, expressed in percentages with 2% step (e.g., $k = 1$ corresponds to rebuffering frequency 2%). Since the rebuffering frequency is kept the same for the two models, then $\phi_c^k = \phi_{\text{dash}}^k = \phi^k$ is used for $k = 0,1,..,5$ in the notations.

The MOS scores for the video streaming at a constant bitrate are shown in Fig. 5.3 (the scores only for the SOCCER video sequence are shown). The MOS scores depend on the bitrate and the rebuffering frequency, reported in x and y axes, hence the scores are reported in the z axis, and are denoted as $Q_c(\phi^k, \rho_c^i)$.

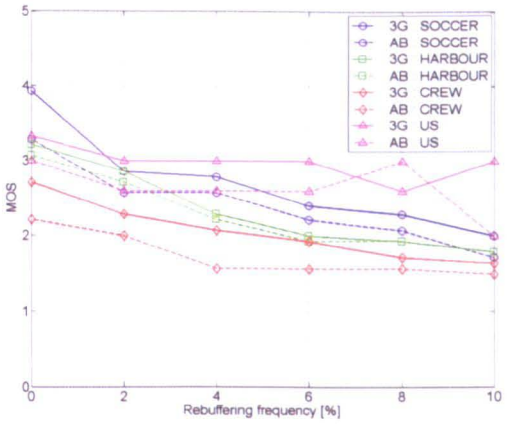


Fig. 5.4 MOS scores for DASH video streaming, all sequences.

The MOS score for DASH video streaming is affected by multiple parameters, but in the modeling the following parameters are considered: the average video bitrate denoted as R_{av}, the variance of the switching denoted as $\hat{\sigma}$ and the rebuffering frequency ϕ^k, hence the scores are denoted as $Q_{\text{dash}}(\phi^k, R_{\text{av}}, \hat{\sigma})$. The MOS scores for the DASH video streaming for all video sequences in the considered rebuffering frequency range are shown in Fig. 5.4. The MOS scores follow a general trend for all the video sequences, where appropriate scaling should be introduced due to content characteristics (an exception to this observation is the case of the "US" sequence, but the medical case is omitted due to smaller number of subjects employed for the tests). Therefore, only the MOS scores for the SOCCER sequence are considered in the modeling as a representative of the sequences that follow the same trend.

5.1.4.2 MOS model

The MOS scores from the subjective tests were processed with the MATLAB curve fitting tool, which was used to obtain the MOS model for video streaming at a constant bitrate and the MOS model for DASH video streaming. The MOS value from the model for the video streaming at a constant bitrate is denoted as $Q_c^m(\phi^k, \rho_c^i)$, whereas the MOS value from the model for the DASH video streaming is denoted as $Q_{\text{dash}}^m(\phi^k, R_{\text{av}}, \hat{\sigma})$.

Fig. 5.5 depicts the MOS scores from the subjective tests and the MOS values from the models. In the case of the video streaming at a constant bitrate, the MOS scores and values are depicted for the bitrate $\rho_c^3 = 800$ kbps. This bitrate is selected as a representative due to the fact that 800 kbps is the first bitrate from the DASH representation set higher than the average bitrates of the DASH video streaming, which are $R_{\text{av}}^{AB} = 598.6$ kbps and $R_{\text{av}}^{3G} = 550.4$ kbps, for the AB bandwidth and the 3G bandwidth, respectively. As expected,

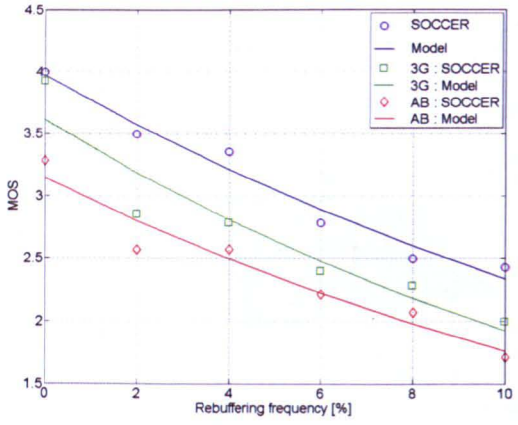


Fig. 5.5 MOS score and MOS model vs. rebuffering frequency

a higher MOS is anticipated and observed in the case of video streaming at a constant 800 kbps bitrate compared to the DASH video streaming cases since the average bitrates of the DASH streaming are lower; additional reason for this is the changes in quality during the DASH video streaming, whereas there is no quality changes during the video streaming at a constant bitrate. It should be also noted that the MOS for the case of video streaming at a constant bitrate is higher compared to the MOS for DASH video streaming irrespectively of the rebuffering frequency. Further, when considering the DASH video streaming results, it is observed that although the average video bitrate in the two cases of DASH video streaming is approximately similar, the MOS for the two bandwidth models shown in Fig. 5.5 is significantly different. It should be further emphasized that the case with a higher average bitrate results in a lower MOS, therefore, the average bitrate alone is not enough to capture the MOS. The aforementioned remarks are valid for both the MOS scores from the subjective tests and the MOS values from the post processing. Similar procedure can be followed by using bandwidth traces from 4G or 5G measurements in order to evaluate the streaming quality for these emerging networks.

The post processing shows that the exponential model with two parameters estimates the MOS score from the subjective tests with high R-square in the considered rebuffering frequency range ϕ^k, for all k values. The MOS models for the constant bitrate video streaming and for DASH video streaming are expressed as:

$$MOS = \begin{cases} Q_c^m(\phi^k, \rho_c^i) = a_c(\rho_c^i) exp(\phi^k b_c(\rho_c^i)) \\ Q_{dash}^m(\phi^k, R_{\text{av}}, \hat{\sigma}) = a_{\text{dash}}(R_{\text{av}}, \hat{\sigma}) exp(\phi^k b_{\text{dash}}(R_{\text{av}}, \hat{\sigma})) \end{cases} \tag{5.3}$$

Table 5.4 Model parameters.

Parameter	a	b	R-square
Constant streaming with 800 kbps	3.975	-0.053	0.97
DASH video streaming over 3G	3.616	-0.063	0.88
DASH video streaming over AB	3.15	-0.058	0.93

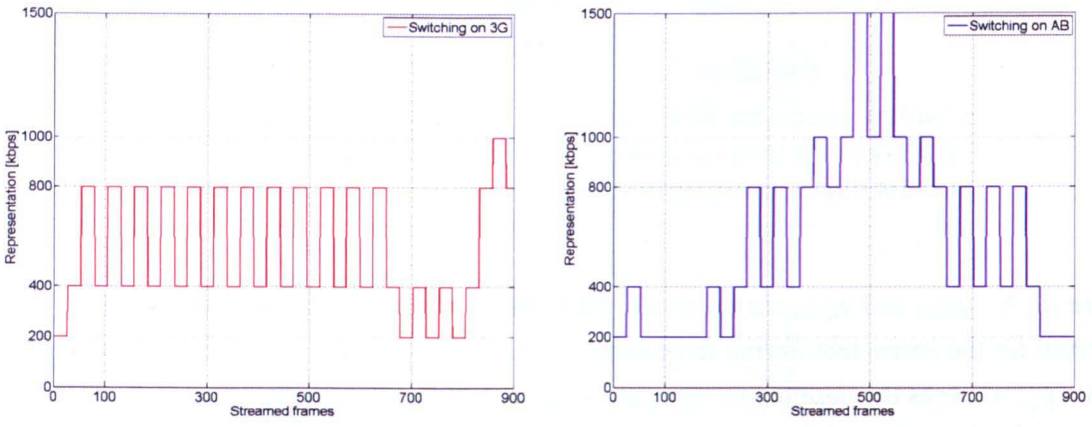
where a_c, b_c, a_{dash} and b_{dash} are the model parameters obtained with exponential fitting and are fixed for the entire rebuffering frequency range. The parameters a_c and b_c depend on the bitrate ρ_c^i, whereas the parameters a_{dash} and b_{dash} depend on the average bitrate R_{av} and the variance $\hat{\sigma}$ of the changes in the representation levels during DASH.

The model parameters for the considered cases are reported in Table 5.4. It can be noted that there is a small difference in the b_c parameter for video streaming at a constant bitrate ($b_c = -0.0053$) and the b_{dash} parameters for DASH video streaming over 3G ($b_{\text{dash}} = -0.0063$) and over AB ($b_{\text{dash}} = -0.0058$). However, there is a larger difference in the a parameters. One of the factors that influences the difference in the a parameters (hence, the difference in the MOS) is the quality variations (additional factors that influence this difference are discussed in subsection 5.1.4.3). The difference between the models is tightly related to the a parameters. This indication is used as a basis for the proposed method.

5.1.4.3 Proposed method

Fig. 5.6 depicts the two switching pattern sets for the DASH video streaming of the SOCCER sequence with the 3G and AB bandwidth models. As mentioned, the average bitrates in the two cases are $R_{\text{av}}^{\text{AB}} = 598.6$ kbps and $R_{\text{av}}^{\text{3G}} = 550.4$ kbps, whereas the variances are equal to $\hat{\sigma}^{\text{AB}} = 0.28$ and $\hat{\sigma}^{\text{3G}} = 0.18$ (values obtained with simple scaling of the set of representations used during DASH). According to the results in Table 5.4, the a_{dash} parameter in the AB case ($a_{\text{dash}} = 3.15$) is smaller compared to the a_{dash} parameter in the 3G case ($a_{\text{dash}} = 3.61$). Hence, in the considered setup, higher variance results in a lower MOS. Further, the a_c parameter in the case of video streaming at a constant 800 kbps bitrate ($a_c = 3.97$), as expected, is larger compared to the a parameters for the DASH video streaming since there is no quality variations, i.e., $\hat{\sigma}^c = 0$ during video streaming at a constant bitrate.

An initial point in the proposed method is to observe the changes in the ratio between the a_c parameter for different birates (not only the considered 800 kbps) and the a_{dash} parameter. The a parameters from the two models are only considered, since the b parameters are approximately equal. The ratio is denoted as K, and it is observed for different bitrates $\{\rho^i, i = 1, 2, .., M\}$, since the a_c parameter depends on the bitrate. During the calculation of



(a) 3G bandwidth model. (b) AB bandwidth model.

Fig. 5.6 Switching between DASH representations.

the ratio K, the a_{dash} parameter remains fixed and depends on the average bitrate R_{av} and the variance $\hat{\sigma}$ of the DASH video streaming. Therefore, the ratio is defined as:

$$K(\rho^i) = \frac{a_c(\rho^i)}{a_{\text{dash}}(R_{\text{av}}, \hat{\sigma})}. \tag{5.4}$$

The idea is to find a particular bitrate, denoted with I_{cb}, in the model for video streaming at a constant bitrate that would provide quality as good as the quality of DASH video streaming characterized with the average bitrate R_{av} and the variance $\hat{\sigma}$. It is reasonable to assume that this bitrate is bounded between two bitrates from the $\{\rho^i, i = 1,2,..,M\}$ set. This bitrate corresponds to a threshold ($K = 1$) for which the parameters in the two models are equal, i.e., $a_c(I_{cb}) = a_{\text{dash}}(R_{\text{av}}, \hat{\sigma})$ and hence, the MOS values from the models are equal. Therefore, assumming that the $a_c(I_{\text{cb}})$ parameter is known for all I_{cb} bitrates, the a_{dash} is obtained and hence, the MOS for the DASH video streaming can be expressed.

Applying this approach to the considered setup, the results for the ratio K are shown in Fig. 5.7. The K ratio is plotted for the $\{\rho^i, i = 1,2,..,M\}$ set (reported in the x-axis) used in subsection 5.1.2.

The intersection (point marked with "x") between the 3G curve and the threshold corresponds to a slightly higher constant bitrate on the x-axis compared to the average bitrate $R_{\text{av}}^{\text{3G}}$ (the point on the 3G curve that corresponds to the $R_{\text{av}}^{\text{3G}}$ is also marked with "x" for comparison), therefore $I_{\text{cb}} > R_{\text{av}}^{\text{3G}}$.

The intersection (point marked with "+") between the AB curve and the threshold corresponds to a significantly lower constant bitrate on the x-axis compared to the average

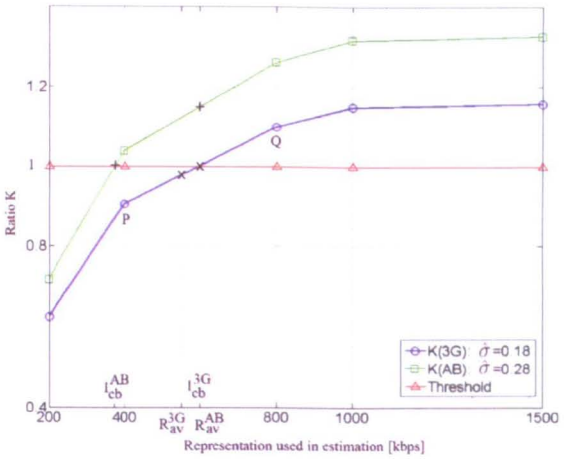


Fig. 5.7 K ratio for DASH video streaming and video streaming at different constant bitrates.

bitrate $R_{\text{av}}^{\text{AB}}$ (the point on the curve for the AB curve that corresponds to the R_{av}^{AB} is also marked with "+" for comparison), therefore $I_{\text{cb}} < R_{\text{av}}^{\text{AB}}$.

The I_{cb} bitrates for the 3G and AB cases are denoted with $I_{\text{cb}}^{\text{3G}}$ and $I_{\text{cb}}^{\text{AB}}$ and are also reported in the x-axis.

The proposed method is summarized with the following observations.

1. The average bitrate and the variance of the DASH video streaming determine a piecewise linear curve that shows how the ratio K changes for different bitrates as shown in Fig. 5.7.

2. Two appropriate points from the curve can be selected and a linear equation between the points by considering the coordinates (as approximation) can be used. For example, in the case of DASH video streaming over 3G, by selecting the points $P(400, \frac{a_c^p}{a_{\text{dash}}})$ and $Q(800, \frac{a_c^q}{a_{\text{dash}}})$, the following is observed:

$$y - \frac{a_c^p}{a_{\text{dash}}} = \frac{\frac{a_c^q}{a_{\text{dash}}} - \frac{a_c^p}{a_{\text{dash}}}}{800 - 400}(x - 400). \tag{5.5}$$

3. The solution from the intersection with the threshold line and the line expressed in (5.5) provides the required bitrate I_{cb} for which the MOS value of the video streaming at a constant bitrate is equal to the MOS value of the DASH video streaming.

4. Assuming that the model parameters for the video streaming at constant bitrate I_{cb} are known, the model parameters for the DASH video streaming are obtained, since $a_c = a_{\text{dash}}$ and $b_c = b_{\text{dash}}$.

5. The MOS value from the model for DASH video streaming $Q^m_{\text{dash}}(\phi^k_{\text{dash}}, R_{\text{av}}, \hat{\sigma})$ is equal to the MOS value from the model for video streaming at a constant bitrate $Q^m_c(\phi^k_c, \rho^i_c)$ for $\rho^i_c = I_{\text{cb}}$.

The appropriate points (P and Q) in the method are selected so that the obtained solution a_{dash} is bounded by $a^p_c < a_{\text{dash}} < a^q_c$ which is valid only for two neighboring points from the piecewise linear curve.

In the end, it should be emphasized that there are several parameters which are not considered in the modeling (amplitude of the switching between different representations, initial buffering, variable rebuffering durations, etc.). In the presented study, fixed duration of the rebuffering events and no initial buffering is adopted; variable durations of the rebuffering events and different initial buffering influence the MOS, but the impact of these parameters is significantly lower compared to the rebuffering frequency [81]. In addition, the study in [143] presents further insight on the influence of the rebuffering interruptions on the user's QoE. In this section, uniform distribution of the rebuffering events is used. However, the position in the sequence where rebuffering occurs has an impact on quality, since the freezing of a frame from a high motion scene might be perceived differently compared to a freezing of a frame from a low motion scene.

5.2 Content-aware quality metric for DASH

This section presents the novel content-aware quality (CAQ) metric which considers content specific, video encoding, and rebuffering related parameters. The proposed metric is evaluated with real-time continuous quality measurements of sequences with different impairments that typically occur in DASH systems (quality switching, rebuffering, etc.). Subsection 5.2.1 addresses related works, whereas the metric is presented in subsection 5.2.2.

5.2.1 Related work

The research in the literature related to QoE for DASH can be categorized in two general groups: works that focus on a quality metric/model and works that propose an adaptation strategy/logic that implicitly uses a quality related function. Henceforth, the term quality metric is used to refer to any function that models the users' QoE by using a set of different

parameters. The quality can be assessed either at the end of the streaming, by assigning a single score, or in real-time during the streaming, by assigning several scores over time at predefined time intervals. Several works on this topic were previously discussed in Chapter 2, Section 2.2.3.3, however there is no metric in the literature that combines content specific, video encoding, and rebuffering related parameters. Several previously discussed related works and additional works are compared and contrasted with the CAQ metric as follows.

Early notable works in the literature that propose a quality metric for DASH are presented in [81] and [82], where both metrics provide the quality (as a MOS value) at the end of the streaming. The metric in [81] expresses the MOS as a linear combination of the initial buffering time, the mean duration of rebuffering events, and the rebuffering frequency. In this metric, the most significant input is provided by the term related to the rebuffering frequency, which is also confirmed in a previous study on understanding the video quality impact on user engagement [61]. A no-reference metric based on Random Neural Networks (RNN) is proposed in [82] which considers the following two parameters: playout interruptions and video quality variations. The playout interruptions are represented as a function of three measurable parameters: the total number of playout interruptions, the average duration of interruptions and its maximum value, whereas the quality variations are represented through the quantization parameter. These works focus on capturing the QoE for DASH through parameters related to the rebuffering, and in the case of [82] the quantization parameter, however additional video encoding parameters (frame rate, PSNR of the frames or bitrate) and content related parameters are not considered.

Few works on TCP video streaming services propose quality metrics and methodologies for quality evaluation at the end of the streaming session [86], [87]. The authors in [86] propose a quality monitoring mechanism in a dedicated measurement point in the network that considers the number and duration of the stalling (rebuffering) events, and the time elapsed since the last stall. This network-based solution merges these three parameters into a single parameter called Negative Impact (NI) that is linearly dependent on the MOS. The contribution of the three parameters towards the NI is expressed through exponential functions and several dimensioning parameters are used. In the end, a general discussion on applying the proposed mechanisms and the quality metric in the DASH context is presented; however evaluations in a general DASH setup and DASH related parameters that impact the quality (PSNR of the streamed frames for different representations during quality switching, frame rate, etc.) are not investigated. The authors in [87] propose a Video Streaming Quality Metric (VsQM) metric to assess the TCP streaming service by considering the number, duration and the temporal location of the pauses. The pauses occur at locations characterized with different spatial and temporal information (three sequences are used) and several application scenarios

are considered. The same authors extended this work in [88] by proposing a no-reference video quality metric for DASH based on two parameters: pauses and changes in the video resolution (temporal resolution). In addition, several weighting factors are introduced and an exponential function is used to map the metric values in the 5-point MOS scale. In contrast to these works, the metric proposed in the dissertation considers a broader range of DASH and content related parameters to capture the quality.

The works in [78, 89] investigate different adaptation approaches based on the metric derived in [78]. This metric considers the average segment quality penalized by the frequency and length of the video freezes (rebuffering) and the quality switching, summarized in three factors that further depend on the number of segments, the segment bitrate, the average bitrate between the switches and the minimum and maximum representations in the setup. The three factors are weighted and linearly combined in a single expression to obtain the quality at the end of the streaming session. Calculating these factors requires a predefined information on setup dependent parameters and this metric is not considering content related parameters. The significance of the content-related parameters is discusses in the subjective quality study presented in [83]. The authors investigate the impact of the adaptation approaches on the visual quality for monoscopic and stereoscopic video and highlight that the QoE resulting from the adaptation on differen* video content and format is diverse. Although no specific metric is proposed in this work, the observations related to the QoE dependency on the content show that content related parameters are relevant and require consideration during the design of a QoE metric for DASH.

Two approaches to model the user experience at the end of the DASH session are presented in [84, 91]. In [91], three main factors: initial delay, stall and bitrate, and multiple dimensions for each factor are considered, including the total stall duration, number of stalls, timing, average bitrate, number of switches and average switch magnitude. Therefore, this work is the first to study both the spatial and temporal quality of the video streaming through deriving separate impairment functions. The proposed metric is derived based on real mobile network condition for short videos and the impairment functions are derived for each of the influencing factors; however, a single metric that considers all factors is not proposed. The work in [84] presents an automated QoE evaluation for DASH that considers media related parameters such as the bitrate and the frame rate and HTTP/TCP-related parameters such as the rebuffering events and the frequency and amplitude of the quality switches. The QoE is expressed through a non-linear metric that weights all the considered parameters. Although these metrics cover a broad range of parameters, content related parameters are not considered.

A comprehensive work on a metric that considers the time varying subjective quality is presented in [85]. The Hammerstein-Wiener model is used for predicting the subjective quality, and the metric enables real time MOS evaluation by considering several factors related to the impact of quality fluctuations on the MOS (sensitivity to switches, recency, etc.). The conclusions indicate that the users' response is more sensitive in the low video quality regions, and the recency effect of quality fluctuations spans across intervals of 15s. However, the rebuffering events are not considered in this metric and several aforementioned studies showed that the rebuffering is a major factor that affects the QoE.

5.2.2 Proposed QoE metric

The proposed CAQ metric is tailored to consider the importance of the content, video encoding and rebuffering parameters for the quality of DASH streaming. Therefore, the content and video decoding impact on quality as expressed in [144] is used, and the rebuffering impact on quality as expressed in [81] is used. The metric in [144] expresses the quality (Q_1), by considering the frame rate (f), the quantization artefacts ($PSNR$), and two content dependent parameters (b,s). The authors show that the impact of the frame rate and the quantization on the overall quality is separable, and by additionally considering the content parameters, the metric is expressed as:

$$Q_1 = 1.04\left(1 - \frac{1}{1+e^{0.34(PSNR-s)}}\right)\left(\frac{1-e^{-b\frac{f}{f_{max}}}}{1-e^{-b}}\right), \tag{5.6}$$

where f_{max} is the highest frame rate considered, and $PSNR$ is the average PSNR of the video frames. The content related parameters (b,s) are associated with several aspects. The b parameter represents the speed at which the perceived quality drops as the frame rate is reduced. This parameter is linked to the difficulty of the human eye to track moving objects in a content when the frame rate is reduced, hence it depends on the temporal variations in the content. The s parameter is linked to the contrast and the amount of details in the content where smaller s values are assigned for content with many details. Effective prediction models for the content related parameters are provided in [144]:

$$b = 10.72 - 0.6\alpha - 0.13\beta, \tag{5.7}$$

$$s = 34.82 - 1.88\gamma - 2.23\delta, \tag{5.8}$$

where α is the motion direction activity, β is the displaced frame difference, γ is a feature obtained as an output from Gabor filter related to the strength and variations of horizontal and vertical edges in a sequence and δ is the normalized frame difference.

The metric in [81] expresses the quality (Q_2), at the end of the adaptive HTTP video streaming by considering: the initial buffering (L_{ti}), the rebuffering frequency (L_{fr}) and the average rebuffering duration (L_{tr}) corresponding to low, medium, and high level of rebuffering impairments where a higher level results in lower quality. The terms assume values from 1 to 3 for low, medium and high level of rebuffering impairments based on the obtained regression model. The metric is expressed as:

$$Q_2 = 4.32 - 0.067L_{ti} - 0.742L_{fr} - 0.106L_{tr}. \tag{5.9}$$

The first metric (5.6) is content specific and it considers video quality parameters, although it is not suitable for HTTP streaming applications, whereas the second metric (5.9) is tailored for adaptive HTTP video streaming, but it is not content specific and it is not dependent on parameters related to the image quality. The proposed metrics captures the aforementioned aspects into one CAQ metric for DASH systems.

5.2.2.1 CAQ metric formulation

Let us consider M possible content classes, indexed with $m = 1,2,..,M$. A video sequence from the content class m is characterized with the content related parameters $(b_m; s_m)$. Further, a video sequence from the content class m can be represented by DASH segments indexed with n_m, where it is assumed $n_m = 1,2,..,N_m$, hence the entire sequence from the content class m is represented by N_m DASH segments. Each of these segments can have a different average PSNR of the frames $PSNR_{n_m}$ and different frame rate f_{n_m}. Initially, the case of streaming a video from one content class represented by one segment which is played out without rebuffering is assumed. In such case an adapted version of the metric (5.6) can be used, where for DASH, the quality denoted as $Q_3(m, n_m)$ depends on the content related parameters for class m and on the video quality parameters related to the considered segment indexed with n_m according to:

$$Q_3(m,n_m) = 1.04\left(1 - \frac{1}{1+e^{0.34(PSNR_{n_m}-s_m)}}\right) \left(\frac{1-e^{-b_m\frac{f_{n_m}}{f_{max}}}}{1-e^{-b_m}}\right). \tag{5.10}$$

On the other side, assuming rebuffering occurs, referring to the metric in (5.9) and inserting the following modifications: the coefficients $\{\alpha_{fr}^{(m)}; \alpha_{tr}^{(m)}\}$ are used to capture the intuitive assumption that the impact of the rebuffering on the MOS depends on the content characteristics (the initial buffering term is omitted due to its low impact). Further, replacing the constant term with a variable term, that depends on the content related parameters for class m and the video quality parameters related to the considered segment indexed with n_m. In streaming, several content classes can occur and the number of segments from each segment class can be different, hence the following term is introduced:

$$K = \frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{N_m}\sum_{n=1}^{N_m} Q_3(m,n_m)\right). \tag{5.11}$$

and the modified metric based on (5.9) and (5.11) is expressed as:

$$Q_4 = K - \alpha_{fr}^{(m)} L_{fr} - \alpha_{tr}^{(m)} L_{tr}. \tag{5.12}$$

where instead of using a fixed term equal to 4.32 as in (5.9), the K value as defined in (5.11) is introduced in order to include the impact of different content classes and the number of segments in each class. Note, that appropriate mapping between the scale used in (5.6) and the MOS scale between 0 and 5 is performed for the K values.

The highest possible QoE value Q_4 in (5.12) is 5, and it is obtained when the streaming is with sufficiently high video quality parameters as calculated according to (5.11) and all the remaining terms are zero, i.e., when there is no rebuffering. Further, the QoE value Q_4 in (5.12) is decreased when the streaming of the content classes is with low video quality parameters as calculated according to (5.11) or the remaining terms are not zero, i.e., rebuffering occurs in some parts of the streaming. The expression given in (5.12) captures the QoE for DASH session composed of total of N_m segments from different classes $m = 1,2,..,M$. This expression is further modified by considering the characteristics of the different content classes and the rebuffering as follows. The proposed coefficients $\{\alpha_{fr}^{(m)}; \alpha_{tr}^{(m)}\}$ are expressed by using the values 0.742 and 0.106 from (5.9) and a parameter related to the content class, i.e., content class index denoted as c_m, hence:

$$\alpha_{fr}^{(m)} = \frac{1}{5} c_m * 0.742 \tag{5.13}$$

$$\alpha_{tr}^{(m)} = \frac{1}{2} c_m * 0.106. \tag{5.14}$$

The content class index c_m is a tunable parameter in the metric, however the general idea is to increase the value of this index when rebuffering occurs in parts of the streaming with

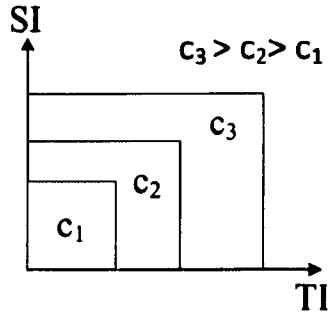


Fig. 5.8 Assignment of content related parameters.

higher spatial and temporal complexity as depicted in Fig. 5.8. The scaling with the terms $\frac{1}{5}$ and $\frac{1}{2}$ is used to normalize the values in the MOS range (0,5) according to the assigned values for the content class index. Further, when substituting (5.11), (5.13), and (5.14) in (5.12), the QoE value denoted as Q_{CAQ} from the proposed content-aware quality metric is obtained as:

$$Q_{CAQ} = \left(\frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{N_m}\sum_{n=1}^{N_m} Q_1(m,n_m)\right) - \left(\frac{1}{5}c_m * 0.742L_{fr} + \frac{1}{2}c_m * 0.106L_{tr}\right)\right) + \delta(x) \tag{5.15}$$

where additionally the function $\delta(x)$ is included to provide normalization of the values in the MOS range (0,5) taking into account the rebuffering levels. It is a linear function due to the linear dependency between the MOS and the coefficients assigned for the different rebuffering levels. This function is defined as:

$$\delta(x) = \begin{cases} 0 & L_{fr} = L_{tr} = 1 \\ max(L_{fr}, L_{tr}) & \text{all others} \end{cases}$$

Table 5.5 presents relevant information regarding the considered QoE metrics, including the used notation for the QoE value from the metric, the key parameters considered in the metric and the suitable video application for the metric.

Table 5.5 Considered QoE metrics.

Metric	QoE notation	Key parameters	Application
Metric in [144]	Q_1	PSNR Frame rate Content parameters	Video
Metric in [81]	Q_2	Initial buffering Rebuffering frequency Rebuffering duration	HTTP streaming
CAQ metric	Q_{CAQ}	PSNR Frame rate Content parameters Rebuffering frequency Rebuffering duration Content class index	HTTP streaming

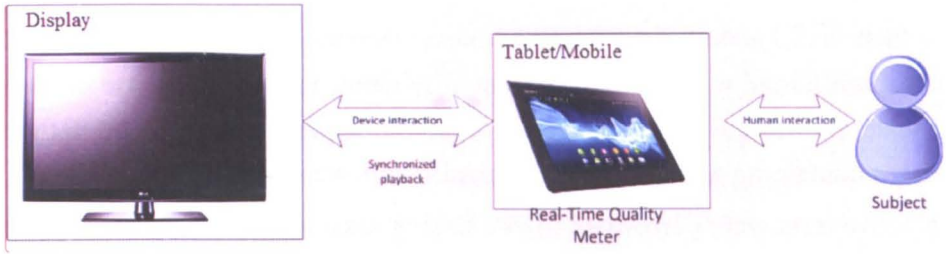


Fig. 5.9 Subjective quality evaluation setup.

5.2.3 Subjective tests & Evaluations

Subjective quality evaluation is necessary in order to investigate how the QoE values from the proposed metric correlate with the collected scores from the subjective tests. The setup for the study is depicted in Fig. 5.9. A new mobile/tablet application (Android APP) has been developed within the Kingston WMN lab in order to collect subject's quality scores in real-time. This real-time quality meter synchronously works with the display device displaying test video sequences and recording user input in a 5-point category scale. A slider is used to record user opinion over time. The initial position of the slider is at the maximum level (i.e., excellent or 5). This device interaction is through an IEEE 802.11n WiFi router. A snapshot of the user interface of the real-time quality meter is shown in Fig. 5.10.

Subjective scores are collected at every second of the viewing session. At the end of the test, the data can be retrieved from the APP or sent to a remote server. Two experiment sessions were conducted and 10 sequences were evaluated by 15 subjects (males and females in the $20-45$ age range). All naïve subjects recruited for quality evaluation tests were

Fig. 5.10 Graphical user interface of the real-time quality meter.

screened for their visual acuity and colour blindness. Two training sessions were conducted for subjects to familiarize with the real-time quality meter interface (the use of the slider, categorical scale, positioning of the tablet, etc.) and the sequences being used in the actual tests (sequences containing a broad range of impairments were used).

The subjective tests were performed in two testing sessions. In the first testing session, different content types, random switching between the quality levels is used and the segment size is set at 5s, thus there is a slow quality variations. In the second testing session, the content type is fixed and fast quality variations by setting the segment size at 1s are considered. The switching between the quality levels is based on the adaptation logic elaborated in subsection 5.2.3.2.

5.2.3.1 Testing session #1

The sequences for the first testing session were produced by concatenating the CITY, CREW and FOREMAN sequences ($M = 3$ and each sequence has duration 10s) and repeating the concatenation (duration 30s) for three cycles, thus the resulting sequences last for 90 seconds. The CITY, CREW and FOREMAN sequences are represented with two 5 seconds long segments ($N_1 = N_2 = N_3 = 6$), hence the total number of segments during the streaming is 18. These three sequences are selected due to their different spatial information (SI) and temporal information (TI), which resulted in different (b, s) parameters associated to the sequences as reported in Table 5.6. The CITY sequence has the lowest SI and TI information, followed by CREW and then by the FOREMAN sequence that has the highest SI and TI.

Table 5.6 SI and TI for the sequences

Sequence	SI	TI	b	s
CITY	11.13	20.22	7.41	26.3
CREW	21.08	26.81	7.23	29.68
FOREMAN	24.11	36.25	8.24	29.09

Table 5.7 Average PSNR of the segments.

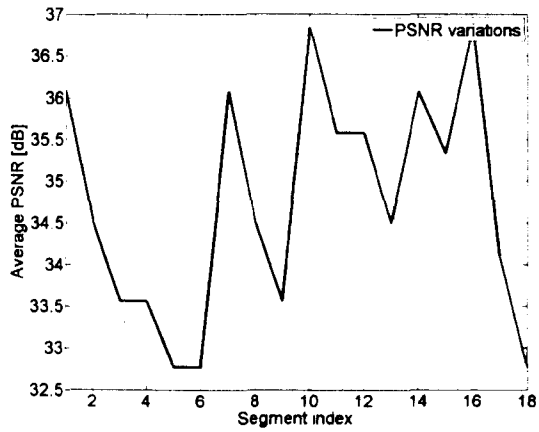
Sequence	Representation [kbps]	Average PSNR
CITY	150	34.50
	200	36.08
	250	37.31
	300	38.45
CREW	150	33.56
	200	35.34
	250	36.84
	300	38.23
FOREMAN	150	32.77
	200	34.11
	250	35.06
	300	35.58

Further, four representation levels $\{150, 200, 250, 300\}$ expressed in *kbps* at resolution 352×288 and frame rate 30 fps were used. The average PSNR of the segment's frames for the representations is reported in Table 5.7, where lower SI and TI results in a higher average PSNR of the frames for a fixed representation.

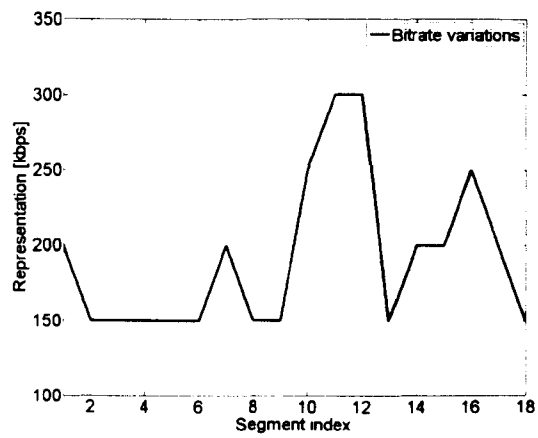
In the end, low and high level rebuffering events as categorized in [81] were inserted in the parts that contain different spatial and temporal information, i.e., parts that can be characterized with different content related parameters. Consequently, 6 sequences with the same switching pattern were obtained, where the pattern is randomly generated under equal probability of occurrence for the representations. The PSNR and the bitrate variations as a result of the switching pattern are shown in 5.11.

The resulting sequences are labeled with $\{S_{L,1}, S_{H,1}, S_{L,2}, S_{H,2}, S_{L,3}, S_{H,3}\}$ where the first index refers to the level of rebuffering, whereas the second index refers to the content affected by rebuffering. The information regarding the rebuffering impairments is given in Table 5.8.

The collected MOS scores from the subjective tests are averaged and depicted in Fig. 5.12. The MOS variations during the streaming show how the quality changes in different parts of the content and as different representations are used. In particular, it is noticeable



(a) PSNR variations.



(b) Bitrate variations.

Fig. 5.11 PSNR and bitrate variations for testing session #1.

Table 5.8 Streaming impairments

Sequence label	Rebuffering level	Rebuffering frequency [%]	Rebuffering duration [s]	Rebuffering position
$S_{L,1}$	Low	$0-2$	$0-5$	Segment:1,2,7,8,13,14
$S_{H,1}$	High	>15	>10	Segment:1,2,7,8,13,14
$S_{L,2}$	Low	$0-2$	$0-5$	Segment:3,4,9,10,15,16
$S_{H,2}$	High	>15	>10	Segment:3,4,9,10,15,16
$S_{L,3}$	Low	$0-2$	$0-5$	Segment:5,6,11,12,17,18
$S_{H,3}$	High	>15	>10	Segment:5,6,11,12,17,18

Table 5.9 Average PSNR of the segments.

Sequence	Representation [kbps]	Average PSNR
CREW	200	27.52
	400	31.12
	800	32.66
	1000	33.26
	1500	36.47

that the MOS is sensitive to the different content for the case of low rebuffering events. Lowest MOS for large part of the streaming is obtained when the rebuffering events occur in the part of the content with the highest SI and TI. This effect diminishes for the high rebuffering events since the changes in the obtained MOS in this case become insignificant. Therefore, in the second testing session, the impact on MOS for low level of rebuffering events is investigated.

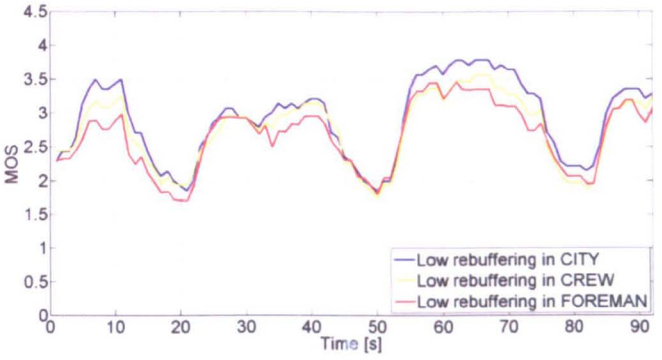
5.2.3.2 Testing session #2

In the second testing session only the CREW sequence (looped three times) is used in order to investigate how the metric performs during frequent quality changes. Therefore, the segment size is set at $1s$ and the artificial bandwidth (AB) model [142] used in Section 5.1. The average PSNR at the level of the segment is reported in Table 6.1.

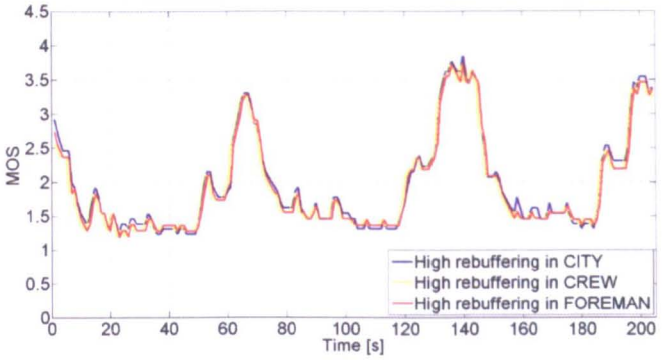
Five representation levels $\{200, 400, 800, 1000, 1500\}$ expressed in *kbps* were used and the video resolution was set at 636×420 and frame rate 25 fps. Four impaired sequences were evaluated corresponding to the following rebuffering frequencies expressed in percentages $\{0, 2\%, 4\%, 6\%\}$, where an adaptation logic as described in Subsection 5.1.2.2 is used. Fig. 5.13 depicts the PSNR and the bitrate variations, whereas the subjective scores are reported in Fig. 5.14.

5.2.3.3 Evaluations

The number of MOS scores samples collected for the sequences in the first testing session is larger than the ones collected for the second testing session; further, higher level of rebuffering results in a larger number of MOS scores samples since the samples are collected for every second during the subjective tests. The previously depicted MOS scores were averaged across the number of subjects. However, in order to characterize the entire DASH streaming, additional averaging of these scores is performed in time, hence a single MOS

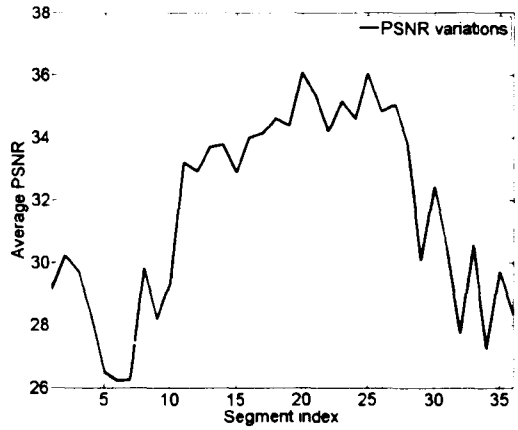


(a) Low rebuffering in different parts of the content.

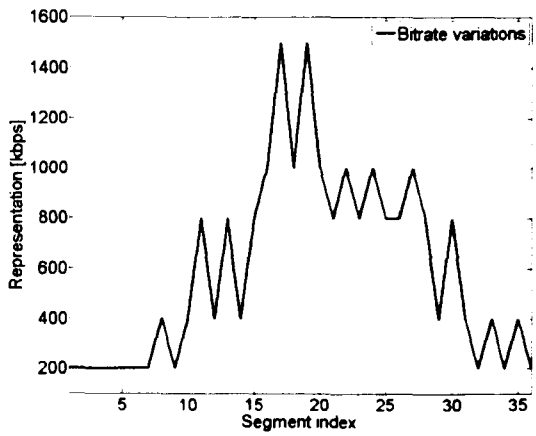


(b) High rebuffering in different parts of the content.

Fig. 5.12 MOS scores for the first testing session.

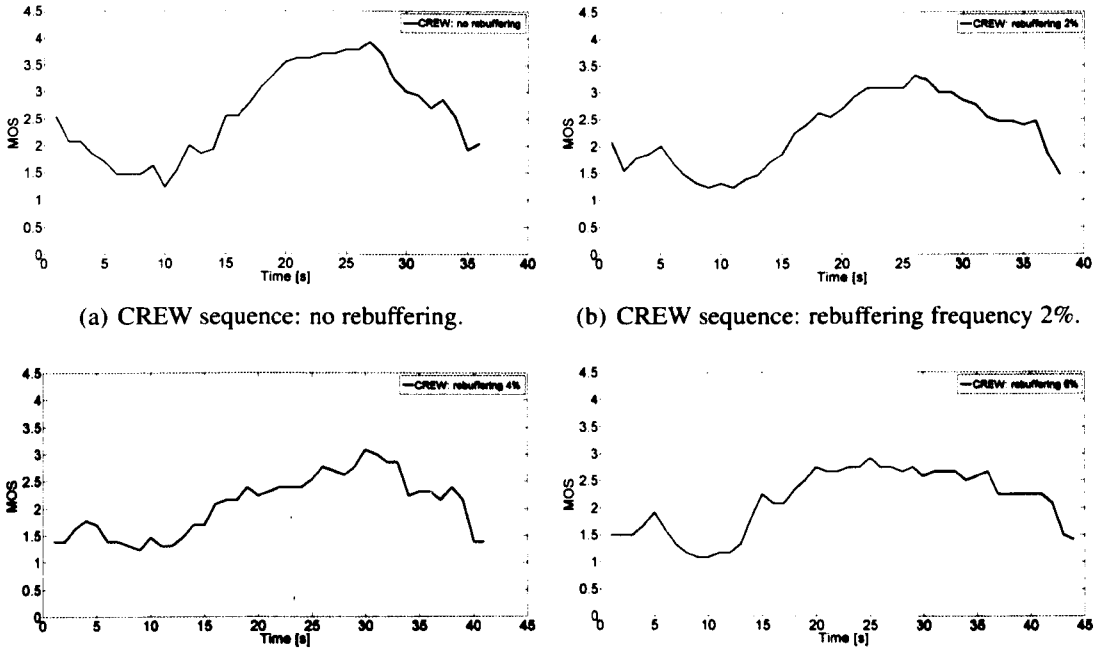


(a) PSNR variations.



(b) Bitrate variations.

Fig. 5.13 PSNR and bitrate variations for testing session #2.



(a) CREW sequence: no rebuffering.

(b) CREW sequence: rebuffering frequency 2%.

(c) CREW sequence: rebuffering frequency 4%.

(d) CREW sequence: rebuffering frequency 6%.

Fig. 5.14 MOS scores for the second testing session.

score is obtained for each sequence. These scores are compared to the MOS values obtained with:

- The metric presented in (5.9) where the MOS value is denoted as Q_2.
- The proposed content-aware quality metric where the MOS value is denoted as Q_{CAQ}.

The Q_2 value is based on the state of the art metric for DASH, however this value is obtained by not considering the video quality parameters for different content classes and the rebuffering impact on different content classes. These aspects are considered in the MOS value Q_{CAQ} from the proposed metric for which the content related parameters are obtained via tuning and are assigned to $c_1 = 8.5$, $c_2 = 9.5$ and $c_3 = 10.5$ for CITY, CREW and FOREMAN, respectively. Note that a higher value for the content related parameter is assigned for the sequence with higher spatial and temporal complexity.

Fig. 5.15 depicts the MOS scores that characterize the entire DASH streaming, the MOS values from the state of the art metric and the MOS values from the proposed content-aware quality metric in ascending order for the considered sequences. The proposed metric provides better estimation of the MOS score for the majority of the sequences (except sequence 1). Table 5.10 reports the exact values depicted in Fig. 5.15 and shows the Spearman, Pearson

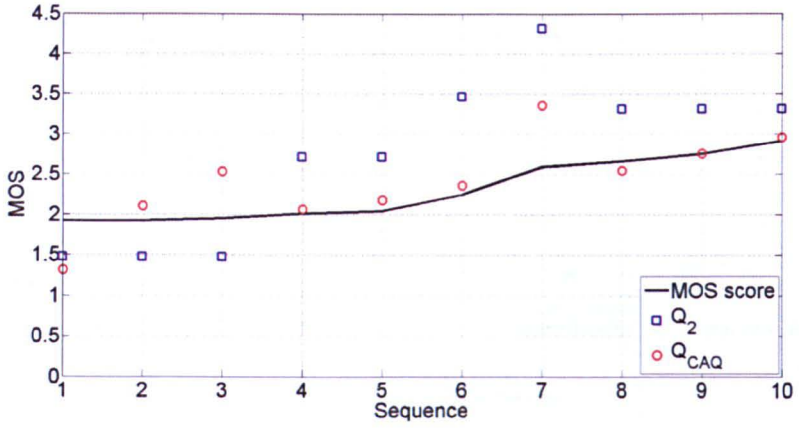


Fig. 5.15 State of the art metric vs. CAQ metric.

and Kendall correlation coefficients for the two metrics as calculated in Matlab using the *corr* function with the appropriate *type* parameter. The Spearman and Kendall coefficients for the proposed metric are 4 percentage points higher compared to the state of the art metric, whereas the Pearson is only 1 percentage point higher for the state of the art metric. The benefit provided by the proposed metric is due to the consideration of video quality parameters for different content classes and the rebuffering impact on different content classes.

5.3 Chapter summary and contributions

This chapter addresses aspects related to the QoS and QoE for DASH. In particular, a study on the QoS and QoE for video streaming for DASH systems is presented and a novel content-aware metric for DASH is proposed and evaluate. As a result, this chapter summarizes the following works and contributions:

- An analysis on how the GOP size in the encoding and the segment size in the segmentation of the DASH dataset impact the following metrics: PSNR, video throughput, rebuffering frequency and MOS of DASH video streaming. The usage of smaller segment sizes enables better adaptation to the bandwidth changes, hence it improves the considered metrics. Reducing of the GOP size with respect to the segment size ($G < T$) is not effective for small segment sizes for all the considered metrics. However, using $G < T$ can slightly improve the rebuffering frequency for large segment sizes, although in this case the MOS is not significantly affected. Also in terms of MOS, the subjective tests showed that using small segment sizes and $G = T$ provides better results.

Table 5.10 MOS scores and MOS values from the metrics

Sequences	Mean MOS	Q_2	Q_{CAQ}
Seq 1	2.91	3.31	2.96
Seq 2	2.75	3.31	2.76
Seq 3	2.66	3.31	2.55
Seq 4	1.95	1.48	2.54
Seq 5	1.92	1.48	2.11
Seq 6	1.92	1.48	1.32
Seq 7	2.59	4.32	3.36
Seq 8	2.24	3.47	2.36
Seq 9	2.04	2.72	2.18
Seq 10	2.01	2.72	2.07
Spearman coefficient	-	0.78	0.82
Pearson coefficient	-	0.76	0.75
Kendall coefficient	-	0.63	0.67

- A model and a novel method for modeling the MOS of the DASH video streaming based on a MOS model for video streaming at a constant video bitrate. It is observed that the exponential model fits the MOS scores from the subjective scores for the video streaming at a constant bitrate and for the DASH video streaming with a high R-square. Then, the MOS models parameters are observed and a novel method to estimate the MOS for DASH video streaming is proposed under the assumption that the MOS for the video streaming at a constant bitrate is known in advance.

- Novel content-aware quality metric for DASH. This metric considers content specific, video encoding, and rebuffering related parameters and it is is evaluated with real-time continuous quality measurements of sequences with different impairments that typically occur in DASH systems (quality switching, rebuffering, etc.). The results show that the metric provides better estimation of the average MOS compared to the state of the art metric, which considering the Spearman, Pearson and Kendall correlation coefficients.

Chapter 6

Navigation experience for DASH

DASH provides an enhanced QoE for linear video streaming (sequential display order of images) by encoding segments in multiple bit rates and fetching segments at the appropriate bit rate, according to the supported data rate in the network. However, users can interact and control the viewing process by using a particular navigation function (e.g., fast forward/backwards, pause.) and hence distort the linear video streaming. Navigation functions enable non-linear video streaming during which the display order of the images is not known in advance and it is controlled by the user. In this chapter, the term navigation experience is used to capture the users' perception of quality for non-linear video streaming, similar to the conventional usage of QoE in the case of linear video streaming. The users' perception of quality is captured by the proposed analytical model that shows the dependency of the expected navigation delay during non-linear video streaming on several parameters related to the DASH video streaming system.

This chapter is organized as follows. A general use case for non-linear DASH is presented in Section 6.1. The proposed analytical model for capturing the navigation experience through the expected navigation delay is derived and evaluated in Section 6.2. Section 6.3 concludes the chapter.

6.1 Use case for non-linear DASH

Besides the conventional DASH-based linear video streaming applications, there are several applications, where media like video or image sequences are watched in a non-linear fashion. One particular example is from the healthcare domain, i.e., viewing Computed Tomography (CT), Magnetic Resonance (MR) or Digital Imaging and Communications in Medicine (DICOM) volume data. In these cases, the date volume consists of a sequence of pictures (so-called slices). Unlike in linear video streaming, the display order of the pictures is not

known in advance, since the user (e.g., the radiologist) controls the viewing process. Further, the users can navigate to a particular image data in the video stream. In such cases, if the relevant image data is not available yet at the client, the user experiences a delay and has to wait until the relevant image data has been transmitted from the server, which stores the full media data set.

A general use case for non-linear DASH is depicted in Fig.6.1. The DASH segments are shown as rectangles spanning across the time and quality direction. The time direction corresponds to the DASH timeline as defined in the standard, whereas the quality direction corresponds to the different DASH representations. In the time direction (indexed with $m = 1,2,..,M$) the segments are with equal size, whereas in the quality direction (indexed with $n = 1,2,..,N$), the segments from the representation with a higher quality are represented with larger rectangles. The unlabeled segments are not retrieved from the server, whereas the labeled segments are downloaded (hence denoted with D) at the client in different observation slots indexed with j. The label $D-j$ indicates that the segment was downloaded in the observation slot j.

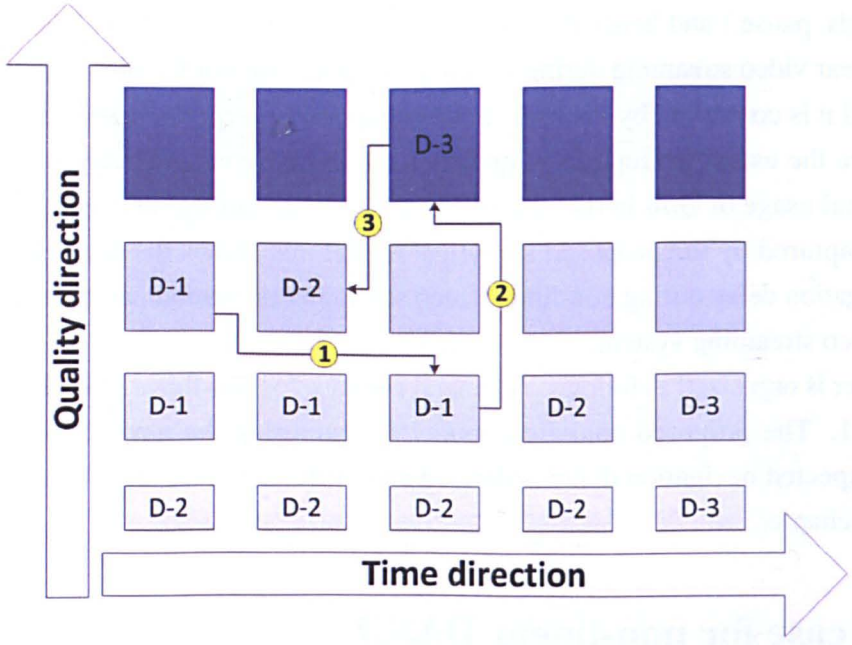


Fig. 6.1 General use case for non-linear DASH.

In the depicted example ($M = 5$ and $N = 4$), at the end of the first observation slot $j = 1$, the segments with indexes $(m = 1, n = 2)$, $(m = 1, n = 2)$, $(m = 2, n = 2)$, and $(m = 3, n = 2)$ are downloaded (label $D-1$). At the end of the observation slot, the user decides to navigate and starts the non-linear video streaming. The first navigation event (denoted with **1**) is towards a downloaded segment, hence at the end of the navigation this segment is

almost instantly played. During the playout of this segment, additional segments (this is the second observation slot, $j = 2$) with indexes $(m = 1, n = 1)$, $(m = 2, n = 1)$, $(m = 3, n = 1)$, $(m = 4, n = 1)$, $(m = 4, n = 2)$, and $(m = 2, n = 3)$ are downloaded (label $D - 2$). At the end of the second observation slot, non-linear streaming starts, i.e., the second navigation event (denoted with **2**) is towards already played segment, however the users wants to view it in a higher quality. In this case, the segment is not available at the client, hence the users waits a particular navigation delay in order to download the segment from the server. The download and the playout of this segment happens in observation slot $j = 3$ (label $D - 3$). In addition, during the playout of the segment, two additional segments with indexes $(m = 5, n = 1)$ and $(m = 5, n = 2)$ are downloaded (label $PF - 3$). At the end of the playout, non-linear streaming starts, i.e., the third navigation event (denoted with **3**) is backwards in time towards lower quality, and towards a downloaded segment, hence the segment is played instantly.

6.2 Model for non-linear DASH video streaming

Several QoE metrics and modeling aspects for DASH were discussed in the previous chapter [82, 84, 85, 88, 91, 143], and models, methods and metrics were proposed and evaluated. In these approaches, it is a common for the QoE models/metrics to express the users' quality perception through one parameter (MOS) for linear video streaming. Similar to this, the term navigation experience is introduced in this chapter in order to capture the users' perception of quality for non-linear video streaming through one parameter, i.e., the expected navigation delay.

A simple example of linear and non-linear video streaming for DASH is presented in [2] similar to the example in Section 6.1. However, in this case non-linear video streaming is achieved by a trick-mode track that is used to pause (linear video streaming is stopped) and rewind the video by navigating backwards. In the same work, more advanced use cases for DASH are listed, including switching between multiple camera views, 3D multimedia content streaming, video streams with subtitles and captions, dynamic ad insertion, etc.

The DASH industry forum recently released guidelines for implementation [145] that provide technical insights on DASH trick mode support, enabling clients to fast forward/backward, navigate/seek to a particular position, rewind and perform other navigation functions. It is recommended that additional representations should be used at different frame rate in order to support navigation while keeping the same decoding and rendering capabilities of the device. Further, it is recommended that these representations are grouped in a separate adaptation set annotated with EssentialProperty or SupplementalProperty descriptor. As such, these representations are not suitable for regular playout.

The model for non-linear DASH video streaming captures the users' perception of quality and it is applicable in any use case where navigation functions determine the selection of a particular DASH segment without the usage of additional representations at the server side. It is tailored to express the analytical dependency of the expected navigation delay on several parameters related to the DASH video streaming system.

6.2.1 Derivation and evaluation of the model

Fig. 6.2 depicts the general work-flow of the model derivation with respect to the time axis, showing the procedures and the specific actions required by the DASH client after the introduction of a navigation function. The result of the navigation function is the request of a particular video content at a specific position (forward/backward) with respect to the DASH timeline or perhaps a video content with a specific quality (low/high) with respect to the available DASH representations. Therefore, the MPD parsing procedure is performed as follows. Initially, a segment selection is needed to identify the exact segment that contains the required content and the segment's URL is obtained. Further, information on the encoding structure of the segment is required that shows the dependency between the reference and predicted frames within the GOP. As a result of the MPD parsing, the client obtains the URL and the encoding information of the required segment which are used in the next procedure, i.e., the request decision. This procedure determines the configuration of the HTTP GET request. In particular, the URL is used to specify the location of the segment at the server side, whereas the encoding information is useful in order to determine the type of the HTTP GET request in the following manner. If the segment size (T) is equal to the GOP size (G) used in the encoding ($T = G$), then full HTTP GET request should be used and the entire segment should be retrieved. However, if the segment size is larger than the GOP size used in the encoding ($T > G$) then partial HTTP GET request should be used, hence the client should use the byte range setting to configure the request. In the following subsection, the rationale behind these remarks is discussed in more details. In the end, the required content needs to be downloaded, decoded and displayed. The elapsed time duration for the client to perform the aforementioned procedures and actions is labeled as expected navigation delay and denoted with D_n.

In Annex A (part A.4) of the MPEG-DASH standard, navigation/seeking is discussed for DASH clients where the adoption of SAP is proposed. These SAPs correspond to the I frames. The playout duration of the I frames with respect to the common DASH timeline as discussed in the standard is denoted with T_{sap}. Hence, the duration T_{sap} is equal to the playout duration of a frame, i.e., $T_{\text{sap}} = \frac{1}{F_r}$, where for example $F_r = 25$ (fps) is the encoding

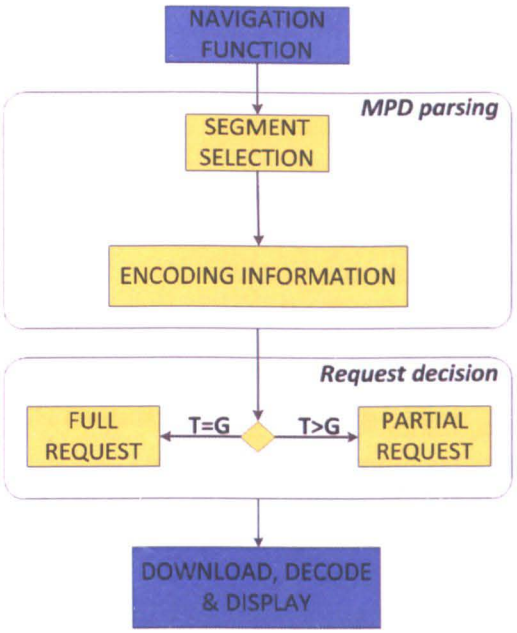


Fig. 6.2 Procedures and specific actions at the client side related to the expected navigation delay.

frame rate. It is preferable to navigate to SAPs since they enable the client to immediately access the DASH stream.

As a result of the navigation function, the following possible outcomes are considered:

- The user navigates to T_{sap} in an arbitrary segment. In this case, the client will select the appropriate segment and after downloading and decoding, it can display the appropriate I frame from the segment since the I frame is a reference frame and can be decoded without any additional information.

- The user navigates outside T_{sap} in an arbitrary segment. In this case, the client will select the appropriate segment and will need to download and decode the closest preceding I frame and the required frames (P or B) before displaying the required frame since a predicted frame can be decoded only by using additional information from the other frames.

The interval between the time instant that corresponds to the user's navigation to or outside SAP and the time instant that corresponds to the display of the frame required with the navigation is adopted as the expected navigation delay. Therefore, in the proposed model, it is assumed that the MPD parsing and request decision procedures are performed instantaneously, whereas the decoding and displaying times are negligible compared to the

Table 6.1 Parameters and notations for the model.

Parameter	Notation and used unit
Expected navigation delay	D_n (seconds)
Segment size	T (seconds)
GOP size	G (seconds)
SAP time interval	T_{sap} (seconds)
Encoding frame rate	F_r (fps)
Probability to seek to SAP	$P(\text{SAP})$ (-)
Download time, seeking to SAP	Δ (seconds)
Download time, seeking outside SAP	$\hat{\Delta}$ (seconds)
Video duration	T_v (seconds)
Total number of segments in the DASH timeline	M (-)
Number of DASH representation	N (-)
Segment index in the video sequence	m (-)
Representation index	n (-)
Bytes in the m-th segment from the n-th representation	B_m^n (-)
Bytes in the I frame of the GOP	B_I (-)
Bytes in the P frames of the GOP	B_P (-)
Bytes in the entire GOP	B_{GOP} (-)
Data rate at navigation instance	τ (kbps)
Representation bit rate for required segment	$\hat{\rho}$ (kbps)
Bytes ratio, I frame and the remaining frames	α (-)

download time. After displaying the frame required with the navigation, the downloading, decoding and displaying of the following frames can continue in a typical linear video streaming manner until the next navigation function is inserted. Parameters and notations related to the model derivation are reported in Table 6.1.

Accounting for the previous remarks, the expected navigation delay is expressed as:

$$D_n = P(\text{SAP})\Delta + (1 - P(\text{SAP}))\hat{\Delta}. \tag{6.1}$$

where $P(\text{SAP})$ is the probability for the user to navigate/seek to T_{sap} in an arbitrary segment, and Δ and $\hat{\Delta}$ are the download times when the user navigates to and outside an SAP, respectively.

Denoting the video duration as T_v, the $P(\text{SAP})$ probability is derived under the assumption that the user navigates randomly (assuming uniform distribution) in the DASH timeline, i.e., $t \sim U(o, T_v)$.

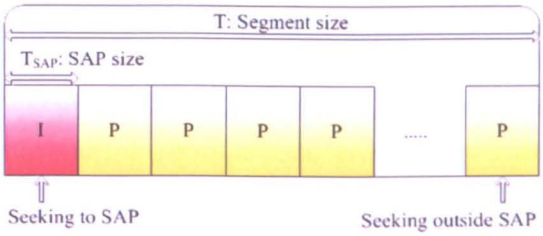


Fig. 6.3 Navigation/seeking outcomes for arbitrary DASH segment.

Considering navigation to an SAP and outside an SAP and denoting the total number of video segments in the DASH timeline with M, the sum of the partial integrals is:

$$\sum_{m=1}^{M} \left(\int_{mT}^{mT+T_{\text{sap}}} \frac{1}{T_v} dt + \int_{mT+T_{\text{sap}}}^{(m+1)T} \frac{1}{T_v} dt \right) = 1. \tag{6.2}$$

The probability to seek into an SAP is obtained as a sum of all the first term integrals:

$$P(\text{SAP}) = \sum_{m=1}^{M} \left(\int_{mT}^{mT+T_{\text{sap}}} \frac{1}{T_v} dt \right) = \frac{T_{\text{sap}}}{T}. \tag{6.3}$$

Further, the download times when navigating to and outside SAP are obtained as follows. The download time when navigating to SAP is equal to the download time of the I frame and it is simply expressed with $\Delta = \frac{B_I}{\tau}$, where B_I is frame size in bytes and τ is the data rate at the navigation instance. The download time when navigating outside SAP, in the worst case, is equal to the download time of the entire segment (the user requires the last frame of the segment) with size in bytes B_m^n (this depends on the representation index n and the segment index m in the video sequence), therefore $\hat{\Delta} = \frac{B_m^n}{\tau}$. An arbitrary IPP...P encoded segment is depicted in Fig. 6.3 showing seeking to and outside SAP, as adopted in the derivation of the model.

The following types of HTTP request can be used by the client for the download:

- *Full HTTP GET requests*. The client requests the segment from its initial SAP, i.e, the first I frame in the segment, and starts to download all the bytes from the following frames in the segment, thus reaching the required frame at the end of the segment.

- *Partial HTTP GET requests*. The client uses the byte range setting in the partial request to access the nearest SAP, i.e, the nearest I frame, and downloads all the bytes from the following frames as specified in the byte range setting, thus reaching the required frame.

The decision on the type of the HTTP GET request, as mentioned in the previous section, should depend on the encoding in order to improve performance in the following manner. For example, assuming that the segment size and the GOP size are equal ($T = G$), the download time in the case of navigating outside SAP and requiring a frame at the end of the segment (worst case) is the same for full and partial HTTP GET request, but full HTTP GET requests do not require specification of the byte range setting, and hence are simpler to use. The download time in this case is $\hat{\Delta} = \frac{B_m^n}{\tau}$. However, assumming that the segment size is larger than the GOP size ($T > G$), the download time in the case of navigating outside SAP and requiring a frame at the end of the segment is not the same for the full and the partial HTTP GET request. In this case, using partial request with specific byte range enables the client not to download the entire segment, but to download only the last GOP in the segment, thus improve performance compared to the full request (lower expected navigation delay). This is particularly useful when subsegments are used since each subsegment corresponds to a GOP and can be independently decoded. The download time in this case is $\hat{\Delta} = \frac{B_{\text{GOP}}}{\tau}$, where B_{GOP} is size of the GOP in bytes.

By adopting the previous notations and remarks, the analytical expression for the expected navigation delay is:

$$D_n = \frac{1/F_r}{T}\frac{B_I}{\tau} + (1 - \frac{1/F_r}{T})\frac{B_{\text{GOP}}}{\tau}. \tag{6.4}$$

Further, by substituting, $B_{\text{GOP}} = G\hat{\rho}$, $B_{\text{GOP}} = B_I + B_P$, and $B_P = \alpha * G\hat{\rho}$, where B_P is the number of bytes in all the P frames in the GOP, B_I number of bytes in the I frame of the corresponding GOP, $\hat{\rho}$ is the representation of the required segment for the navigation (expressed in kbps), $\hat{\rho} \in \{\rho^n, n = 1,2,..,N\}$, N is the total number of available DASH representations, and the parameter α is defined in the interval $(0,1)$. The expected navigation delay expression is in this case:

$$D_n = \frac{G\hat{\rho}}{\tau}(1 - \alpha * \frac{1/F_r}{T}). \tag{6.5}$$

It follows that the expected navigation delay shows linear dependency on the GOP size, and it further depends on the representation of the required segment for the navigation, the supported data rate at the navigation instance, the encoding frame rate, the segment size and the ratio α. By considering the expressions used in the substitution step, the α parameter can be expressed as:

$$\alpha = \frac{B_P}{B_{\text{GOP}}}. \tag{6.6}$$

This parameter depends on the content and on the encoding. However, from (6.5), it can be concluded that this ratio has a small impact on the expected navigation delay when assuming that the number of bytes in the P frames is significantly smaller compared to the number of bytes in the entire GOP. This assumption depends on the the number of P frames in the GOP and on the quantization parameter of the encoding (in some cases α could be even larger than 1). In addition to this, assuming that the bit rate of the representation of the required segment for the navigation is approximately equal to the supported data rate at the navigation instance, the ratio $\frac{\hat{\rho}}{\tau} \approx 1$, and hence the following term $\alpha * \frac{1/F_r}{T} << 1$, therefore a simple linear approximation is obtained for the expected navigation delay:

$$D_n \approx G. \tag{6.7}$$

The expected navigation delay depends linearly on the GOP size used in the encoding when the aforementioned assumptions are adopted. However, this linear approximation changes if a more aggressive or conservative approach is deployed at the client. For example, if the client requests a segment from representation with a higher bit rate compared to the supported data rate at the navigation instance, then the expected navigation delay would be larger than the GOP size (expressed in seconds). In addition, the adopted assumption $\alpha * \frac{1/F_r}{T} << 1$ depends on the encoding and segmentation and requires careful consideration since the impact of this term on the expected navigation delay might not be negligible in all cases (different encoding standards, segment sizes, content types, etc.).

Fig. 6.4 shows the results obtained from numerical simulations for the following sequences: ultrasound sequence (US), HARBOUR, CREW, SOCCER and the analytical model under the assumptions adopted in the derivation of the linear approximation. The results show the expected navigation delay as a function of the GOP size and confirm the validity of the model. Additionally, it is observed that the content of the different sequences does not significantly affect the navigation delay. Further, using small GOP size during the encoding of the segments provides lower expected navigation delay due to the usage of the partial HTTP GET request, hence better navigation experience for non-linear video streaming. Using segment size larger than the GOP size ($T > G$) decreases the content quality expressed through the PSNR in a similar manner as discussed in [54] where the impact of using different segment sizes with fixed GOP size is investigated. However, this drawback can be potentially mitigated by selecting the segment from a representation that enables sufficient content quality provided that this is supported by the network condition at the navigation instance.

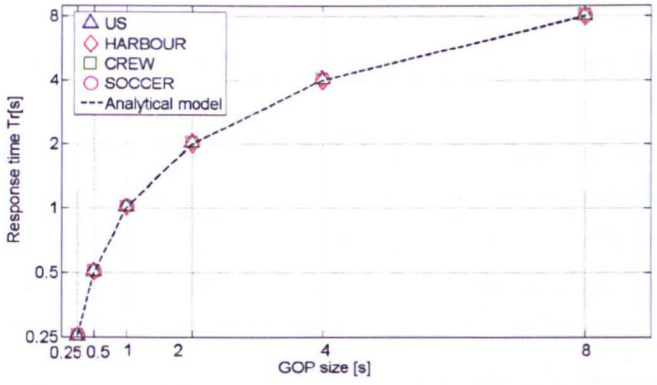


Fig. 6.4 Expected navigation delay for different GOP sizes.

6.3 Chapter summary and contribution

DASH provides an enhanced QoE for linear video streaming (display of images in order) by encoding segments in multiple bit rates and fetching segments at the appropriate bit rate, according to the supported data rate in the network. However, users can interact and control the viewing process by selecting a particular navigation function (e.g., fast forward/backwards, pause) and hence interrupt the linear video streaming. Navigation functions enable non-linear video streaming during which, the display order of the images is not known in advance. In this chapter, a model that captures the navigation experience for non-linear video streaming through the expected navigation delay is presented. The model shows the analytical dependency between the expected navigation delay and several parameters related to the DASH system. Reducing the GOP size with respect to the segment size during the preparation of the DASH dataset can improve navigation experience when partial HTTP requests are used at the client side. Further, the throughput at the navigation instance can be used to identify the representation of the required segment during the navigation, however the encoding frame rate and additional content parameter related to encoding can have significant impact on the expected navigation delay. According to the outlined assumptions used in the derivation of the model it is shown that the expected navigation delay is simply equal the GOP size.

Chapter 7

Conclusion

The growth of the adaptive HTTP video streaming portion as part of the Internet video traffic is immense in recent years; further, future projections and surveys also show that over two-thirds of the worlds mobile data traffic will be adaptive HTTP video streaming. Initially, proprietary systems from Microsoft, Adobe, Apple appeared in the market, however, nowadays all major sides in the industry are catalyzing the adoption of a general standard framework, the MPEG-DASH standard. In parallel, the adaptive HTTP video streaming research area is expanding and evolving. The works in this dissertation relate to several topics in the research area and are supported by simulations, experimental demo setups, subjective tests and analytical models. In particular, the following research topics are considered. First, the benefits and drawbacks of applying adaptive HTTP streaming system in three specific case studies are observed. The case studies consider adaptive HTTP streaming over LTE network, live video streaming over cloud, and streaming of medical videos. Second, teletraffic and statistical analysis of the client-server model as described in the MPEG-DASH standard provide quantifiable insights regarding the dependency between system, QoS, and QoE parameters. Third, the QoS and QoE for adaptive HTTP video streaming are studied by considering measurable and observable parameters. The QoS study provides the basis for developing model, method and metric related to the QoE. In the end, unlike the conventional usage of DASH for linear video streaming, the navigation experience is introduced as a term used to capture users' perception of quality for non-linear video streaming. In the end, an analytical model capturing the navigation of experience is derived.

The conclusion chapter is organized as follows. A general summary of the works in the dissertation is presented in Section 7.1, whereas plans for future work are outlined in Section 7.2

7.1 Summary of the works

The works in the dissertation are summarized with a brief description, lessons learned, and the used tools. In the end, the main contributions to knowledge from these works are discussed.

Work 1

- Brief description: A system model for adaptive video streaming from the Internet network to mobile users, i.e., users of the LTE technology in the last hop link.
- Lessons learned: The scheduling strategy has a significant effect on the observed metrics, hence it is an important procedure for improving the adaptive HTTP video streaming quality for mobile LTE users. In particular, the rebuffering probability is strongly affected, and this metric is essential for the QoE. In addition, the packet loss and packet delay parameters of the scheduling strategy require consideration during the tailoring of the strategy since the trade-off between these two parameters directly impacts the adaptive HTTP video streaming metrics. If retransmission procedure is enabled in the lower communication layers, then scheduling with low packet delay is preferable.
- Tools: This work is produced by combination of existing analytical models for TCP traffic and simulations results from the LTE simulator.

Work 2

- Brief description: A study on the QoS of adaptive HTTP video streaming for live services over cloud infrastructure with respect to the segment size selection.
- Lessons learned: The results show that the segment size is important parameter that requires consideration regarding the QoS of live adaptive HTTP video streaming over cloud infrastructure. The QoS is captured via the throughput and the packet inter-arrival time for both uplink (video source-server) and downlink (server-client) direction. In uplink, there is a statistically significant difference in the QoS with respect to the segment size, whereas in downlink, impact on the QoS is also observed.
- Tools: This work is produced by using an experimental demo setup with the Rackspace cloud server and Wowza streaming engine and real-time tracing of the results.

Work 3

- Brief description: A study on using adaptive video streaming over Internet for medical videos where the requirements for quality of medical video streaming are highlighted.

- Lessons learned: This contribution focuses on utilizing the MPEG-DASH framework for medical video streaming and provided an analysis of the quality requirements for medical video streaming with DASH. Therefore, the video encoding, segmentation and two client adaptation approaches are considered in order to analyse the re-buffering, duration of re-buffering, the added playout time and the PSNR of the streamed frames. The results show how performances depend on the selected segment size and the adaptation approach at the client.

- Tools: This work is produced by using simulations in Matlab.

Work 4

- Brief description: A model based on teletraffic queuing theory for probabilistic evaluation of adaptive HTTP video streaming where probabilistic metrics are derived and evaluated.

- Lessons learned: The MPEG-DASH functionality is mapped in teletraffic domain by using a mathematical approach based on queuing theory and relevant probability metrics are derived and investigated for different segment size of the encoded video at the server and the buffer size at the client. Furthermore, the results of this analysis are used to provide information on how to configure the HTTP/TCP communication protocol stack so that the streaming is improved.

- Tools: This work is produced experimental demo setup with the libdash library and Matlab for the teletraffic model.

Work 5

- Brief description: A methodology for applying statistical analysis for adaptive video streaming systems.

- Lessons learned: The presented methodology shows how QoS parameters which are related to the QoE can be evaluated by applying fundamental principles from statistics. As a result, relevant statistics for rebuffering and video quality parameters are obtained and transvariation-based approach is proposed for comparative analysis. The results of this analysis show how the considered parameters are affected by the segment size.

- Tools: This work is produced with Matlab simulations and the R package for the relevant statistics.

Work 6

- Brief description: A comprehensive study on QoS for adaptive video streaming of videos with respect to using different GOP sizes in the encoding and different segment sizes in the segmentation.

- Lessons learned: The study shows how the PSNR, video throughput, rebuffering frequency and MOS of DASH video streaming depend on the GOP sizes and segment sizes. The usage of smaller segment sizes enables better adaptation to the bandwidth changes, hence it improves the considered metrics. Reducing of the GOP size with respect to the segment size is not effective for small segment sizes for all the considered metrics. However, it can slightly improve the rebuffering frequency for large segment sizes, although in this case the MOS is not significantly affected. Also in terms of MOS, the subjective tests showed that using small segment sizes and GOP size equal to the segment sizes provides better results;

- Tools: This work is produced with Matlab simulations where a system model is implemented and the metrics are observed.

Work 7

- Brief description: A model and a method for capturing the quality as perceived by the users at the end of the adaptive HTTP video streaming.

- Lessons learned: Models for the MOS of DASH and video streaming at a constant bitrate are obtained, and further a method is proposed that uses the MOS model for video streaming at a constant video bitrate to express the MOS for DASH. The two models are two parameter exponential models that correlate with the MOS scores from the subjective tests with a high R-square. The method is derived under the assumption that the MOS model for video streaming at a constant bitrate is known in advance.

- Tools: This work is produced with subjective tests and the Matlab curve fitting tool.

Work 8

- Brief description: A content aware quality metric for capturing the quality for DASH video streaming.

- Lessons learned: The proposed QoE metric consider content specific, video encoding and rebuffering related and it shows that the quality of DASH video streaming depends on the content characteristics. This metric provides better estimation of the MOS score compared to the state of the art metric.

- Tools: This work is produced by using existing analytical models, subjective tests and Matlab.

Work 9

- Brief description: An analytical model for navigation experience in non-linear DASH video streaming through the expected navigation delay when a navigation event is introduced.

- Lessons learned: The expected navigation delay depends on several parameters related to the DASH system, including the GOP size used in the encoding, the segment size, the representation of the required segment, the throughput at the navigation instance, the encoding frame rate, and additional content parameter related to encoding. According to the outlined assumptions in the derivation of the model, it is shown that the expected navigation delay can be approximated with a simple linear dependency on the GOP size.

- Tools: This work is produced by analytical models, DASHEncoder, and Matlab.

These works are related to the design of content-aware and context-aware solutions for adaptive HTTP video streaming. As a result of the work in the dissertation, four major contributions to knowledge were outlined in the introductory chapter, Section 1.3. The first contribution is the novel system model for adaptive HTTP video streaming from Internet to LTE users that is based on combined usage of analytical dependencies and simulations traces. This model provides a general framework that can be easily modified and applied for a variety of use cases where DASH client uses mobile/wireless access and obtains the video content from the Internet network. The second contribution is the novel system model for probabilistic evaluation of adaptive HTTP video streaming. This model can be used for linking probabilistic performance metrics and DASH system parameters at the server and the client side. The third contribution is a content-aware quality metric for capturing the quality for DASH video streaming. This metric expresses the quality for DASH streaming by considering rebuffering, video quality and content-related parameters. In the end, the fourth contribution is a novel analytical model for navigation experience in non-linear DASH. This model can be used for expressing the expected navigation delay as a function of DASH parameters for random navigation/seeking within the DASH stream.

7.2 Future research directions

Future research directions for each of the working chapters are briefly discussed as follows.

The work presented in the chapter on DASH case studies can be further extended as follows. In the LTE study, an over-provisioning in the proposed system model can be introduced. Further, observing additional metrics related to the QoE for adaptive HTTP video streaming can provide further insights related to DASH over LTE. In the study of live adaptive HTTP video, latency measurements can provide additional indications about the quality of the live adaptive HTTP streaming session. The case study on medical video streaming using DASH, can be further upgraded by considering a specific video encoding and segmentation techniques based on diagnosis requirements and particular regions of interest in the video.

The work presented in the chapter on teletraffic and statistical analysis of DASH system can be further extended by considering different aspects from queuing theory like Markov models for multiple users and by considering real traces in the statistical analysis within the proposed model. Further, similar performances as in the proposed analysis can be observed for different adaptation algorithms at the client side so that the impact on adaptation is quantified.

The work presented in the chapter on QoS and QoE for DASH can be further extended by considering additional QoS parameters like end to end delays and by MOS modeling in the case of medical videos by supporting the model with sufficient representative sample. Further, in the MOS models, the position in the sequence where the rebuffering occurs was not considered. Intuitively, freezing of a frame from a high motion scene might be perceived differently compared to a freezing of a frame from a low motion scene. In the end, the impact of different adaptation strategies on the QoS and QoE for DASH is planned as a potential future work.

The work presented in the chapter on navigation experience for DASH can be further extended by improving the analytical model and including the required decoding time and complexity for specific GOP structures.

This dissertation presents several works and contributions related to content-aware and context-aware approaches for adaptive HTTP video streaming systems primarily based on the MPEG-DASH standard. Content-awareness refers to obtaining knowledge or cognizance regarding the video content features and availability, whereas the context-awareness refers to the design principle that adapts to the dynamic changes in the network environment. These works and contributions are supported by simulations, experimental demo setups, subjective tests and analytical modeling by considering the current state of the art compression and network solutions during the research period. However, it should be noted that the current specification of the MPEG-DASH framework is sufficiently generic to accommodate future video compression techniques and network solutions as enablers of improved quality to

bitrate ratio and high end-to-end bandwidth to users. Accounting for emerging compression and network solutions, the approaches presented in the dissertation can be modified and applied to provide insights about future designs of DASH systems.

Acronyms

3GPP 3rd Generation Partnership Project

AAU Alpen-Adria-Universität Klagenfurt

AMC Adaptive Modulation Coding

AMVSC Adaptive Mobile Video Streaming in the Cloud

ARQ Automatic Repeat Request

AVC Advanced Video Coding

CAQ Content Aware Quality

CBR Constant Bit Rate

CDF Cumulative Distribution Function

CDN Content Delivery Network

CQI Channel Quality Information

CT Computed Tomography

DASH Dynamic Adaptive Streaming over HTTP

DICOM Digital Imaging and Communications in Medicine

EXP-PF Exponential Proportional Fair

FTP File Transfer Protocol

GOP Group of Pictures

GUI Graphical User Interface

HDS HTTP Dynamic Streaming

HEVC High Efficiency Video Coding

HLS HTTP Live Streaming

HoL Head of Line

HTTP Hypertext Transfer Protocol

HTTPBis Hypertext Transfer Protocol Bis

HVS Human Visual System

IETF Internet Engineering Task Force

IP Internet Protocol

ISAVS Intelligent Bitrate Switching-based Adaptive Video Streaming

ITEC Institute of Information Technology

JPEG Joint Photographic Experts Group

LSS Live Smooth Streaming

LTE Long-Term Evolution

MAC Medium Access Control

MIME Multipurpose Internet Mail Extensions

M-LWDF Modified Largest Weighted Delay First

MOS Mean Opinion Score

MPD Media Presentation Description

MPEG Moving Picture Experts Group

MR Magnetic Resonance

MSE Mean Squared Error

NAT Network Address Translation

NRT non-Real Time

OIPF Open IPTV Forum

OSI Open Systems Interconnection

PDU Protocol Data Unit

PF Proportional Fair

PLR Packet Loss Rate

PRB Physical Resource Block

PSNR Peak Signal-to-Noise Ratio

PSQA Pseudo-Subjective Quality Assessment

PSS Packet-switched Streaming Service

QoS Quality of Service

QoE Quality of Experience

RMSE Root Mean Square Error

RT Real Time

RTCP RTP Control Protocol

RTMP Real Time Messaging Protocol

RTP Real-time Transport Protocol

RTSP Real Time Streaming Protocol

SAP Stream Access Point

SDU Service Data Unit

SI Spatial Information

SINR Signal-To-Interference-plus-Noise

SIP Session Initiation Protocol

SNR Signal-to-Noise Ratio

SVC Scalable Video Coding

TCP Transmission Control Protocol

TI Temporal Information

TS Transport Stream

UDP User Datagram Protocol

UE User Equipment

URL Uniform Resource Locator

URN Uniform Resource Number

VoD Video on Demand

VT-M-LWDF Virtual Token Modified Largest Weighted Delay First

WLAN Wireless Local Area Network

WMN Wireless Multimedia and Networking

WTCP Wireless Transmission Control Protocol

WWW World Wide Web

References

[1] A. Leon-Garcia and I. Widjaja, *Communication networks*. McGraw-Hill, Inc., 2003.

[2] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the internet," *Multimedia, IEEE*, vol. 18, no. 4, pp. 62–67, April 2011.

[3] M. Twain, *From the" London Times" of 1904*. Century Company, 1898.

[4] R. R. Schaller, "Moore's law: past, present and future," *Spectrum, IEEE*, vol. 34, no. 6, pp. 52–59, 1997.

[5] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update 2011-2016," White Paper, February 2012.

[6] ——, "Cisco visual networking index: Forecast and methodology 2013-2018," White Paper, February 2012.

[7] J. Y. B. Lee, *Scalable Continuous Media Streaming Systems: Architecture, Design, Analysis and Implementation*. John Wiley & Sons, Ltd, 2005.

[8] The TDG website, http://tdgresearch.com/.

[9] M. R. Harwell, "Research design in qualitative/quantitative/mixed methods," *The Sage handbook for research in education. 2nd ed. Los Angeles, CA: Sage*, p. 147, 2011.

[10] Mathworks, *MATLAB version 8.5.0.197613 (R2015a)*, Natick, Massachusetts, 2015.

[11] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013.

[12] DASHEncoder, itec/alpen-adria-university klagenfurt.

[13] JM, http://iphome.hhi.de/suehring/tml/.

[14] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE Cellular Systems: an Open Source Framework," in *IEEE Trans. Veh. Technol.*, vol. 60, no. 2, October, 2010, pp. 498–513.

[15] W. Simpson, *Video over IP: Internet video, H.264, P2P, web TV & streaming*. Focal Press, 2008.

[16] H. Zimmermann, "OSI reference model–ISO model of architecture for Open Systems Interconnection," *Communications, IEEE Transactions on*, vol. 28, no. 4, pp. 425–432, 1980.

[17] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol–HTTP/1.1," Tech. Rep., 1999.

[18] R. T. Fielding, T. Berners-Lee, and H. Frystyk, "Hypertext Transfer Protocol–HTTP/1.0," 1996.

[19] B. Krishnamurthy, *Web protocols and practice: HTTP/1.1, networking protocols, caching, and traffic measurement.* Addison-Wesley Professional, 2001.

[20] I. H. W. Group, "HTTP 2.0 specifications," 2013.

[21] J. Postel, "Transmission Control Protocol specification," 1981.

[22] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose, "Modeling TCP Reno performance: a simple model and its empirical validation," *IEEE/ACM Transactions on Networking (ToN)*, vol. 8, no. 2, pp. 133–145, 2000.

[23] S. Floyd, A. Gurtov, and T. Henderson, "The NewReno modification to TCP's fast recovery algorithm," 2004.

[24] Z. Wang and J. Crowcroft, "Eliminating periodic packet losses in the 4.3-Tahoe BSD TCP congestion control algorithm," *ACM SIGCOMM Computer Communication Review*, vol. 22, no. 2, pp. 9–16, 1992.

[25] E. Blanton, K. Fall, and M. Allman, "A conservative selective acknowledgment (SACK)-based loss recovery algorithm for TCP," 2003.

[26] J. S. Ahn, P. B. Danzig, Z. Liu, and L. Yan, "Evaluation of TCP Vegas: emulation and experiment," in *ACM SIGCOMM Computer Communication Review*, vol. 25, no. 4, 1995, pp. 185–195.

[27] I. Khalifa and L. Trajkovic, "An overview and comparison of analytical TCP models," in *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, vol. 5, May 2004, pp. 469–472 Vol.5.

[28] K. Fall and S. Floyd, "Simulation-based Comparisons of Tahoe, Reno and SACK TCP," *SIGCOMM Comput. Commun. Rev.*, vol. 26, no. 3, pp. 5–21, Jul. 1996.

[29] Z.-N. Li, M. S. Drew, and J. Liu, *Fundamentals of multimedia.* Springer, 2004.

[30] I. E. Richardson, *H. 264 and MPEG-4 video compression: video coding for next-generation multimedia.* John Wiley & Sons, 2004.

[31] D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications," *Commun. ACM*, vol. 34, no. 4, pp. 46–58, Apr. 1991.

[32] I. Rec, "H. 264,advanced video coding for generic audiovisual services," *ITU-T Rec. H. 264-ISO/IEC 14496-10 AVC*, 2005.

[33] S. Winkler, *Digital video quality: vision models and metrics.* John Wiley & Sons, 2005.

[34] R. Aldridge, J. Davidoff, M. Ghanbari, D. Hands, and D. Pearson, "Recency effect in the subjective assessment of digitally-coded television pictures," in *Fifth International Conference on Image Processing and its Applications*, July 1995, pp. 336–339.

[35] ITU-R, "BT. 500-13 Methodology for the subjective assessment of the quality of television pictures," Tech. Rep., 2012.

[36] ITU-T Recommendation, P.910, "Subjective video quality assessment methods for multimedia applications," International Telecommunication Union, Tech. Rep., 2008.

[37] S. Chikkerur, V. Sundaram, M. Reisslein, and L. Karam, "Objective video quality assessment methods: A classification, review, and performance comparison," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 165–182, June 2011.

[38] ITU-T, "Objective perceptual assessment of video quality: full reference television," *ITU-T Telecommunication Standardization Bureau*, 2004.

[39] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.

[40] Y. Wang, "Survey of objective video quality measurements," 2006.

[41] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," Tech. Rep., 2003.

[42] C. Huitema, "Real time control protocol (RTCP) attribute in session description protocol (SDP)," 2003.

[43] G. Thom *et al.*, "H. 323: the multimedia communications standard for local area networks," *IEEE Communications Magazine*, vol. 34, no. 12, pp. 52–56, 1996.

[44] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: session initiation protocol," Tech. Rep., 2002.

[45] S. Hull, *Content Delivery Networks: Web Switching for Security, Availability, and Speed*. McGraw-Hill, Inc., 2002.

[46] "ISO/IEC 23009-1, Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats," 2012.

[47] 3GPP, *TS 26.234: Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs*, Std. TS 26.234, 2010.

[48] M. Cedervall, U. Horn, Y. Hu, I. M. Ivars, and T. Näsström, "Open IPTV Forum–Toward an open IPTV standard," *Ericsson Review*, vol. 3, 2007.

[49] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 8," *JCTVC-J1003, July*, 2012.

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

[51] I. Irondi, Q. Wang, and C. Grecos, "Empirical evaluation of H. 265/HEVC-based dynamic adaptive video streaming over HTTP (HEVC-DASH)," in *SPIE Photonics Europe*, 2014.

[52] ——, "Subjective evaluation of H. 265/HEVC based dynamic adaptive video streaming over HTTP (HEVC-DASH)," in *IS&T/SPIE Electronic Imaging*.

[53] V. Adzic, H. Kalva, and B. Furht, "Optimizing video encoding for adaptive streaming over HTTP," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 397–403, 2012.

[54] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over HTTP dataset," in *Proceedings of the 3rd Multimedia Systems Conference*, 2012, pp. 89–94.

[55] C. Liu, I. Bouazizi, and M. Gabbouj, "Segment duration for rate adaptation of adaptive HTTP streaming," in *2011 IEEE International Conference on Multimedia and Expo (ICME)*, July 2011, pp. 1–4.

[56] I. Hofmann, N. Farber, and H. Fuchs, "A study of network performance with application to adaptive HTTP streaming," in *2011 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, June 2011, pp. 1–6.

[57] Y. Hassan, A. Helmy, and M. Rehan, "Effect of varying segment size on dash streaming quality for mobile user," in *2014 International Conference on Engineering and Technology (ICET)*, April 2014, pp. 1–4.

[58] A. Sideris, E. Markakis, N. Zotos, E. Pallis, and C. Skianis, "MPEG-DASH users' QoE: The segment duration effect," in *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, May 2015, pp. 1–6.

[59] C. Müller, D. Renzi, S. Lederer, S. Battista, and C. Timmerer, "Using scalable video coding for dynamic adaptive streaming over HTTP in mobile environments," in *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, 2012, pp. 2208–2212.

[60] 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*. ACM, 2012, pp. 37–42.

[61] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. A. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," *SIGCOMM-Computer Communication Review*, vol. 41, no. 4, p. 362, 2011.

[62] T. Kupka, P. Halvorsen, and C. Griwodz, "An evaluation of live adaptive HTTP segment streaming request strategies," in *2011 IEEE 36th Conference on Local Computer Networks (LCN)*, 2011, pp. 604–612.

[63] T. Lohmar, T. Einarsson, P. Frojdh, F. Gabin, and M. Kampmann, "Dynamic adaptive HTTP streaming of live content," in *2011 IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, 2011, pp. 1–8.

[64] X. Qiu, H. Liu, D. Li, S. Zhang, D. Ghosal, and B. Mukherjee, "Optimizing HTTP-based Adaptive Video Streaming for wireless access networks," in *3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*. IEEE, 2010, pp. 838–845.

[65] C. Liu, I. Bouazizi, and M. Gabbouj, "Parallel adaptive HTTP media streaming," in *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, 2011, pp. 1–6.

[66] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, "Adaptation algorithm for adaptive streaming over HTTP," in *19th International Packet Video Workshop (PV)*, Munich, 2012, pp. 173–178.

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

[68] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. Begen, and D. Oran, "Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719–733, April 2014.

[69] C. Zhou, C.-W. Lin, X. Zhang, and Z. Guo, "A Control-Theoretic Approach to Rate Adaption for DASH Over Multiple Content Distribution Servers," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 681–694, April 2014.

[70] 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*, 2012, pp. 97–108.

[71] S. Hu, L. Sun, C. Gui, E. Jammeh, and I.-H. Mkwawa, "Content-aware adaptation scheme for qoe optimized dashf applications," in *2014 IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 1336–1341.

[72] A. Khan, L. Sun, E. Jammeh, and E. Ifeachor, "Quality of experience-driven adaptation scheme for video applications over wireless networks," *IET communications*, vol. 4, no. 11, pp. 1337–1347, 2010.

[73] M. Claeys, S. Latré, J. Famaey, and F. De Turck, "Design and evaluation of a self-learning HTTP adaptive video streaming clienft," *IEEE Communications Letters*, vol. 18, no. 4, pp. 716–719, 2014.

[74] 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 (PV)*. IEEE, 2013, pp. 1–8.

[75] D. Suh, I. Jang, and S. Pack, "QoE-enhanced adaptation algorithm over DASH for multimedia streaming," in *International Conference on Information Networking (ICOIN)*. IEEE, 2014, pp. 497–501.

[76] R. K. Mok, X. Luo, E. W. Chan, and R. K. Chang, "QDASH: a QoE-aware DASH system," in *Proceedings of the 3rd Multimedia Systems Conference*, 2012, pp. 11–22.

[77] S. Garcia, J. Cabrera, and N. Garcia, "Quality-optimization algorithm based on stochastic dynamic programming for MPEG DASH video streaming," in *2014 IEEE International Conference on Consumer Electronics (ICCE)*, 2014, pp. 574–575.

[78] M. Claeys, S. Latré, J. Famaey, T. Wu, W. Van Leekwijck, and F. De Turck, "Design of a Q-learning-based client quality selection algorithm for HTTP adaptive video streaming," in *Adaptive and Learning Agents Workshop, part of AAMAS2013 (ALA-2013)*, 2013, pp. 30–37.

[79] T. Kim and M. H. Ammar, "Receiver buffer requirement for video streaming over tcp," in *Electronic Imaging 2006*. International Society for Optics and Photonics, 2006, pp. 607 718–607 718.

[80] W. Pu, Z. Zou, and C. W. Chen, "Video adaptation proxy for wireless dynamic adaptive streaming over http," in *Packet Video Workshop (PV), 2012 19th International*. IEEE, 2012, pp. 65–70.

[81] R. Mok, E. Chan, and R. Chang, "Measuring the quality of experience of HTTP video streaming," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2011, pp. 485–492.

[82] K. Singh, Y. Hadjadj-Aoul, and G. Rubino, "Quality of experience estimation for adaptive HTTP/TCP video streaming using H.264/AVC," in *IEEE Consumer Communications and Networking Conference (CCNC)*, Jan 2012, pp. 127–131.

[83] S. Tavakoli, J. Gutierrez, and N. Garcia, "Subjective quality study of adaptive streaming of monoscopic and stereoscopic video," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 684–692, April 2014.

[84] C. Alberti, D. Renzi, C. Timmerer, C. Mueller, S. Lederer, S. Battista, and M. Mattavelli, "Automated QoE evaluation of Dynamic Adaptive Streaming over HTTP," in *Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*, July 2013, pp. 58–63.

[85] C. Chen, L. K. Choi, G. de Veciana, C. Caramanis, R. Heath, and A. 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 (ICASSP)*, May 2013, pp. 3602–3606.

[86] M. Eckert, T. M. Knoll, and F. Schlegel, "Advanced MOS calculation for network based QoE Estimation of TCP streamed Video Services," in *7th International Conference on Signal Processing and Communication Systems (ICSPCS)*. IEEE, 2013, pp. 1–9.

[87] D. Z. Rodriguez, J. Abrahao, D. C. Begazo, R. L. Rosa, and G. Bressan, "Quality metric to assess video streaming service over tcp considering temporal location of pauses," vol. 58, no. 3, 2012, pp. 985–992.

[88] D. Z. Rodriguez, R. L. Rosa, and G. Bressan, "No-reference video quality metric for streaming service using DASH standard," in *IEEE International Conference on Consumer Electronics (ICCE)*, 2015, pp. 106–107.

[89] S. Garcia Lobo, J. Cabrera, and N. García, "Quality-control algorithm for adaptive streaming services over wireless channels," pp. 50–59, 2015.

[90] G. Papadogiannopoulos, I. Politis, T. Dagiuklas, and L. Dounis, "Perceptual quality assessment of HTTP adaptive video streaming," in *IEEE Globecom Workshops (GC Wkshps)*, 2013, pp. 1174–1179.

[91] Y. Liu, S. Dey, D. Gillies, F. Ulupinar, and M. Luby, "User Experience Modeling for DASH Video," in *20th International Packet Video Workshop (PV)*. IEEE, 2013, pp. 1–8.

[92] J. De Vriendt, D. De Vleeschauwer, and D. Robinson, "Model for estimating QoE of video delivered using HTTP adaptive streaming," in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. IEEE, 2013, pp. 1288–1293.

[93] B. Krogfoss, A. Agrawal, and L. Sofman, "Analytical method for objective scoring of HTTP Adaptive Streaming (HAS)," in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2012, pp. 1–6.

[94] Y. Xu, Y. Zhou, and D.-M. Chiu, "Analytical QoE models for bit-rate switching in dynamic adaptive streaming systems," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2734–2748, 2014.

[95] W. Tu, E. Steinbach, M. Muhammad, and X. Li, "Proxy Caching for Video-on-Demand Using Flexible Starting Point Selection," *IEEE Transactions on Multimedia*, vol. 11, no. 4, pp. 716–729, June 2009.

[96] H. Tani and T. Nunome, "QoE assessment of a web-based streaming service in seeking operation," in *International Conference on Information Networking (ICOIN)*, Feb 2014, pp. 254–259.

[97] J. Paluska and H. Pham, "Interactive streaming of structured data," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, March 2010, pp. 11–19.

[98] B. Gao, J. Jansen, P. Cesar, and D. C. Bulterman, "Accurate and low-delay seeking within and across mash-ups of highly-compressed videos," in *Proceedings of the 21st international workshop on Network and operating systems support for digital audio and video*, 2011, pp. 105–110.

[99] V. Krishnamoorthi, P. Bergström, N. Carlsson, D. Eager, A. Mahanti, and N. Shahmehri, "Empowering the creative user: personalized HTTP-based adaptive streaming of multi-path nonlinear video," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 53–58, 2013.

[100] V. Krishnamoorthi, N. Carlsson, D. Eager, A. Mahanti, and N. Shahmehri, "Quality-adaptive prefetching for interactive branched video using HTTP-based Adaptive Streaming," in *Proceedings of the ACM International Conference on Multimedia*, 2014, pp. 317–326.

[101] R. Kuschnig, I. Kofler, and H. Hellwagner, "An evaluation of TCP-based rate-control algorithms for adaptive internet streaming of H.264/SVC," in *Proceedings of the first annual ACM SIGMM conference on Multimedia systems*, 2010, pp. 157–168.

[102] C. Müller, D. Renzi, S. Lederer, S. Battista, and C. Timmerer, "Using scalable video coding for dynamic adaptive streaming over HTTP in mobile environments," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Bucharest, 2012, pp. 2208–2212.

[103] 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*, New York, NY, USA, 2012, pp. 37–42.

[104] O. Oyman and S. Singh, "Quality of experience for HTTP adaptive streaming services," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 20–27, April 2012.

[105] ——, "On capacity-quality tradeoffs in HTTP adaptive streaming over LTE networks," in *Information Theory and Applications Workshop (ITA)*, San Diego, CA, 5-10 Feb. 2012, pp. 54–55.

[106] J. Foerster, O. Oyman, Y. Liao, M. Rehan, and W. Taie, "Enhanced adaptive streaming over LTE-advanced wireless networks," in *Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Pacific Grove, CA, 2012, pp. 915–919.

[107] W. Pu, Z. Zou, and C. W. Chen, "Dynamic adaptive streaming over HTTP from multiple content distribution servers," in *IEEE Global Telecommunications Conference (GLOBECOM)*, 2011, pp. 1–5.

[108] ——, "Video adaptation proxy for wireless Dynamic Adaptive Streaming over HTTP," in *19th International Packet Video Workshop (PV)*, Munich, 2012, pp. 65–70.

[109] A. E. 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)*, 2013, pp. 2480–2485.

[110] S. Khan, D. Schroeder, A. El Essaili, and E. Steinbach, "Energy-efficient and QoE-driven adaptive HTTP streaming over LTE," in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, April 2014, pp. 2354–2359.

[111] M. Radosavljevic, S. Sladojevic, D. Culibrk, and D. Vukobratovic, "QoE-aware Rate-Conservative dynamic HTTP streaming over mobile cellular networks," in *IEEE Network Operations and Management Symposium (NOMS)*, May 2014, pp. 1–6.

[112] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, and M. Chiang, "A scheduling framework for adaptive video delivery over cellular networks," in *Proceedings of the 19th Annual International Conference on Mobile computing & networking*, 2013, pp. 389–400.

[113] 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 *Communications (ICC), IEEE International Conference on*, June 2014, pp. 1681–1686.

[114] S. Cicalo, N. Changuel, R. Miller, B. Sayadi, and V. Tralli, "Quality-fair HTTP adaptive streaming over LTE network," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 714–718.

[115] J. G. Choi and S. Bahk, "Cell-throughput analysis of the proportional fair scheduler in the single-cell environment," *IEEE Trans. Veh. Technol.*, vol. 56, no. 2, pp. 766–778, Mar, 2007.

[116] R. Basukala, H. M. Ramli, and K. Sandrasegaran, "Performance analysis of EXP/PF and M-LWDF in downlink 3GPP LTE system," in *IEEE F. Asian Himalayas Conf.*

[117] M. M. Nasralla and M. G. Martini, "A downlink scheduling approach for balancing QoS in LTE wireless networks," in *IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, London, United Kingdom, 2013, pp. 1571–1575.

[118] M. Iturralde, A. Wei, T. Yahiya, and A. Beylot, " Performance study of multimedia services using virtual token mechanism for resource allocation in LTE networks." in *IEEE Vehicular Technology Conference (VTC)*, Sept. 2011, pp. 1–5.

[119] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 4, pp. 303–314, Oct. 1998.

[120] M. Chen, "AMVSC: A framework of adaptive mobile video streaming in the cloud," in *IEEE Global Communications Conference (GLOBECOM)*, Dec 2012, pp. 2042–2047.

[121] C. Wang and M. Zink, "On the Feasibility of DASH Streaming in the Cloud," in *Proceedings of Network and Operating System Support on Digital Audio and Video Workshop*, 2014, pp. 49:49–49:54.

[122] G. Xi, X. Zhang, and K. Qu, "On adaptive live streaming in mobile cloud computing environments with D2D cooperation," in *21st International Conference on Telecommunications (ICT)*, May 2014, pp. 405–409.

[123] H. Ma, B. Seo, and R. Zimmermann, "Dynamic Scheduling on Video Transcoding for MPEG DASH in the Cloud Environment," in *Proceedings of the 5th ACM Multimedia Systems Conference*, 2014, pp. 283–294.

[124] D. Won, Y. Cho, K. Park, and D. Y. Suh, "High resolution video streaming method by cloud and DASH," in *IEEE International Conference on Consumer Electronics (ICCE)*, Jan 2014, pp. 552–553.

[125] D. K. Krishnappa, M. Zink, and R. K. Sitaraman, "Optimizing the video transcoding workflow in content delivery networks," in *Proceedings of the 6th ACM Multimedia Systems Conference*, 2015, pp. 37–48.

[126] A. Kantarcı, "Bandwidth-effective streaming of educational medical videos," *Multimedia systems*, vol. 16, no. 6, pp. 381–397, Nov. 2010.

[127] M. G. Martini, "Wireless broadband multimedia health services: current status and emerging concepts," in *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sept. 2008, pp. 1–6.

[128] M. G. Martini and C. T. E. R. Hewage, "Flexible macroblock ordering for context-aware ultrasound video transmission over mobile WiMAX," *Int. J. Telemedicine and Appl.*, vol. 2010, pp. 1–7, Jan. 2010.

[129] J. Li, Z. Chen, and W. Li, "A multimedia telemedicine system," in *International Conference on Biomedical Engineering and Computer Science (ICBECS)*, 2010, pp. 1–3.

[130] X. M. Zhang and N. Zhang, "An open, secure and flexible platform based on internet of things and cloud computing for ambient aiding living and telemedicine," in *2011 International Conference on Computer and Management (CAMAN)*, 2011, pp. 1–4.

[131] M.-C. Lin, L.-R. Dung, and P.-K. Weng, "An ultra-low-power image compressor for capsule endoscope," *BioMedical Engineering OnLine*, vol. 5, no. 1, p. 14, 2006.

[132] R. S. Istepanian, N. Y. Philip, and M. G. Martini, "Medical QoS provision based on reinforcement learning in ultrasound streaming over 3.5 G wireless systems," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 4, pp. 566–574, 2009.

[133] M. Pantziaris, "Open-source telemedicine platform for wireless medical video communication," *International Journal of Telemedicine and Applications*, vol. 2013, 2013.

[134] A. Panayides, M. Pattichis, C. Pattichis, C. Loizou, M. Pantziaris, and A. Pitsillides, "Atherosclerotic plaque ultrasound video encoding, wireless transmission, and quality assessment using H.264," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 3, pp. 387–397, 2011.

[135] M. Schwartz, *Telecommunication networks: protocols, modeling and analysis*. Addison-Wesley Reading, 1987, vol. 7.

[136] V. B. Iversen, *Teletraffic Engineering and Network Planning*. Lecture Notes, Technical University of Denmark, 2010.

[137] The ITEC-DASH website, http://www-itec.uni-klu.ac.at/dash/. [Online]. Available: http://www-itec.uni-klu.ac.at/dash/

[138] C. Gini, "Il concetto di transvariazione e le sue prime applicazioni," *Giornale degli Economisti e Rivista di Statistica, In: C. Gini (ed.)*, pp. 21–44, 1959.

[139] I. Recommendation, "E. 800: Terms and definitions related to quality of service and network performance including dependability," *ITU-T August*, vol. 1994, 1994.

[140] I. Rec, "P. 10: Vocabulary for performance and quality of service, Amendment 2: New definitions for inclusion in Recommendation ITU-T P. 10/G. 100," *Int. Telecomm. Union, Geneva*, 2008.

[141] M. G. Martini, C. T. Hewage, M. M. Nasralla, and O. Ognenoski, "QoE Control, Monitoring, and Management Strategies," in *Multimedia Quality of Experience (QoE): Current Status and Future Requirements*, 1st ed., C. W. Chen, P. Chatzimisios, T. Dagiuklas, and L. Atzori, Eds. John Wiley & Sons, 2014, ch. 7.

[142] F. De Simone and F. Dufaux, "Comparison of DASH adaptation strategies based on bitrate and quality signalling," in *IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, 2013, pp. 087–092.

[143] T. De Pessemier, K. De Moor, W. Joseph, L. De Marez, and L. Martens, "Quantifying the Influence of Rebuffering Interruptions on the User's Quality of Experience During Mobile Video Watching," pp. 47–61, March 2013.

[144] Y.-F. Ou, Z. Ma, T. Liu, and Y. Wang, "Perceptual quality assessment of video considering both frame rate and quantization artifacts," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 3, pp. 286–298, 2011.

[145] DASH Industry Forum, "Guidelines for implementation: DASH-IF Interoperability Points," April, 2015.

Appendix A: List of publications

(i) O. Ognenoski, C.T. Hewage, M.G Martini, "QoS for Adaptive HTTP video streaming over cloud", submitted to International conference.

(ii) O. Ognenoski, S. Pezzulli, M.G. Martini, P. Amon, "Statistical analysis of rebuffering events and video quality for MPEG-DASH", submitted to International conference.

(iii) O. Ognenoski, M.G.Martini, P. Amon, "A study on QoS and QoE for adaptive HTTP video streaming", submitted to IEEE Journal.

(iv) O. Ognenoski, M.G.Martini, P.Amon, "Content-aware quality metric for MPEG-DASH", submitted to IEEE Journal.

(v) O. Ognenoski, M.G. Martini, P.Amon, "An analytical model for navigation experience in non-linear adaptive HTTP streaming", submitted to IEEE letters.

(vi) M.G. Martini, C.T. Hewage, M. M. Nasralla, O. Ognenoski, Chapter "QoE Control, Monitoring and Management Strategies" in 'Multimedia Quality of Experience (QoE): Current Status and Future Requirements', Edited by Atzori, L.; Dagiuklas, T.; Chatzimisios, P.; Chen, C., Wiley. (2014)

(vii) O. Ognenoski, M. M. Nasralla, M. Razaak, M.G. Martini, P. Amon, "DASH-based video transmission over LTE networks", IEEE International Conference on Communications (ICC 2015) - Workshop on Quality of Experience-based Management for Future Internet Applications and Services (QoE-FI), Jun 8-12, London, UK. (2015)

(viii) O. Ognenoski, M. Razaak, M.G. Martini, P. Amon, "Medical Video Streaming Utilizing MPEG-DASH", IEEE Healthcom - Mobile Medical Imaging Workshop (MMIW), Oct 9-12 , Lisbon, Portugal. (2013)

(ix) O. Ognenoski, M.G. Martini, P. Amon, "Segment-based Teletraffic Model for MPEG-DASH", 15th IEEE International Workshop on Multimedia Signal Processing , IEEE, Sept. 30 - Oct. 2, Pula (Sardinia), Italy. (2013)

(x) M. M. Nasralla, O. Ognenoski, M.G. Martini, "Bandwidth Scalability and Efficient 2D and 3D Video Transmission over LTE Networks", IEEE ICC'13 - Workshop on Immersive & Interactive Multimedia Communications over the Future Internet, IEEE , June 9-13, Budapest, Hungary. (2013)