DASH-based Network Performance-aware Solution for Personalised Video Delivery Systems

Lejla Rovcanin

PhD 2016

DASH-based Network Performance-aware Solution for Personalised Video Delivery Systems

Lejla Rovcanin

M.Sc. (Res) Computer Applications, B.Eng.

Thesis Submitted for the Award of

Doctor of Philosophy (Ph.D.)

to

School of Electronic Engineering

Dublin City University

Supervisor: Dr. Gabriel-Miro Muntean

January 2016

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work

Signed: ____________

ID No.: 96971631

Date: 10/01/2016

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abbreviations

3GPP: Third Generation Partnership Project

AE: Adaptation Engine

AEH: Adaptive Educational Hypermedia

AH: Adaptive Hypermedia

AHA!: Adaptive hypermedia Architecture

AHS: Adaptive Hypermedia System

AM: Adaptation Model

APL: Adaptive Personalised Learning

AVC: Advanced Video Coding (H.264/MPEG-4 Part 10)

CC/PP: W3C's Composite Capabilities/Preference Profile

CDN: Content Delivery Network

DASH: Dynamic Adaptive Streaming over HTTP

DAV: DASH-based performance oriented Adaptive Video distribution solution

DCCP: Datagram Congestion Control Protocol

DE: Distance Education

DER: Digital Educational Repository

dLOR: digital Learning Object Repository

DM: Domain Model

DNS: Domain Name System

DPEA: DASH-based Performance Enhancement Architecture

dPOAA PE: dPOAA Performance Engine

dPOAA PM: dPOAA Performance Model

dPOAA: DASH-based Performance Oriented Adaptation Agent

DRM: Digital Rights Management

ETV: Educational TV

FTP: File Transfer Protocol

GM: Goal and Constraints Model

GOP: Group of Pictures

HAS: HTTP Adaptive Streaming

HD: High Definition

HDS: Adobe Dynamic HTTP Streaming

HEVC: High Efficiency Video Coding (MPEG-H Part 2/H.265)

HLS: Apple HTTP Live Streaming

HTTP: Hypertext Transport Protocol

ICT: Information and Communication Technology

IEC: International Electrotechnical Commission

IEEE: Institute of Electrical and Electronics Engineers

IETF: Internet Engineering Task Force

IP: Internet Protocol

ISO BMFF: ISO/IEC 14496-12 ISO Base Media File Format

ISO: International Organization for Standardization

ISP: Internet Service Provider

ITU: International Telecommunication Union

ITU-T: International Telecommunication Union - Telecommunication Standardization Sector

JPEG: Joint Photographic Experts Group

KE: Knowledge Elements

LAN: Local Area Network

LMS: Learning Management System

LO: Learning Object

LOM: Learning Object Metadata

MAC: Medium Access Control

MOOC: Massive Open Online Courses

MOS: Mean Opinion Score

MPD: Media Presentation Description

MPEG: Moving Picture Experts Group

MPEG-2 TS: ISO/IEC 13818-1 MPEG-2 Transport Stream

MPEG-DASH: MPEG Dynamic Adaptive Streaming over HTTP

MPQM: Moving Picture Quality Metric

MPTCP: Multi-Path Transmission Control Protocol

NAT: Network Address Translation

NS-2: Network Simulator 2

NS-3: Network Simulator 3

OAEHS: Open Adaptive Educational Hypermedia System

oPL: open corpus adaptive Personalised Learning

oPOAA CM: oPOAA Content Model

oPOAA PE: oPOAA Performance Engine

oPOAA PM: oPOAA Performance Model

oPOAA: open Performance Oriented Adaptation Agent

P2P: Peer to Peer

PL System: Personalised Learning System

PM: Presentation Model

PSNR: Peak-Signal-to-Noise-Ratio

PVQM: Perceptual Video Quality Measure

QoE: Quality of Experience

QoEAHA PM: Performance Monitor

QoEAHA PPM: Perceived Performance Model

QoP: Quality of Perception

QoS: Quality of Service

RDF: Resource Description Framework

RTCP: RTP Control Protocol

RTP: Real Time Transport Protocol

RTSP: Real Time Streaming Protocol

RTT: Round-Trip Time

SAP: Stream Access Point

SCORM: Sharable Content Object Reference model

SCTP: Stream Control Transmission Protocol

SLA: Service Level Agreement

SNMP: Simple Network Management Protocol

SNR: Signal to Noise Ratio

SSIM: Structural Similarity Index

TCP: Transmission Control Protocol

UAProf: User Agent Profile

UDP: User Datagram Protocol

UM: User Model

UPnP: Microsoft proposed Universal Plug and Play

URI: Uniform Resource Identifier

URL: Uniform Resource Locator

UTC: Coordinated Universal Time

VoD: Video on Demand

VQM: Video Quality Metric

W3C: World Wide Web Consortium

WAN: Wide Area Network

WAP: Wireless Application Protocol

WLAN: Wireless Local Area Networks

WML: Wireless Markup Language

WURFL: Wireless Universal Resource File

WWW: World Wide Web

XML: eXtensible Markup Language

Abstract

Video content is an increasingly prevalent contributor of Internet traffic. The proliferation of available video content has been fuelled by both Internet expansion and the growing power and affordability of viewing devices. Such content can be consumed anywhere and anytime, using a variety of technologies. The high data rates required for streaming video content and the large volume of requests for such content degrade network performance when devices compete for finite network bandwidth. The results are prolonged startup delays and frequent stops for rebuffering during video playout. Such effects are especially significant for third level educational settings where, on-demand access to high quality educational video content by on-campus students is an increasingly important requirement. Although purely online courses are attracting growing interest traditional campus-based classes remain large. In the latter setting, frequently large numbers of students may simultaneously request identical video content.

Adaptive HTTP-based streaming technologies such as DASH introduce client-controlled delivery of video in order to dynamically adapt to varying bandwidth and viewing device characteristics. However, although DASH allows for individual clients to adapt to network conditions it does not support multiple local clients in co-ordinating their actions. Thus, despite DASH-aware devices, problems remain when numerous local clients simultaneously request high bandwidth video.

This thesis addresses the problem of quality degradation in personalised video delivery by developing mechanisms which raise video quality levels in a campus setting. A DASH-based Performance Enhancement Architecture (DPEA) is proposed to enhance the performance of existing personalised systems. Under DPEA, the quality of the delivered video is increased by deploying a Performance Oriented Adaptation Agent (POAA) that considers the characteristics of the links connecting video providers and the campus network in order to select remote servers with the best current performance. Furthermore, this solution proposes a DASH-based Adaptive Video Distribution Solution (DAV) which considers both device characteristics and recently downloaded (locally available) video segments in order to improve the content delivery process thereby improving the video viewing experience. The proposed solutions maintain satisfactory quality levels when multiple requests for identical video content are generated in an on-campus setting. The solutions are evaluated by simulations in which various network parameters are considered. The results clearly demonstrate improved video quality when the proposed solutions are deployed.

Acknowledgement

The completion of this thesis has been made possible through the encouragement, expert guidance and feedback of my supervisor, Dr. Gabriel-Miro Muntean. I am grateful for his kindness, patience and enthusiasm, and for the time and effort he has devoted towards my research.

I am also grateful to my examiners, Dr. Gheorghita Ghinea and Dr. Claus Pahl for their constructive feedback.

I thank DCU and DIT for providing the educational and financial support for this research.

I benefited from membership of the DCU Performance Engineering Laboratory where I befriended talented researchers.

I am forever grateful to my family and friends who unconditionally supported me throughout this process.

Dedication

Mojoj porodici.

1 Introduction

Demand for relevant digital video content is increasing [1]. We are witnessing a strong trend towards online publishing of free multimedia content (e.g. [2], [3] in the context of higher education). With the proliferation of multimedia (video, audio, image) recording devices, generating content "on the fly" has become part of everyday life, where large quantities of digital video are generated, stored and shared free of charge. Online learning systems follow this trend and facilitate the use of media-rich learning content and of video/audio streaming. For example, students employ diverse mobile devices (tablets, smartphones, laptops) [4], [5] and demand diverse media as part of the educational process [6]. Notwithstanding the popularity of mobile devices, more than half (56% [4]) of students use desktop computers, either personal or college-provided. A fast and reliable Internet connection from such devices could be the reason for the lasting popularity of institution-provisioned hardware [4]. At the same time, online, blended and technology-enabled learning is growing increasingly popular [7]. However, learners are less likely to adopt an online activity if they consider it too slow [8]. Furthermore, university classes are growing larger, where hundreds of students with similar/identical learning needs will live/study on campus. Such students frequently request identical educational video content, e.g. in-class activities involving the analysis of the content of video clips.

The aim of the work described in this thesis is to improve access to rich media content for all users regardless of delivery network conditions. The context for this research is Adaptive Personalised Systems, such as Personalised Learning (PL) systems, that adjust their content and/or presentation to match a learner's needs and that source their learning content from open corpora (consisting of distributed and remote servers including digital educational repositories and/or the WWW in general). We propose a Performance Enhancement Architecture (DPEA) which combines two novel solutions that provide better video quality and lead to an improved viewing experience:

- Performance Oriented Adaptation Agent (POAA): Open POAA (**oPOAA**) enhances the content selection process for open corpus PL (oPL) systems and DASH-based (**dPOAA**) provides server selection for DASH enabled systems;
- DASH-based performance oriented Adaptive Video distribution solution (**DAV**): enables utilisation of portions of content available within the campus area network.

1.1 Research Motivation

Despite continued developments and existing high capacity communication networks, Web users continue to discover new network intensive applications that consume Internet resources and their expectations continue to outpace the provision of infrastructure [9]. The focus of this research is on mitigating the limitations of the infrastructure with no additional hardware investment, more specifically, efficiently exploiting the communications facet of the campus environment in which the personalised video delivery system is deployed.

Production of video content is becoming less expensive and more efficient (self-recorded user videos, etc.) and is easily made available online. For example, there are already many offerings of free educational video content such as edX [2] and Coursera [2]. Expanding university classes frequently involve students using university-provided computers and personal portable devices (smartphones, netbooks, tablets, etc.) to gain access to multimedia-based learning content. As the number of videos available for download or streaming grows rapidly, not all viewing devices (e.g. handheld devices) are capable of receiving, storing and playing the same (large) volume of video content at the highest quality. However, users expect steady non-interrupted streaming of video data regardless of viewing device type and network delivery characteristics.

There is a clear need to adjust video content selection and to adapt delivery in response to both prevailing network conditions and device characteristics in order to improve user-perceived video quality levels and to make a positive impact on the overall viewing experience. This is particularly important in an educational setting, as video viewing is a growing element of the learning process.

1.1.1 Example

An example of a campus setting is a metropolitan university that utilises educational multimedia content that is freely available online. Students interact with a PL system that maintains user models and tailors learning content and its presentation to suit the students' learning styles. Apart from closed corpus content hosted by the PL system servers, the system provides access to open corpus content (hosted by distributed, geographically remote servers, where multiple servers possibly host the same learning content). This university is learner-centred, providing interactive and collaborative learning opportunities in and out of the classroom setting. During a typical lecture/tutorial students are asked to interact with online learning resources and learn new concepts through collaborative activities, such as in-class discussions. A group of 300 students within a classroom (as indicated in Figure 1-1 on page 3) is asked to watch an educational video clip. The PL system selects the video based on student learning profile, and a group of 30 students watch the same video using their viewing devices and/or university-provided computers. The high data rate of the video content and the large number of viewers impose high demands on the delivery network, which may result in long delays, frequent stalls for rebuffering, etc. negatively impacting the viewing experience.

With the deployment of the DPEA architecture proposed in this research, the students receive higher quality video content with reduced interruptions for buffering. This in turn, ensures a better overall viewing experience.

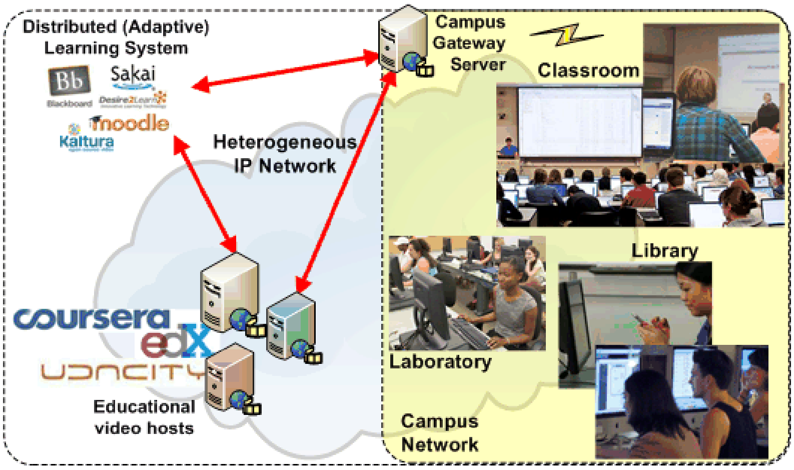

Figure 1-1: Video Delivery in a University Campus Setting

1.2 Problem Statement

Network conditions depend not only on the location of the user but also on the number of other network users and on the demands imposed by their online activity. For example, university classes frequently involve hundreds of students with similar learning needs. Thus a coincidence of demand for similar video content often emerges within the campus network. Network traffic is not typically evenly distributed over time i.e. there are periods when a large number of requests for similar (sometimes identical) high-bandwidth content are made, e.g. when a large class of students is required to view a video clip recommended by a PL system. At such times, a burst of requests is transmitted and the same video content is delivered repeatedly, overburdening the campus network and the communication link between the campus and the Internet. Therefore, the required network resources cannot be consistently guaranteed across the campus network. Poor network conditions result in long delays and frequent rebuffering. Such phenomena lead to a poor quality video viewing experience that negatively impacts on learning outcomes [10] and may ultimately result in increased drop-out rates [11]. In contrast, the university-provisioned and well-resourced devices and campus-wide network are under-utilised. A campus network typically consists of multiple interconnected Local Area Networks (LAN) with a shared Internet connection in a limited geographical area. Communication between the nodes on such LANs exhibits high bandwidth and low delay. Therefore, terms *campus network* and *LAN* are used interchangeably in this thesis.

However, the issues (e.g. delays) with video delivery are not limited to the campus Internet connection bottleneck as video delivery also depends on the hosting server (e.g. quality of Internet connection link, server response time, etc.). Frequently identical video content resides on multiple remote servers, and recent standards allow the specification of multiple hosting servers. Better quality of content could be delivered if the best performing hosting servers were selected. Furthermore, campus based well-resourced devices can be used to provide video content segments for other viewing devices within the campus network, thus significantly reducing the number of requests sent to remote servers hosting requested videos, and consequently improving the quality of delivered content.

This thesis investigates the issues associated with the personalised systems and answers the following question:

What actions can be taken to reduce the negative effects that congestion on the underlying best-effort delivery network has on the viewing experience in personalised systems providing external content to a campus area network?

Specifically, the following aspects were investigated in detail:

1.2.1. How can better video quality be obtained when video content resides on multiple remote servers?

1.2.2. How can video streaming be improved using video content available within the campus network? How can new standards for Internet video delivery be best utilised in this context?

The proposed solutions require a personalised system and are contextualised in a university campus setting, but they could be applied to other situations where a coincidence of demands for similar video content emerges in a large group of users interacting with a personalised system, such as a personalised video retrieval system in a corporate network or at a public performance.

1.3 Proposed Solutions

This section outlines the proposed solutions for improving the quality of delivery of multimedia learning content.

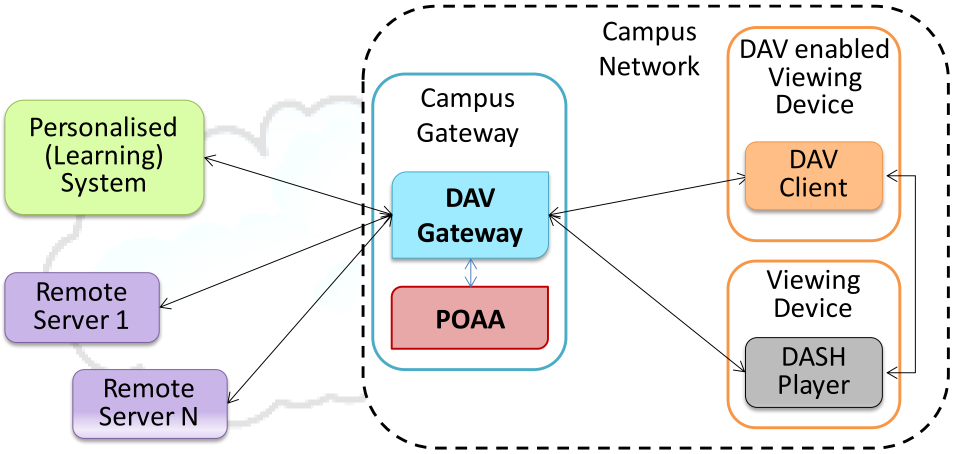

Figure 1-2: DPEA Architecture Block Diagram

A DASH [12], [13] based Performance Enhancement Architecture (DPEA) is proposed to enhance the performance of existing systems for personalised distribution of learning content. DPEA consists of two components: (a) Performance Oriented Adaptation Agent (POAA), used to provide the information necessary for selection of the best performing remote host and (b) DASH-based performance oriented Adaptive Video distribution solution (DAV), used to improve the content delivery process by utilising locally available content. This solution improves the content delivery process thereby increasing the overall viewing and hence learning experience. A DPEA block diagram is provided in Figure 1-2.

1.3.1 Performance Oriented Adaptation Agent

Two versions of the Performance Oriented Adaptation Agent (POAA) are developed to address question 1.2.1 of the research problem in Section 1.2 and ultimately to enhance video delivery by considering historic performance of the links to remote servers hosting requested video content.

The Open corpus Performance Oriented Adaptation Agent (oPOAA) [14]–[16] enhances the learning content selection process in adaptive open corpus PL (oPL) systems (and could be also used to enhance recommender systems set in an educational context). oPOAA considers underlying network conditions to perform adaptation of learning content through content selection, when several versions of the same or similar learning object (LO)/educational content are available at remote servers. This solution deals with different media types such as text, images, sound and video. oPOAA considers the statistics relating to the network connection to the server which hosts the content to identify the appropriate hosting server (from those available) that will achieve uninterrupted content delivery.

DASH-based POAA (dPOAA) [17], [18], focuses on video content only. This type of content selection may also involve sending different quality versions (differing bitrates) of LO (video) to the same learner depending on their geographical location (e.g. access to wired vs. wireless networks) or end user device used (e.g. laptop vs. smartphone). Located at the campus gateway, dPOAA evaluates remote servers based on the observed quality of the connection links between the servers (e.g. learning content repositories) and the campus network. The video content is then requested from the best performing remote server.

When a large number of students (e.g. a whole class, all students with same/similar learning profile, etc.) simultaneously watch the same educational video content, this solution is further improved by utilising content available in the campus network, which is achieved by using the DAV Solution described in Section 1.3.2.

1.3.2 DASH-based Performance Oriented Adaptive Video Distribution Solution

The proposed solution, DASH-based Performance Oriented Adaptive Video Distribution Solution (DAV) [19] addresses question 1.2.2 of the research problem presented in Section 1.2 and ultimately enhances video delivery by using video content available within the campus network using Dynamic Adaptive Streaming over HTTP (DASH) [12]. DASH provides client-controlled video content delivery via consecutive downloads of short video segments of varying bitrates (where higher bitrate means higher quality). Based on the Hypertext Transfer Protocol (HTTP) [20], DASH delivers content over the Transmission Control Protocol (TCP) [21] where variations in throughput are overcome by requesting the video segment at the bitrate that best matches delivery conditions. The popularity of DASH is growing [22] as it leverages existing HTTP based multimedia content delivery infrastructure and provides support for dynamic bitrate switching and live media services. Our adaptation process is conducted in two phases. First, DAV groups local nodes based on user profile information provided by an external personalisation system (e.g. a PL system) and on viewing device type. Second, it considers segments stored on nodes within a campus network, as well as data collected over time (e.g. remote host performance) to select the most suitable host (local or remote) for segment delivery.

1.4 Research Context and Scope

Students connect to the campus network to use PL systems. While there may be a number of networks available within a university campus (e.g. private providers, mobile networks, etc.) it is to be expected that students will choose the campus provided network on a number of grounds, including the following:

- Policy (e.g. the access to materials is often limited to devices within an educational network),
- Physical location, availability and performance (e.g. students have access to the university computer laboratories where they can use available computers and connect their terminals, such as laptops, to the wired network), and/or
- Economic (e.g. the access to educational networks is free of charge for registered students).

The creation (e.g. narratives, presentations and content production in general) and educational quality of the learning objects are not considered here.

1.5 Research Methodology

This section provides an outline of the methodological approach adopted. While some aspects of the design paradigm [23] were adopted in this research, the emphasis is placed on abstraction (modelling). This experimental scientific method consists of hypothesis establishment, model construction, experiment design and data collection, followed by results analysis.

Literature Review. The literature review sets the context and provides initial input for this research. It gives an outline of the related research in the areas of (a) Internet video streaming over TCP and emerging standards (with a focus on the MPEG-DASH standard) in Chapter 2; (b) Technology enhanced learning with a focus on open corpus PL systems including Adaptive Hypermedia systems, adaptation approaches in educational content provision in Chapter 3.

Model Construction. DPEA, comprising DAV and dPOAA components, and oPOAA are presented in Chapter 4.

Evaluation (Experiment design, data collection and analysis). To evaluate the proposed solutions, rather than developing a new evaluation platform, a simulated network environment is used. The tool of choice is Network Simulator (NS) version 2 [24] and version 3 [25], as NS is a well-established, open source simulation environment, widely used by the networking research community. New application modules were developed in C++ within the NS setting. Simulated sessions were implemented using these modules in a setting characterised by changing network conditions, e.g. the quality of connecting links, size of requested objects, number of concurrent users. A number of test cases were simulated to investigate the impact of different algorithms on the quality of delivered video. Results are presented in Chapter 5.

1.6 Research Contributions

This dissertation addresses several issues in the field of adaptive learning Web video delivery. A number of innovative solutions to the problem are presented and existing video streaming technologies are extended. Contributions are summarised below:

1. oPOAA

As video delays are known to frustrate computer users [26], the goal of the open Performance Oriented Adaptation Agent (oPOAA) is to select a server hosting requested learning objects so as to minimise initial delays (addressing question 1.2.1). The associated research contribution is:

- Design and evaluation of the **oPOAA** Algorithm based on a utility function.

2. dPOAA

The goal of the DASH-based Performance Oriented Adaptation Agent (dPOAA) is to evaluate hosting servers that store requested video content so as to minimise delays and stops in delivery (addressing question 1.2.1). The associated research contribution is:

- Design and evaluation of the **dPOAA** Algorithm based on a utility function.

3. DAV

The proliferation of educational video content and increased use of video viewing for learning has heightened demand for high quality service. In this context, the goal of the DASH-based Performance Oriented Adaptive Video Distribution Solution (DAV) is to deploy and innovatively exploit a current standard in dynamic adaptive streaming (MPEG-DASH) to better utilise locally available content leading to a better quality viewing experience. The associated research contribution addressing question 1.2.2 is:

- Design and evaluation of the **DAV** Algorithm based on a utility function located on the campus gateway.

4. Overview of MPEG-DASH Standard and Related Issues

Additionally, a literature review in the area of media streaming over HTTP and emerging standards with an analysis and comparison of different approaches to MPEG-DASH implementation was compiled to address question 1.2.2.

5. NS-3 DASH Modules

A number of NS-3 application modules were implemented to simulate video playback in the DASH research context. These modules are deployed to request, deliver and track video delivery. For example, these modules track, among other parameters: the number and duration of stalls due to rebuffering (stops in video playout to replenish the player's buffer); initial waiting times; requested bitrates during playout. These modules are described in Section 5.2.1.

Commercial and economic impacts of the proposed solutions were not the focus of this research. However, universities (educational institutions) operate under economic constraints, on the one hand they need to address the increasing demand for delivery of video content by providing faster network infrastructure, while on the other hand, they must consider return on their investment. This is even more important in recessionary times and in developing and post-crisis regions. DPEA provides enhanced user experience quality without further investment in campus network infrastructure.

1.7 Thesis Outline

This section outlines the structure of the remainder of this thesis. Chapter 2 examines different approaches to video content delivery over heterogeneous networks and related concepts and standards. Furthermore, this chapter defines Quality of Experience (QoE) and investigates how QoE is measured. Chapter 3 reviews the state-of-the-art in learning systems with an emphasis on PL systems with the aim of identifying systems that will benefit from solutions developed here. Chapter 4 presents our developed solutions in terms of architecture with related components, algorithms and deployment context. Chapter 5 presents the evaluation process of the proposed solutions. It provides test bed descriptions, evaluation scenarios, results and analysis. The final chapter, Chapter 6, is a summary of the main findings, conclusions and contributions. This chapter also identifies systems that could benefit from the solutions presented and provides directions and areas in which future research could be undertaken. Furthermore, limitations and overheads incurred by deployment of the proposed solutions are discussed. The technology context is further explored in Appendix A.

1.8 Related Publications

Research in the area resulted in the following publications:

- **Performance Oriented Adaptation Agent - oPOAA:**

    1. L. Rovcanin, C. H. Muntean, and G-M. Muntean, "Performance Aware Adaptation in Open Corpus E-Learning Systems," International Workshop on Technologies for Mobile and Wireless-based Adaptive e-Learning Environments, Adaptive Hypermedia and Adaptive Web-Based Systems (AH2008), 2008, p. 27 [16] *Best Paper Award.*
    2. L. Rovcanin, C. Muntean, and G-M. Muntean, "Performance Enhancement for Open Corpus Adaptive Hypermedia Systems," in *Adaptive Hypermedia and Adaptive Web-Based Systems*, vol. 4018, V. Wade, H. Ashman, and B. Smyth, Eds. Springer Berlin / Heidelberg, 2006, pp. 462–466 [15].
    3. L. Rovcanin, C. H. Muntean and G-M. Muntean, (2005): A Performance Oriented Adaptation Agent for Open Adaptive Hypermedia Systems, IT&T 2005 Cork, Ireland [14].

- **DASH-based Performance Oriented Adaptive Video Distribution - DAV:**

    4. L. Rovcanin and G-M. Muntean, "A DASH-based Performance-oriented Adaptive Video Distribution Solution," IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, London, UK, 2013 [19].

- **DASH-based Performance Oriented Adaptation Agent - dPOAA:**

    5. L. Rovcanin and G-M. Muntean, "A DASH-based Performance-oriented Adaptive Video Distribution Solution," IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, Beijing, China, 2014 [17].
    6. L. Rovcanin and G.-M. Muntean, "DASH: A Solution for Improving Video Delivery Quality in Heterogeneous Network Environments," in Convergence of Broadband, Broadcast, and Cellular Network Technology, G.-M. Muntean and R. Trestian, Eds. IGI Global, 2014 [18].

2 Technology Context

This chapter sets the technological context for the solutions presented in this thesis. Streaming high-quality video is becoming increasingly popular. However, due to the limited capabilities of viewing devices and the unreliable bandwidth and unpredictable nature of delivery networks, the overall viewing experience can deteriorate due to frequent periods of rebuffering.

Close to 85% of 800 European university students surveyed, regardless of their gender and origin, indicated that they would like to have online access to video recordings of their lectures [27]. The effects of the use of technology in learning are reported by a number of studies including annual longitudinal studies by the EDUCAUSE[1] Center for Analysis and Research (ECAR). ECAR surveys collated data relating to: technology (IT/mobile equipment, etc.) owned by students; the use of technology; the perceptions of how technology is affecting the learning experience. For example, their 2014 report [28] indicated the percentages of students using portable devices (laptops/smartphones/tablets) in class as 70%/59%/35% respectively. Longitudinal trends in undergraduate technology ownership presented in Figure 2-1 indicate a movement towards the adoption of portable devices for academic purposes.

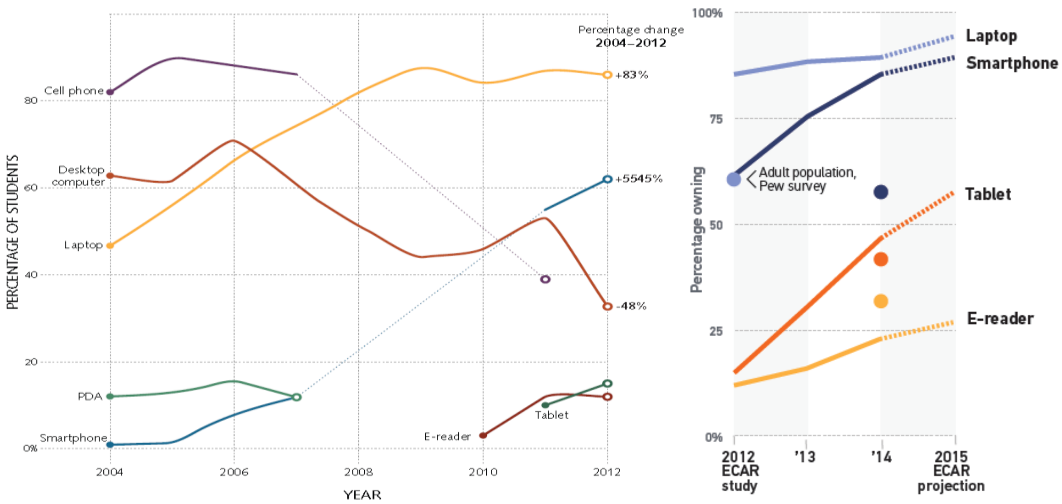

Figure 2-1: Undergraduate Technology Ownership, 2004–2012 [4] and 2012-2014 [28]

Among the many areas in which the online distribution of video content is expanding, education may be regarded as one of the most important. The cost of increasingly rapid production of educational video content (e.g. lecture recordings, student videos, etc.) is reducing and the resultant material is readily made available online. There are already many online offerings of free educational video content, including Coursera [3], Udacity [29] and edX [2]. At the same time, university classes are growing larger and more interactive, and students routinely use

[1] http://www.educause.edu [Accessed: 14-Dec-2015]

university-provided computers and personal portable devices (e.g. netbooks, tablets, etc.) to gain access to multimedia-based learning content.

The high data rates required for streaming video content and the large number of viewers (e.g. class, university or any company campus in general) impose significant demands on the delivery network. Inadequate capacity may result in long delays, high loss, frequent rebuffering events, etc. potentially affecting viewing experience. At the same time that the number of videos available for download or streaming is growing rapidly the demand for higher quality is increasing. Historically, many viewing devices (e.g. handheld devices) were incapable of receiving, storing and playing a given (large) amount of video content at the highest level of quality. But, some of today’s handheld devices have extended storage and processing capabilities.

The solutions proposed as part of this research are set in the context of adaptive online video delivery in a personalised context. Consequently, this chapter provides an overview of related technology factors, including video compression and methods for the evaluation of both video quality and user experience. This chapter also outlines the essential components of the delivery network and end user (viewing) devices. The proposed solutions utilise DASH-formatted video, so relevant DASH-related issues are addressed here.

2.1 Video

Digitisation is performed to capture video and enable subsequent transfer of captured video files. The human eye perceives continuous motion of separate images viewed rapidly in succession. This optical illusion is called the phi phenomenon [30]. The visual component (pictorial information) of the captured video is considered as a collection of still images (frames) that are displayed rapidly in sequence. These images are digitised (sampled) spatially and temporally using video recording devices at different bit depths (quantization). Raw digital video data is then represented as three separate component data streams for each colour (RGB – Red, Green and Blue). This representation can be easily translated to a luminance component (Y) and two colour differences for blue and red (Cb, Cr). The size of the resulting video file is determined by the number of pixels per line (horizontal resolution) and number of lines per frame (vertical resolution), the number of frames per second (frame rate) and the number of bits used to represent the colour of a single pixel.

Storing/transferring raw video is not practical due to the sheer volume of data, hence video/audio signals are compressed as described in following sections. Once compressed, video and audio streams together with metadata (subtitles, chapter-information, synchronisation information, etc.) are packaged into encapsulation containers, or wrapper formats, that contain all the information needed to present video. Streams may subsequently be encrypted for security and then distributed.

Typically, MPEG-like encoding algorithms are used in video coding. They have three types of frames: (a) I−frame, (intra-coded picture) is independently encoded and contains full specification (low compression); (b) P−frame (predictive coded picture) saves space as it contains the motion-compensated difference relative to the previously decoded frames; and (c) B−frame (bi-predictive coded picture) which potentially saves even more space as its content is the difference between the current frame and both the preceding and following frames. A Group of Pictures (GOP) is a group of successive frames within a coded video stream which specifies the order in which the I, P and B frames are arranged. It begins with an I-frame and the structure is described by M - the frame distance between two anchor frames (I or P) and N - the frame distance between two I-frames.

2.1.1 Outline of Video Compression Process

The quality (resolution, storage capacity, etc.) of video acquisition equipment is continuously increasing resulting in very large amounts of raw digitised video data, impractical for storage and network. Therefore, video compression is necessary to reduce the amount of data stored/transmitted while maintaining acceptable video quality. However, there is a trade-off between the resulting video quality, the cost of implementing the compression and decompression, and system requirements. Lossless compression preserves all of the original image information at the expense of a very low compression factor. In contrast, lossy compression techniques can attain high video compression factors but may omit detailed components of the original recording and even give rise to the presence of visible or distracting artefacts.

Compression ultimately aims to eliminate redundant elements in the source signal. Practically, compression performance is limited by algorithm efficiency. Typically, four types of redundancy present in a video signal may be exploited as follows [31]:

- ***Perceptual***: The human visual system is more sensitive to variations in luminance (brightness) than chrominance (colour difference); the well-known YUV (YCbCr) representation defines a colour space in terms of one luminance (Y, brightness information) and two chrominance (UV, colour differences, Cb – blue, Cr – red) components. This colour format is based on the visual perception characteristics of the human eye which relies primarily on brightness information (Y) to interpret image detail. Therefore, within the YUV system, colour (Cb, Cr) components may be represented at a lower resolution than luminance (Y) thus optimising the use of the available data space while preserving perceived visual quality;
- ***Spatial***: Within an image region, pixels are likely to have similar colour properties (intraframe correlation). Two-dimensional mathematical transforms (e.g. the discrete cosine transform - DCT) may be used to differentiate between lower (more important,

coded with higher accuracy) and higher spatial frequencies. Resultant coefficients may be appropriately quantized to reduce the transmission bitrate.

- ***Temporal***: Successive video frames tend to exhibit a high degree of similarity (interframe correlation). The bitrate requirement may be reduced by encoding the interframe difference rather than full frame information. While this approach is particularly effective for low motion videos, those with a high level of dynamic content (forward/backward/bidirectional) necessitate the inclusion of a provision for motion compensation where the currently processed block of pixels is compared with a reference block taken from (previous/future/both) frames to create an updated motion vector.
- ***Statistical***: some video coefficients may be observed to statistically recur more frequently than others. Statistical encoding techniques may be used to assign shorter codewords to more frequently occurring coefficients and longer codewords to less often used ones resulting in a reduction in the video bitrate requirement.

Most video codecs (software/hardware tools for compression and decompression of digital video) also use audio compression techniques in parallel to compress the separate, but combined data streams.

2.1.2 Video Compression Standards

Numerous proprietary and/or standardised algorithms are used to compress digital video signals. Standardised algorithms offer global and interworking capability and are proposed by organisations such as International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), International Telecommunication Union (ITU) and Motion Picture Expert Group (MPEG).

The Joint Photographic Experts Group (JPEG) [32], a joint effort of the ITU and ISO standardisation bodies, has produced a JPEG family of international standards for compression of colour and gray-scale still images. This popular standard offers variable compression ratios, where very high compression ratios result in “blockiness” of the compressed image. The JPEG standard uses the DCT transform and a quantization technique to eliminate redundant information. More complex, JPEG2000 [33] replaces the DCT transform with the Wavelet transform and consequently increases the compression ratio as compared to JPEG (”blockiness” is replaced with slight ”fuzziness” in the picture).

The Moving Picture Experts Group (MPEG) [34], a working group of ISO/IEC, has developed a set of popular standards for video compression and encapsulation. MPEG-1, the first standard developed by the group, is capable of compressing high motion video scenes, while maintaining a performance comparable with VHS quality at a bitrate of 1.5 Mbps. MPEG-1 was fully replaced by MPEG-2 which targeted compression of standard definition (SD) and high definition (HD) video signals at bitrates of up to 20 Mbps and high picture quality. The most important improvement brought by MPEG-2 was the compression of interlaced video. MPEG-3 was discontinued, as the same results (HDTV compression) could be achieved with minimum modifications of the MPEG-2 standards. Although constructed on similar principles, MPEG-4 and MPEG-H offer much higher flexibility and are gradually replacing MPEG-2.

The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) [35] has developed the H.26x video and audio compression standards. These standardisation efforts at times paralleled the MPEG activities resulting in a set of H.26x standards similar to and in some cases identical to the MPEG standards.

H.264/MPEG-4 Part 10 Advanced Video Coding (AVC) [36] format is a block-oriented motion compensation based standard developed by ITU-T Video Coding Experts Group (VCEG) and ISO/IEC MPEG to preserve image quality whilst allowing a high compression capability. It is an evolution of the existing ITU-T video coding standards (H.261/2/3) designed to provide "higher compression of moving pictures for various applications, stored on various storage media, transmitted and received over existing and future networks and distributed on existing and future broadcasting channels" [36, p. i].

High Efficiency Video Coding (HEVC) [37] is a (MPEG-H Part 2 ISO/IEC 23008-2)/(ITU-T Recommendation H.265) coding standard that significantly improves compression performance (e.g. 50% bitrate reduction) relative to existing standards - whilst maintaining the same perceptual video quality.

2.2 Network Delivery

Various wired and wireless network solutions have been proposed to address multimedia content delivery, and in many cases, multiple solutions are supported by the same device, e.g., most laptops are equipped with LAN and WLAN interfaces, smartphones typically connect to both WLANs and mobile networks. This section looks at different options for media content delivery over computer networks. They are categorised by corresponding ISO OSI layers, based on their functionality.

2.2.1 Application Layer Protocols

The ***Real Time Streaming Protocol*** (RTSP) [38] is a protocol used for control information exchange. It establishes and controls media sessions between end points in entertainment and communications systems. RTSP controls streaming media servers, while the actual media stream delivery is performed either by RTP in conjunction with RTCP or proprietary transport protocols e.g. Real Data Transport (RDT) by RealNetworks[2]. When video learning content is streamed over the ***Real-Time Transport Protocol*** (RTP) [39], RTP is responsible for framing, payload identification and sequencing. It adds timing data to the packets so that both jitter and packet loss can be monitored. The ***Real-Time Transport Control Protocol*** (RTCP) [40] is used

[2] http://www.realnetworks.com/ [Accessed: 14-Dec-2015]

to relay feedback information between client and server for RTP. RTP is independent of the transport protocol, however datagram transport protocols are used such as UDP, DCCP and SCTP. These transport protocols are outlined in Section 2.2.2.

The ***HyperText Transfer Protocol*** (HTTP) [20] is a generic, stateless application-level protocol that provides typing and negotiation of data representation, allowing systems to be built independently of the data being transferred. Although HTTP communication usually takes place over TCP/IP connections the standard (RFC 2616) "does not preclude HTTP from being implemented on top of any other protocol on the Internet or on other networks" [20, p. 13], as long as the protocol used guarantees a reliable transport.

2.2.2 Transport Layer Protocols

Once upper layer services are applied the media is passed to the lower transport layer for end-to-end data transmission. Typical protocol options at this layer include:

- ***User Datagram Protocol*** (UDP) [41] does not provide either reliability or congestion control features. It aims to meet the requirements of delay sensitive applications that generally tolerate or deal with loss, duplication or out-of-order delivery and rely on network-based mechanisms to minimise the potential for congestion collapse. Therefore, UDP is well suited to real-time multimedia streaming applications;
- ***Transport Control Protocol*** (TCP) [21] provides a reliable, connection-oriented, window-based congestion controlled byte-stream service aimed at applications requiring a high degree of reliability, but which are not overly sensitive to delays.

UDP has historically been favoured over TCP for the timely delivery of video packets, as no acknowledgement of delivery is required, but at the cost of reliability (no monitoring and retransmission in case of packet losses); TCP was avoided in video applications due to its throughput variations and excessive retransmission delays. However, over the past decade the choice of transport layer protocol for video (multimedia) delivery has shifted from UDP to TCP thanks to the popularity of HTTP-based streaming. The evident benefits of HTTP streaming when compared to UDP-based streaming protocols include exploiting of the existing Internet infrastructure, such as proxies, caches and Content Delivery Networks (CDN) and overcoming security obstacles such as firewalls and network address translation (NAT) [42] gateways. HTTP streaming however introduces larger overheads compared to RTP, mainly due to TCP overheads. Client player buffers can be used to deal with transient fluctuations of the transmission rate. While TCP was not designed for media streaming, it generally provides good streaming performance when the achievable TCP throughput is roughly twice the media bitrate with a startup delay of a few seconds [43].

Historically, TCP was designed and optimised for delivery of static files (e.g. FTP-like applications). TCP deploys several mechanisms to regulate the sending rate in response to network congestion. Congestion avoidance and timeout have significant impacts on the throughput between the sender and receiver. TCP starts a retransmission timer for every packet sent by the sender and waits for an acknowledgment from the receiver. The retransmission timer expires if an acknowledgment packet (ACK) for the corresponding packet does not arrive within a specified time period. This is remedied by the retransmission of the packet. The window size is then reduced and the retransmission timer value for this retransmitted packet is doubled. This behaviour, known as exponential backoff, continues until the retransmitted packet is successfully acknowledged. In congestion avoidance, the window size increases by one packet when all packets in the current window are acknowledged. More information about TCP may be found in [44].

Other transport layer protocols include: (a) a message-oriented **Datagram Congestion Control Protocol** (DCCP) [45] a hybrid solution which provides fair bandwidth sharing using session and congestion control (similar to TCP) without reliability or requiring message retransmission; (b) a reliable, message-oriented **Stream Control Transmission Protocol** (SCTP) [46] which, compared to TCP, provides multi-streaming (several independent streams of chunks are simultaneously transmitted bundling connections into a single SCTP association, allowing for independently sequenced delivery) and multi-homing (enabling transparent failover between redundant network paths for endpoints with multiple IP addresses); and (c) a connection-oriented **Multi-Path Transmission Control Protocol** (MPTCP) [47], an extension of TCP, which supports multiple sub-flows for a single connection session to increase network resource usage and redundancy.

2.2.3 Network Layer Protocols

Both UDP and TCP depend on the Internet Protocol (IP), a network layer protocol, for essential services such as addressing, routing and fragmentation if necessary. There are two IP versions: the original one - IPv4 [48] that is being gradually replaced by IPv6 [49] which among other improvements, offers a larger address space. This layer utilises lower layer protocols such as Ethernet or IEE 802.11 family.

2.2.4 Data Link Layer Protocols

The Ethernet (IEEE 802.3) family of protocols has remained the dominant enabling technology for local area networks (LANs) for the past four decades.

Wireless Local Area Networks WLAN (IEEE 802.11) are a very successful and cost-effective option for multimedia delivery. The popularity of WLANs is constantly increasing, and it may currently be considered to be the de-facto standard solution for university and other campus based wireless Internet access. However, due to the open nature of the transmission medium, WLAN performance is moderated by the range, unpredictability and vulnerability to interference of the wireless links themselves. QoS may be further severely degraded due to the inevitable congestion caused by increased number of learners simultaneously engaged in learning sessions.

While IEEE 802.11 standards focus on the physical and MAC layer, IEEE 802.11e and IEEE 802.11n provide QoS support features. Various prioritisation schemes [50] have been proposed to provide QoS improvement on the basis of differentiation between different traffic types.

2.3 Video Quality and User Experience

Internet video has recently become mainstream [51], and consequently it is important to investigate the impact of video quality on viewers and hence on learning processes and outcomes. This impact is recognised as an important issue by both academia (e.g. [51]–[55]) and industry (e.g. [56]). Relevant definitions are provided below:

- Quality of Service (QoS) refers to technical, objectively measurable network properties that influence the quality of content transport. Factors such as delay, packet loss, bitrate and jitter determine QoS and are described in Section 2.3.1.
- Quality of Experience (QoE) refers to the viewer's experience - the degree of delight, satisfaction or annoyance with the delivered content. QoE describes qualitative network performance and reflects the subjective perspective of the end user, which enables a more holistic understanding of the network quality as opposed to the more technology-oriented QoS perspective. It links objectively measurable network performance to subjective perception of network quality by the end users. ITU-T defines QoE as the "Overall ***acceptability*** of an application or service, as perceived subjectively by the end user" [57, p. 2]. A more precise definition is provided in [58, p. 5]: "the binary measure to locate the threshold of minimum acceptable quality that fulfills user quality expectations and needs for a certain application or system". More detailed discussion on QoE is provided in [59], [60]. QoE in the educational setting focuses on how technical settings/conditions affect learning experience. A model of QoE in eLearning [11] considers different learner roles: (a) learner (mainly affected by learning aspects), (b) user (mainly affected by usability and flow experience) and (c) customer (mainly affected by aspects of QoS). Research in the area indicates a clear link between QoE and QoS factors [61]. Effects of delay on QoE are presented in Section 2.3.2.
- Quality of Perception (QoP) [62], similar to QoE, considers enjoyment and satisfaction, however it is also concerned with the viewer's ability to analyse, synthesise and assimilate multimedia informational content. Quality of Experience in technology-enhanced learning is frequently linked to Quality of Service (QoS) or Quality of Perception (QoP) [62], [63]. A mapping of QoS parameters to QoE and QoP in the educational setting is presented in Section 2.3.2.

Therefore, apart from QoS parameters, user-related factors such as past experience, expectations, degree of fulfilment of user expectations, level of enjoyment, task at hand, etc. can be considered in video viewing evaluation.

2.3.1 QoS and Delivery Network Conditions

Best-effort IP networks are dynamic in nature (unreliable and unpredictable) and fluctuations in bandwidth and time-varying delays make it challenging for personalised learning systems to provide consistently good quality delivery of multimedia learning content over such networks. This issue is even more important in wireless settings, where losses and excessive delays can be caused by network congestion, noise disturbances and co-channel interference as well as by user mobility, multipath fading and weak radio conditions. From the network and transport technology point of view, several factors affect the streaming video quality, these include network throughput, packet delay, loss and jitter.

Throughput is “a measure of the rate at which data can be sent through the network, and is usually specified in bits per second (bps) .” [64, p. 198]. Generally, the higher the bandwidth, and consequently the throughput achieved by an application, the better the QoE experienced by the end user. Throughput fluctuations cause delays, which directly impact on QoE.

Packet Loss occurs when sent packets fail to reach their destination in time for playout. Congestive losses dominate in wired networks and occur when routers’ buffers overflow due to the data rates exceeding the available link capacity. Transmission losses are prominent in wireless networks and are caused by interference on the physical medium. Loss is a serious issue for multimedia transmissions as it may have a serious negative effect on perceptual quality. To avoid this, the packet loss ratio must be maintained below a certain threshold to achieve acceptable QoE. However, loss can also be counteracted with error control mechanisms (forward error correction (FEC), retransmission, error-resilience and error concealment). The packet loss will not be an issue if handled by transport layer (e.g. TCP).

Delay “of a network specifies how long it takes for a bit of data to travel across the network, from one computer to another; delay is measured in seconds or fractions of seconds.” [64, p. 197]. There are many sources of delay in any network in addition to those associated with propagation; delays are incurred by queuing and switching at each router along the path, while, in wireless systems, retransmissions introduce further delays. At the end-points, delays are incurred in capturing, encoding/decoding and de/packetising the data. Real-time multimedia, in which packets must maintain a strict order, is particularly sensitive to delay.

Jitter can be defined as variance in delay [64]. It is caused by network congestion, queuing delays, processing delays, signal drop, path changes or other reasons. While different buffering technologies can be implemented at the receiver end to collect arriving packets and forward them reordered to the decoder, little can be done when the buffer is full (arriving packets need to be discarded). When packets arrive at too slow a rate, the buffer makes no data available to the decoder which results in observable stalls in playout (rebuffering).

Download latency can be defined as the time that elapses from the user requesting learning content to the moment the user receives the requested page. A related video performance term is “join time” which can be defined as the delay between the time a player initiates a connection to a video server and the point at which the player video buffer has filled up sufficiently to allow playback to commence (i.e., moves to playing state) [51]. Open Adaptive Hypermedia Systems (see Section 3.4) are distributed by nature and their response times depend on the performance of the content hosting repository and the underlying network.

2.3.2 QoE/QoP and Delay/Latency

This section investigates how delay affects the perceived quality of Web content.

A number of surveys [52], [53], [65] indicate several significant adverse effects of long download waiting times on the Web. ITU-T [66] sets maximum waiting times, but fails to present empirical evidence of the effects on user perception when these targets are missed. Prolonged delays result in changes in user attitude [53], behaviour (e.g. a decision to abandon a Web page or an intention not to visit the site again) and perceptions regarding Web page quality and usability [55], where low quality of network access directly translates into user annoyance [67]. Even delays as short as four seconds decrease performance and may change behavioural intentions [68]. While such studies are not recent they retain their relevance. Web users continue to discover new applications that consume Internet resources [69] while their expectations continue to rise [67] and exceed the responsive capability of the infrastructure. With the recent shift to TCP-based media streaming, there is increasing interest in the effect of waiting times [67]; a number of recent studies (e.g. [70] for VoIP and data services) attempt to identify psychophysical relationships between the waiting time, network bandwidth and user perception.

In terms of TCP-based video streaming, a straightforward increase in the video player buffer size to alleviate the rebuffering issues may be counterproductive as it may result in an increased join time, which, may reduce the likelihood of a viewer visiting the site again [51]. The most significant factors which influence QoE are the frequency and duration of noticeable rebuffering events. ”Initial buffering is more tolerated by mobile customers. It is better to have a single rebuffering than repeated events if interruption is unavoidable. “ [71]. Users who are not merely sampling videos, but are actually interested in the content are more tolerant of longer join times (and buffering) [51], however, the tolerance drops at a certain point (around 15 seconds for join times). The impact of video quality on user engagement was investigated in [51] where it was found that viewing time decreased between 1 and 3 minutes for every 1% increase in the buffering time. An example of a commercial provider measuring the impact of page load times on user satisfaction is provided in [56].

One of the seminal works [54] proposing metrics for user-perceived quality recommends adaptation of network-level parameters, such as delay and jitter, to ensure the satisfactory transfer of information. A coarse mapping proposed in [72] links network QoS parameters (bit error rate, delay, jitter, segment order and segment loss) with QoP. Here video is most affected by changes in segment order. The video QoP is moderately affected by delay and jitter, while bit error rate and segment loss had little effect on the reported QoP levels. It should be noted that in case of frame loss, received frames were replicated resulting in the prolonged display of identical visual information.

A number of studies have investigated the effect of multimedia quality on learning and QoP. An empirical study [73], conducted with 132 participants to determine the effect of cognitive styles on users' subjective perceptions of multimedia quality, concluded that the technical quality (frame rate and colour depth) of educational multimedia clips did not impact the viewer's educational experience. However, the study did not investigate the effect of stalls, frequently experienced in TCP-based streaming in bandwidth-constrained environments.

Overall learning experience depends on a plethora of parameters, however download latency has proved to be the key factor that directly affects user productivity, perception and satisfaction. Furthermore, download latency has direct implications for user retention; Web users readily abandon pages which fail to download within tolerable waiting times and similarly discard videos with prolonged join times [51]. Any download delay that learners "experience while using an online instructional tool may have detrimental effects on performance and satisfaction" [74, p. 250]. Issues relating to instructional Web page (text and graphic objects) delays were investigated in [75]. This comparison-based study (original AHA! vs. QoEAHA [63], [76]) demonstrates that the end-user perceived quality of online interaction with a personalised learning system is, among other factors, affected by network-related and user device-related factors. While QoE model deployment had no effect on learning outcomes, significant learning performance improvements in terms of reduced: (a) study session time, (b) information processing time per page and (c) number of revisits to a page were reported. Furthermore, the perceived end-user QoE was increased in case of QoEAHA. In distance education applications, a video conference delay of approximately 3 to 5 seconds is often distracting to both presenters and students [77]. Recent trends in the use of limited capability hand-held devices have introduced yet another source of delay to the Web user - Web system interaction.

However, carefully paced delays can benefit the learning process. For example, the findings presented in [78], [79] indicate that longer delays might improve performance on more cognitively demanding tasks as they allow for more "thinking time." An empirical study described in [80] reports that a 10 seconds delay between tutorial questions only slightly increased session time and that subjects preferred consistent delay to zero delay or variable delays. The increased performance under consistent delay was attributed to students using delay periods to study. A subsequent study in [81] confirmed the initial findings and reported a 7% increase in productivity. These improvements in speed, accuracy and maintenance of learning outcomes were attributed to an externally imposed pace of learning. Consideration of the impact of the deliberate introduction of positioned delays by means of time fillers or design options (e.g. [82]) is outside the intended scope of this thesis. Rather our focus is on removing delays and interruptions of playback due to poor network throughput. Such occurrences are unpredictable by nature and result in the display of random "frozen" content that does not contribute to the learning process.

Further, somewhat related topics outside the present scope of this research include the effects of delays in educational/instructional feedback [83], including delays in the communication of results [84] and in reinforcement.

In conclusion, poor network conditions result in long initial delays and frequent interruptions for buffering in TCP-based streaming. While planned and consistent delays in educational media content can be used for learning, delays introduced by poor network conditions are unpredictable and result in the display of random content. Video consumers' experience depends on the context, and while learners accept degradation of quality in terms of colour depth and frame rate, they are generally annoyed by stops and interruptions of playback (see Sections 2.4.7 and 2.4.13). Such annoyance negatively impacts learning outcomes.

2.3.3 QoS Standards Relating to Delay/Latency

ITU-T Recommendation G.1010 [66] introduces thresholds on delay, delay variation and information loss in the context of different applications. Furthermore, an associated model for multimedia QoS categories based on user expectations for a wide range of multimedia applications is indicated in Figure 2-2.

ITU-T Recommendation Y.1541 [85] identifies six QoS classes based on end-to-end performance parameter values for packet transfer over IP-based networks (packet transfer delay, packet delay variations, packet loss ratio and packet error ratio).

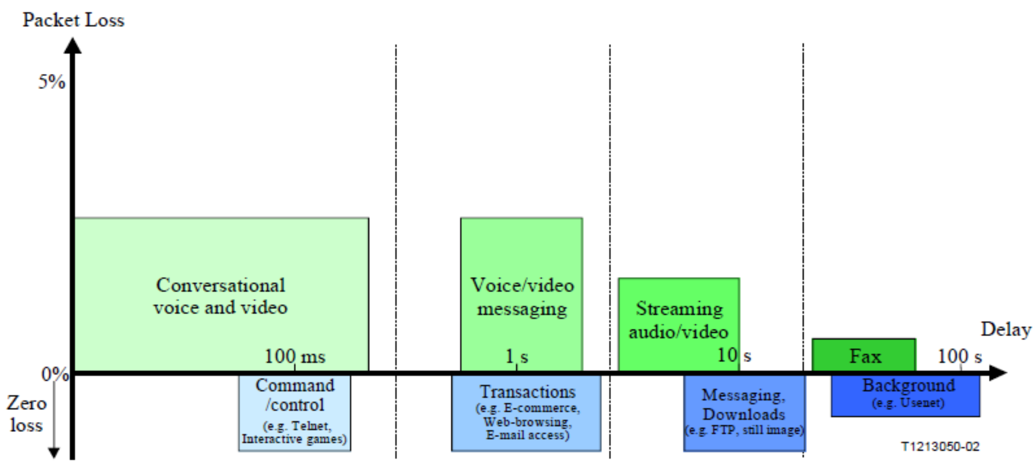

Figure 2-2: Mapping of User-centric QoS Requirements [66]

3GPP specifications [86] provide a classification of services into four QoS classes depending on the degree of sensitivity to delay of application traffic: (a) Conversational class - highly delay sensitive conversational streaming (e.g. Voice over IP, and videoconferencing); (b) Streaming class – sensitive to delay variations real-time streaming (e.g. real-time video/audio), which is the focus of this work; (c) Interactive class – low bit error rate request-response interactions (e.g.

Web browsing, data retrieval); (d) Background class - sent and received data with no preset time expectation (low bit error rate and no specific requirements on delay).

2.3.4 Methods for Video Quality Measurements

Video quality degradation has four main sources in a typical end to end system: (a) Encoding with lossy compression algorithms (may introduce spatial distortion, such as blockiness and blur), (b) Transmission through a network (may introduce both temporal and spatial distortion, such as stalls, jerkiness, missing frames, etc.), (c) Decoding inaccuracy/error by the end user player and (d) physical limitations (e.g. screen size and resolution) of the rendering devices. While the impact of compression (encoding/decoding) in relation to video quality is undeniable, the focus in this research is on the impact of transmission errors on perceived video quality.

It may be claimed that true video quality is a combination of human perceived quality and objective (technical) quality. Human perception of video quality is to some degree subjective as it depends on the relationship between sensory (e.g. aural/visual) channel processing and higher level processing that includes experience, emotions, knowledge, expectations and context, tactile, olfaction, and gustatory senses, etc. While QoS parameters can be objectively measured, both objective and subjective methodologies are needed to determine a useful performance indicator for perceived video quality. A number of previously proposed video quality measurement methodologies are outlined in this section.

2.3.4.1 Methods for Objective Estimation of Video Quality

Objective quality tests and metrics are used to automatically predict human perceptual "experience" in evaluating image/video quality. These methods typically involve algorithms and formulas that "measure the quality in an automatic, quantitative, and repeatable way, based on either signal processing algorithms or network-level quantitative measurements" [87]. The industry widely adopts objective estimation of video quality as subjective tests tend to be resource intensive [88]. Research proposed solutions are typically tested in a simulated setting that implies a degree of simplification when compared to a real-world setting. While simulations capture only the principal aspects of the system, they focus on the aspects that are significant for the context and algorithm evaluated.

In this research, measurements obtained in a simulated environment (as described in Section 5.2.1) are used to objectively measure video quality. The focus is on DASH-formatted video delivery where TCP is used at the transport layer, so there will be no packet loss. In this case, evaluation of the delivered video is based on metrics more suited to the nature of the HTTP-based video delivery described in Section 2.4.13 where the initial buffering and rebuffering periods have a significant impact on the extent of the quality degradation perceived by the user. This is a well researched area and comprehensive surveys may be found in [89], [90].

For completeness sake, an overview of objective estimation methods is provided in Section A.1 of Appendix A.

2.3.4.2 Methods for Subjective Estimation of Video Quality

Subjective assessment methods evaluate the quality of a video using measurements based on the reactions of real viewers (end users, learners, etc.). This approach is considered as the most appropriate way of predictively determining the reactions of those who might view the tested video. Tests with human subjects tend to be very expensive (e.g. test equipment and room setting, test subject expenses, time consumption, etc.), and the tests need to involve large number of subjects for statistically relevant results.

Mean Opinion Score (MOS) [91] is one of the most popular metrics used for both quantitative and subjective quality evaluation. It provides a numerical indication of user satisfaction with received media after compression and/or transmission. MOS is generated by averaging (arithmetic mean) the results (individual scores) of a set of standard, subjective tests performed across a number of viewers. MOS values range between 1 (lowest perceived quality) and 5 (highest perceived quality) as given in Table 2-1.

MOS	Quality	PSNR(db)	Impairment
5	Excellent	>37	Imperceptible
4	Good	31-37	Perceptible but not annoying
3	Fair	25-31	Slightly annoying
2	Poor	20-25	Annoying
1	Bad	<20	Very annoying

Table 2-1: PSNR to MOS Mapping and Impairment Scale

ITU-R BT.500-13 [92] formalises subjective quality evaluation and recommends experimental conditions including viewing distance and conditions (room lighting, display features, etc.), test subjects and material selection, data analysis methods, etc. Objective Score scales used for video and audio quality assessment can be used for Web QoE [88]. ITU-T RP.910 [93] also provides recommendations for assessing the visual quality of multimedia applications. Viewers are expected to judge video sequences independently by providing a rating on a category scale, such as the one given in Table 2-1. Web-based crowdsourcing and access to a large pool of (self-selected) subjects can be used as a cost effective alternative to laboratory-based studies. A survey of such frameworks may be found in [94].

2.4 MPEG Dynamic Adaptive Streaming over HTTP (DASH)

Approaches to video delivery over the Internet have evolved from datagram-based to adaptive bitrate-based streaming over HTTP. MPEG-DASH is one such example. As the proposed solutions utilise DASH-formatted video, this section addresses relevant DASH-related issues.

2.4.1 Overview of Web Video Delivery Approaches

Video streaming is a topic which attracts a high level of interest in the field of multimedia communication [95]. The result has been new protocols specifically designed to provide a video streaming service over the Internet. Historically, video was streamed over a best-effort network using a datagram protocol with packet-level control. Video streaming applications require real-time and consistent transmission throughput that is provided with efficient flow and rate control mechanisms. RTP is one such protocol providing full control over packet transmission and it is widely used in combination with RTCP over UDP. However, RTP suffers from a number of shortfalls, (a) payload format is video compression format specific (so there are problems with the support of the new/future media compression formats), (b) out-of-band signalling is required, for which RTSP is required, (c) implementation is complex as flow and congestion control, packet loss and out-of-order delivery must be handled at packet level, (d) Firewall and NAT routers have high failure rates with datagram transport protocols, a problem which severely afflicts the deployment of UDP-based streaming solutions (e) specialised infrastructure for caching and load balancing is required.

For the above reasons, protocols such as HTTP although not designed with real-time media delivery in mind, are being adapted for streaming due to their general popularity. The majority of the deployed adaptive multimedia streaming solutions are based on HTTP [20], which easily traverses firewalls and NAT devices, and makes full use of existing Web infrastructure. **Progressive download** [96] is an example of a HTTP-based approach to video delivery as used by most Flash-based sites. The user simply downloads a media stream as a file and it allows playing of incompletely downloaded videos using simple players or HTML5 enabled browsers. As the send rate is not limited, a large buffer is required on the client side. Here, the entire video is stored as a single file and servers provide multiple versions of these files, thus meeting requirements of heterogeneous viewing devices. However, users are expected to select the "right" video version which could lead to incorrect choices [97]. In general the use of progressive download reduces the initial delay (time between the start of the video download and video play out), however the approach is somewhat inefficient in terms of resource utilisation, since if the viewer abandons the viewing, portions of unwanted video are buffered unnecessarily. Furthermore, there is no mechanism to permit dynamic changes in video quality (as the video is played from one file) when delivery network conditions change (e.g. playout interruptions are common occurrence) often with consequent negative effects on the viewing experience.

Importantly, video consumers are particularly sensitive to interruptions for rebuffering [51], [71] and quality expectations for streamed video continue to rise. It was recognised that improved approaches to video delivery over HTTP were required while retaining the redeeming features of existing approaches to video delivery. In this context, Dynamic Adaptive Streaming over HTTP (DASH) [98] [12] was developed.

2.4.2 MPEG-DASH Overview

MPEG-DASH is a relatively recent standard (ratified in December 2011 [12], tested in 2012, edited in 2014 [13]) that has being proposed by ISO/IEC MPEG and the 3rd Generation

Partnership Project (3GPP) to address the problems of interoperability and traditional approaches to web streaming as well as to improve Quality of Experience (QoE) levels. Vendor specific HTTP-based adaptive streaming solutions have been available since 2007. Move Networks, Inc. was the first to adopt HTTP-based streaming and other vendors followed. Commercial (vendor-specific) implementations include (a) Adobe's Dynamic HTTP Streaming (HDS) [99] which is platform agnostic and supported by the Adobe Flash Player, (b) Apple's HTTP Live Streaming (HLS) [100] based on Apple's iOS and Google's Android operating systems, and supported by Apple's Quicktime media player, and (c) Microsoft Smooth Streaming [101] based on the Microsoft's Windows operating system and supported by Microsoft's Silverlight application framework. Each implementation provides adaptive bitrate streaming and uses the MPEG-4 H.264/AVC coded video as input.

MPEG-DASH is a standard for a client controlled media delivery model. Media content is typically stored on standard HTTP servers in multiple versions, further divided into segments of varying duration. The logic of a typical DASH-based adaptive system is located at the client side, which scales well. As a client/server paradigm, it uses existing HTTP-based multimedia content delivery infrastructure, such as web servers, HTTP caches and CDNs without the need for specialised servers such as the Flash Media Server (or other competing products). MPEG-DASH is HTTP/TCP based which eliminates the firewall and NAT gateway traversal issues that plague UDP-based approaches. Unlike progressive download, MPEG-DASH supports dynamic bitrate switching and live media services.

In a MPEG-DASH context, web servers host multiple presentations (versions/copies) of video content differing in temporal, spatial or fidelity quality (e.g. frame rate, resolution, colour depth, level of detail) ranging from lower quality renditions for 3G connections, up to very high quality (AVC/HEVC HD). Each representation consists of segments (i.e. fragments, media chunks) of predefined duration, e.g. 10 seconds. MPEG-DASH performs video streaming using consecutive downloads of these video segments. The process is initiated by the client, and the server responds with a video manifest (description) file. The client then proceeds by requesting content quality that matches initial conditions (e.g. connection type, buffer size, remaining battery life) without the need for negotiation with the hosting server. After a segment is received, the client simply requests (via the HTTP GET method) the next segment of the quality that matches changes of the device state (e.g. buffer fill level, battery life), network traffic (e.g. drop/increase in estimated throughput) or user preferences (e.g. viewer profile, current task) [98]. This process is illustrated in Figure 2-3.

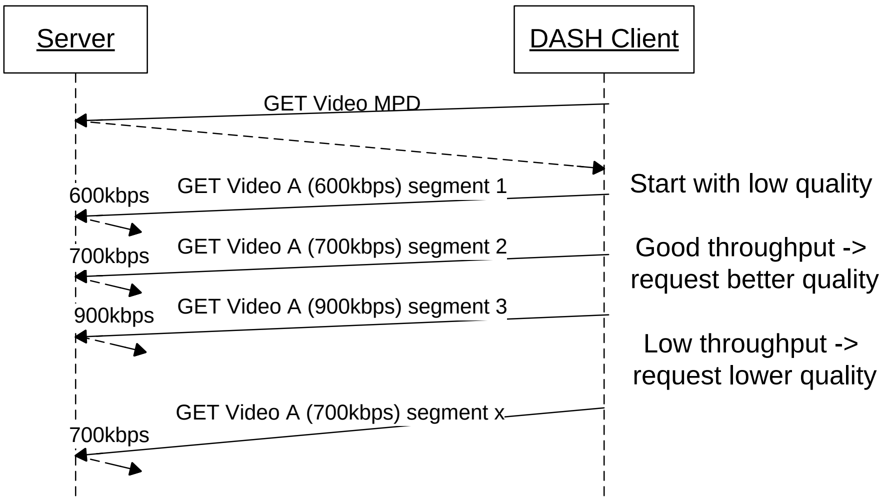

Figure 2-3: MPEG-DASH Video Requesting Process

The MPEG-DASH model places decision-making at the client side. The client's insight into performance yields the most informed adaptation decision on what quality to request from the server, which leads to optimum QoE levels under given delivery conditions. This part of the standard "does not provide a normative specification for such a client" [12, p. 7], however it provides an "informative client model" [12, p. 7] which is utilised in Section 2.4.5 to describe client-side architecture and behaviour.

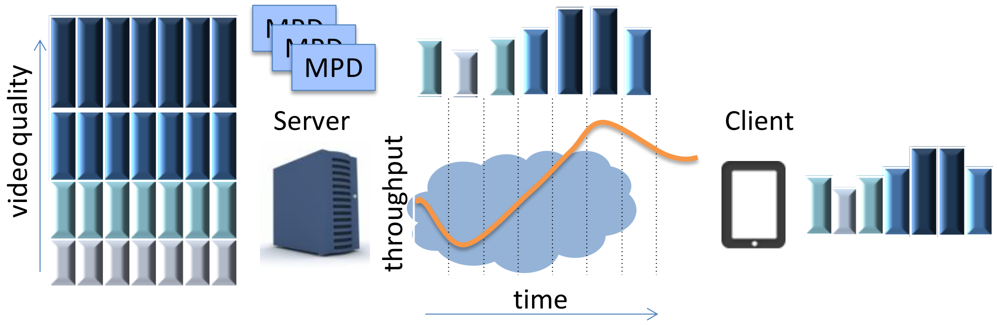

Figure 2-4: MPEG-DASH Adaptation Overview

Figure 2-4 illustrates an example of the segment selection process where the server stores a video file in four qualities (Low, Medium, High and Highest), the video is divided in seven segments and is streamed over a network of variable bandwidth. The quality (bitrate) of segments requested by the client with a portable device (e.g. tablet), depends on the current network bandwidth. When bandwidth is very low, the client requests the lowest available quality (e.g. the second segment), as the bandwidth improves, the quality of requested segments also improves (e.g. third segment), finally, when the bandwidth improves further, the better quality is requested (e.g. segments four and five). Clearly, since a client's requests must take into account network bandwidth, a client requires a bandwidth estimator. Approaches to client-side bandwidth estimation are described in Section 2.4.8.

This approach is also cost effective as there is no need to pay for specialised video streaming servers. The carrier's network delivers just the video segments that are needed (as opposed to progressive download, where a long initial buffering is required prior to the playout).

MPEG-DASH is gaining popularity and the main industry players are collaborating in building compatible clients and content creation tools. These industry groups include: Open IPTV Forum (www.oipf.tv/), HbbTV (www.hbbtv.org/), UK Digital TV Group (www.dtg.org.uk/) and the DASH Industry Forum (www.dashif.org/). An example of mainstream DASH applications include BBC pilot using HTML 5 compatible browsers [102].

In summary, the idea behind MPEG-DASH is to harness the available, low cost HTTP infrastructure to meet expanding demands for streamed video. The web servers provide multiple versions of a video, thus meeting the requirements of heterogeneous viewing devices, making MPEG-DASH a practical solution for addressing video streaming demands due to the surge in availability of fast mobile Internet connections and the ubiquitous utilisation of portable devices.

2.4.3 MPEG-DASH Data Model Overview

An MPEG-DASH **Media Presentation** is a collection of encoded (and deliverable) versions of media content (and the appropriate description of these). **Media content** is composed of a single or multiple contiguous media content periods in time. Each **media content period** is in turn composed of one or multiple **media content components** (e.g. audio components in various languages and a video component). Each **media content component** is one continuous component of the media content with an assigned media component type (e.g. audio or video) and may have several encoded versions (i.e. **media streams**). Each **media stream** inherits the properties of the media content, the media content period and the media content component from which it was encoded and in addition is assigned the properties of the encoding process such as sub-sampling, coding parameters, encoding bitrate, etc. These describing metadata are relevant for static and dynamic selection of media content components and media streams.

2.4.4 MPEG-DASH Media Presentation Description (MPD)

Videos are described in MPEG-DASH Media Presentation Description (MPD) files. An MPD is an eXtensible Markup Language (XML) formatted manifest file that describes media presentations and provides references to media streams. MPDs contain sufficient information for a client to implement a streaming service and may contain information about "program timing, media-content availability, media types, resolutions, minimum and maximum bandwidths, and the existence of various encoded alternatives of multimedia components, accessibility features and required digital rights management (DRM), media-component locations on the network, and other content characteristics" [103, p. 64] including video segment timing, Uniform Resource Locator (URL), media characteristics such as video resolution and bitrates. The MPD is analogous to an HLS [100] m3u8 file, a Smooth Streaming Manifest file [101] or an f4m file in HDS [99]. MPDs are created by the content provider and are typically stored at the HTTP server hosting its associated segments. The standard assumes that the client has access to the MPD. The structure of an MPD file is defined by MPD Schema given in Annex B of the standard [12].

The structure of an MPD file is illustrated in Figure 2-5, where the Media Presentation is a sequence of one or more Periods (temporal sections) containing one or more Adaptation Sets. Adaptation Sets of a particular Period may be assigned to a group indicated by a group attribute in the MPD. Adaptation Sets in the same group are generally considered alternatives to each other. Representations (content alternatives) are grouped into Adaptation Sets and consist of media segments of predefined duration (e.g. 6 seconds). At most one Representation within an Adaptation Set is selected to compose the delivered presentation. The client processes video per period, requesting metadata for the period and, consequently, relevant segment(s) within that period. A consistent set of encoded versions of the Period media content is available (i.e. the set of available bitrates, languages, captions, subtitles etc.) and does not change during a Period.

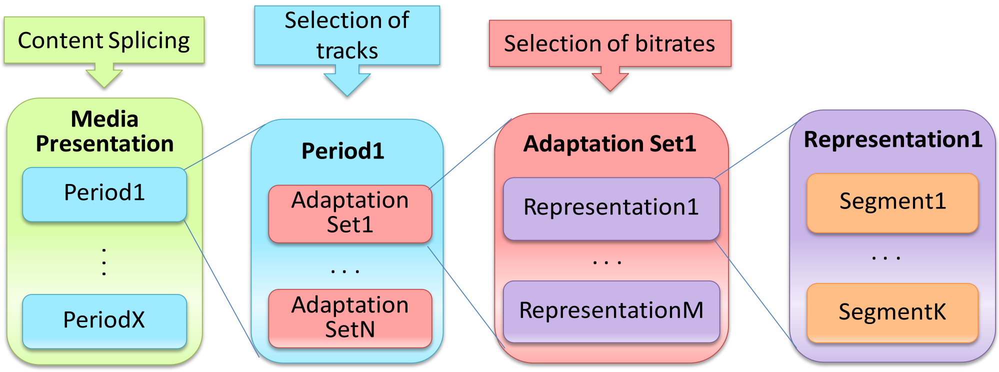

Figure 2-5: MPEG-DASH MPD Structure and Associated Processes

MPDs also contain redundant information and metadata relating to Media Streams for the purpose of selecting or rejecting Adaptation Sets or Representations. The associated metadata includes: role, coding format, DRM, language, resolution, bandwidth, etc. The MPD may be grouped in four levels: Video content - MPD level, Period level, Representation level and video mapping levels.

Presentation rendering starts at a Stream Access Point (SAP) - the position in a representation enabling playback using only the information contained in the representation data from the SAP onwards. A client may switch (change) media representation based on an updated MPD or changes in its delivery environment. The switch occurs at a SAP. The URL(s) and, optionally, byte range(s) are provided for each accessible Segment.

2.4.5 MPEG-DASH Client Side Architecture and Behaviour

The logical components of a conceptual DASH client model are depicted in Figure 2-6. The DASH Access Engine first requests and receives the MPD file, then constructs and issues requests (HTTP GET) and receives Segments (or parts of Segments). The output of the DASH Access Engine consists of media in MPEG container formats such as ISO BMFF and MPEG-2

TS. The timing information maps the internal timing of the media to the timeline of the Media Presentation [12]. The actual media playback is controlled by the Media Engine operating on the media streams contained in the Representations. It follows that the Media Engine is not controlled by the MPD and does not require any information in the MPD for successful decoding and presentation of the contained media streams. The Media Engine processes the Initialization Segment enabling it to start decoding the payload of any media stream within a Segment.

The DASH access engine (Figure 2-6) processes the Index Segment (providing timing and stream access information) in order to access Subsegments by the use of HTTP partial GET requests. This index may be downloaded in advance.

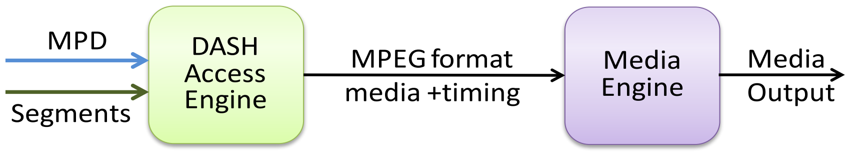

Figure 2-6: MPEG-DASH Client Model

In MPEG-DASH, the control of media delivery lies exclusively with the client, but the standard [12] does not provide normative procedures on DASH client implementations. However, Annex A of the standard provides "informative" description of client behaviour. The Media Engine can be a vendor specific or a plug-in module that can process MPEG-formatted media. Thus, playback is controlled by the Media Engine operating on the media streams in the usual way [12]. The standard provides an example of client behaviour necessary for a continuous streaming experience. This behaviour is outlined in the steps below:

1. The MPD is parsed to select a set of Adaptation Sets suitable for the client's environment considering the values for AdaptationSet elements, the AdaptationSet@group attribute and any constraints in Subset element if provided.
2. A Representation from each Adaptation Set is selected (based on the value of the @bandwidth attribute and client decoding and rendering capabilities).
3. A list of accessible Segments for each Representation is created. The segments are accessible if they are available for the actual client-local time measured in wall-clock time (and other timing restrictions when dynamic MPDs are used). The Segment list contains timing/location information for all types of segments.
4. The media content is accessed via *requests* for (entire or byte ranges of) Segments as given in the Segment list (step three).
5. The requested media is buffered (at least for the value of @minBufferTime attribute duration) before starting the presentation.
6. The rendering starts when (1) a Stream Access Point (SAP) is identified for each of the media streams in the different Representations, (2) the timing is right and the observed

throughput is greater than or equal the sum of the @bandwidth attributes of the selected Representation (if not, longer buffering may be required).

7. The presentation continues with continuous requests for (parts of) Media Segments.

The Client may switch Representations when the environment changes (e.g. a change in observed throughput) or the MPD is updated. The switch to a different Representation takes place at a SAP (typically at any segment boundary), where different Representations may be time-aligned to aid seamless switching. The switching points are announced in the MPD or/and the Segment Index. Over time, the list of available Segments can be expanded for dynamic MPDs.

MPEG-DASH supports live media streaming using dynamic MPDs. In the case of dynamic MPDs, an updated MPD is fetched if MPD@minimumUpdatePeriod is present and the current playback time lies within a threshold defined in the current MPD for the Representation. The client processes the fetched MPD and updates accessible Segment list (e.g. add newly available segments) for each Representation if required.

The Client should handle HTTP redirections and respond appropriately to various HTTP client and/or server errors (e.g. when a Client receives a HTTP error for the request of Initializaiton/Media Segment). Repeated HTTP server errors for the Client's requests may involve terminating the streaming service or, when multiple BaseURL elements are available, the client may also check for alternative instances of the same content hosted on a different server.

2.4.6 Player Buffer Considerations

Media players store prefetched media data in buffers to aid processing and allow for error correction in order to absorb short-term fluctuations in the TCP throughput. For example, when the connection throughput drops below the bitrate of the currently requested segment, the quality level can be maintained by consuming the buffered content.

Client buffers store data, but are frequently defined in terms of storage time (e.g. 10 seconds of buffered data), as audio and video are temporal media. Buffer content is constantly changing during media playout - new data is continuously added and processed data removed. Client-side buffering loads pre-fetched data into the client buffer by introducing a startup delay. This initial delay period may be adjusted in response to network conditions and the bitrate of the data stream.

An experimental study [104] consisting of 1000 minutes of video streamed over LANs and WANs suggests a buffer of size 5 seconds when no bandwidth estimation is possible. The same study suggests 5 second buffers for high bandwidth and 3 second buffers for low bandwidth when bandwidth estimation is possible. Early versions of Windows Media Player [105] used a default buffer length of 5 seconds.

An experimental evaluation [106] of three commercial players - Microsoft Smooth Streaming, Adobe HTTP Dynamic Streaming and Netflix [107], with persistent and short term bandwidth variations, found that playback buffer size in Smooth Streaming decreased when the available bandwidth was less than the requested bitrate and increased when the available bandwidth increased. Netflix employs a large playback buffer (up to few minutes) and sometimes changes to bitrates higher than the available bandwidth as long as the playback buffer remains almost full. A study [108] reported on the impact of changing HTTP adaptive streaming (using Apple Live Streaming, Adobe Dynamic Streaming and Microsoft Smooth Streaming clients) rates on user QoE. It was found that clients were required to maintain a video buffer of 15-60 seconds in order to ensure seamless transitions during changing network conditions and the scaling up or down of the video quality level.

2.4.7 Startup Delay and Initial Buffering Considerations

Initial buffering is the minimum amount of pre-buffered media content (measured in seconds) that is required to commence video playback. Excessive startup delays give rise to user annoyance [67] and lead to a drop in user experience regardless of the received video quality [109].

The DASH standard [12] MPD element @minBufferTime attribute specifies a common duration (e.g. minBufferTime="PT1.2S"). The client buffers media for a period which is at least that given by the value of the @minBufferTime attribute before starting the presentation. This attribute is linked with @bandwidth at Representation level. If the Representation is continuously delivered at @bandwidth bitrate, when starting at any SAP, a client will have enough data for continuous playback "providing playout begins after @minBufferTime * @bandwidth bits have been received" [12, p. 31]. The standard considers startup delays for video seek tasks and the Initialization Segment. It suggests improving seek times by the use of partial HTTP GET requests to initially request the Segment Index from the beginning of the Media Segment. This Segment Index can be then used to map Segment timing to byte ranges of the Segment. "By continuously using partial HTTP GET requests, only the relevant parts of the Media Segment may be accessed for improved user experience and low startup delays" [12, p. 97]. Since, the Initialization Segment needs to be downloaded before any Media Segment can be processed, startup time may be reduced significantly by keeping the Initialization Segment small [12].

The video streaming solution proposed in [110] considered 15 seconds as the maximum startup delay. The algorithm proposed in [111] aims to reduce the initial delay by requesting the lowest quality for the first segment downloaded. The fact that the quality of the first few seconds of the requested video will be of lowest quality is mitigated by an aggressive "fast start" phase where for each subsequent segment the next higher quality level (bitrate) is requested as long as the measured throughput is sufficiently higher than the requested bitrate and the buffer level is sufficient. Higher bitrates are monotonically increasingly requested when the buffer fill level is higher. Whilst the initial delay is not explicitly discussed, the solution proposed in [112] assumes that the first segment is usually downloaded by simply requesting the lowest bitrate alternative. The streaming solution proposed in [112] uses initial buffering of two segment durations. A detailed analysis on the use of DASH for live service conducted in [113] recommends that the initial buffering should be about twice as long as the segment duration.

2.4.8 Network Performance Estimation for HTTP Adaptive Streaming

HTTP adaptive streaming clients determine the quality of the next requested segment based on an estimation of the current network bandwidth or other QoS factors. A selection of bandwidth estimation algorithms deployed by DASH-based clients is presented in this section.

Traditionally, network performance measurements are obtained by service providers in order to verify that Service Level Agreement (SLA) performance targets are being met within acceptably high levels of probability. Such measurement data are collected either passively within the network (e.g. Simple Network Management Protocol (SNMP) [114], SLA compliance monitoring [115]) or by actively injecting measurement probes (e.g. OneProbe[3] [116]), or by using a combination of both techniques. In MPEG-DASH setting, DASH clients typically request the next segment at a bitrate suitable to the measured network throughput (as measured at the client side).

The bandwidth measured during the download of a current segment ($bw(s_{i-1})$) and the buffer fill level at the decision time (bl_i) for the next segment are used to calculate the maximum bitrate of the next segment ($maxbw(s_i)$) in [117] as given in equation (2.4.8.1).

$$\max bw(s_i) = \begin{cases} bw(s_{i-1}) * 0.5 & if \quad 0.0 \le bl_i < 0.3 \\ bw(s_{i-1}) & if \quad 0.3 \le bl_i \end{cases} \tag{2.4.8.1}$$

DASH-JS [118] estimates the bitrate of the next segment b_n on the basis of equation (2.4.8.2) where b_{n-1} is the bitrate calculated for the previous segment, b_m denotes the actual measured throughput for the previous segment, and w_1 and w_2 are the weighting factors used to adjust the influence of the recently measured segment throughput on the previously estimated throughput value. The bandwidth measured during the MPD download is used for initialisation. A number of simulations using $w_1 = 0.7$ and $w_2 = 1.3$ were conducted in [118]. The Overlay Buffer (which mimics the actual buffer) is used for tracking the progress of playout. The implemented adaptation logic does not seem to consider the player's buffer fill level.

$$b_n = \frac{w_1 b_{n-1} + w_2 b_m}{w_1 + w_2} \tag{2.4.8.2}$$

A prototype of an MPEG-DASH client proposed in [111] estimates the available network throughput, controlling the filling level of the client buffer, avoiding playback interruptions,

[3] Available from www4.comp.polyu.edu.hk/~oneprobe [Accessed: 14-Dec-2015]

maximising the quality of the stream, and avoiding unnecessary fluctuations in quality, while minimising the initial delay. The adaptation algorithm uses data on historic throughput and buffer levels to produce the required quality level of the next segment and the minimum buffer level (in seconds of playback) at which the download of the next segment must start.

A receiver-driven rate adaptation algorithm for adaptive HTTP streaming proposed in [119] detects bandwidth changes using a smoothed HTTP throughput measurement based on the segment fetch time (SFT). The smoothed HTTP throughput, instead of the instantaneous TCP transmission rate, is used to determine if the bitrate of the current media matches the end-to-end network bandwidth capacity. The proposed algorithm deploys a step-wise increase and aggressive decrease method to switch up/down between the different bitrates, without requiring transport layer information such as RTT and packet loss rates. The ratio of media segment duration (MSD) to SFT is used to detect congestion and to probe the spare network capacity as indicated in equation (2.4.8.3).

$$\mu = \frac{MSD}{SFT} \qquad \varepsilon = \max\left\{\frac{br_{i+1} - br_i}{br_i}, \forall i = [0,\dots,N-1]\right\}$$

$$t_s = t_m - t_{\min} - \frac{b_c}{b_{\min}} MSD > 0 \qquad (2.4.8.3)$$

The switch to the next higher quality level takes place if $\mu > 1 + \varepsilon$ and the buffered media time is larger than the predefined minimum. Equation (2.4.8.3) defines μ and ε, where br_i denotes the bitrate of quality i and N denotes the highest quality level. A switch to a lower quality level takes place when $\mu < \gamma_d$, where γ_d is the switch down threshold related to the buffered media time and used to detect network congestion before the buffer drains. The idle time before sending the next request, t_s, depends on the buffered media time, t_m, a predefined minimum buffered media time, t_{min}, current bitrate, b_c, and the minimum bitrate b_{min}, and media segment duration - as shown in equation (2.4.8.3). Consequently, the buffer fill level is considered for step down and request timing decisions.

Throughput smoothing by considering historic recordings is also used in [120] and [121]. Throughput variance is used to compute a safety margin for estimated throughput in [120].

However, it can be argued that the smoothed throughput approach delays the reaction of the client to significant drops in throughput, which in turn necessitates a large initial buffering and continuous checking to determine whether the buffer level is lower than a safety threshold [112]. The adaptation algorithm proposed in [112] calculates throughput based on equation (2.4.8.4).

$$T_e(i) = \begin{cases} (1-\delta)T_e(i-2) + \delta T_s(i-1) & i > 2 \\ T_s(i-1) & i = 1,2 \end{cases} \qquad (2.4.8.4)$$

$$p = \frac{|T_s(i) - T_e(i)|}{T_e(i)} \qquad \delta = \frac{1}{1 + e^{-k(p-P_0)}} \qquad (2.4.8.5)$$

The estimated throughput, T_e is more sensitive to the last segment throughput, T_s for larger values of δ, whilst, for smaller values the estimated throughput is smoothed (the value of δ is adaptively controlled). p is the normalised throughput deviation, indicating the significance of change in throughput. Larger changes in throughput, require a quick reaction (δ is set to 1). The values of k and P_0 were determined on the basis of testbed observations and the values used for evaluation were k = 21 and $P_0 = 0.2$ [112].

An Open Source Media Framework (OSMF) adaptation algorithm (as presented in [122]) solely relies on the throughput information based on the time taken for the download of the most recent segment of the requested video.

An algorithm performing bandwidth measurements and enabling dynamic switching between quality levels proposed in [110] calculates the adaptation strategy using a Markov Decision Process. The aims of this process are to (i) minimise the number of deadline misses, (ii) minimise the number of quality level changes and (iii) maximise the chosen quality level. This approach selects a fixed distribution function based on pre-computed network and video statistics without considering the dynamics of the throughput (a numerical evaluation of the approach using fixed, uniform and normal distributions of the available bandwidth was conducted). The controller strategy is determined at run-time by using statistics gathered by the controller from receiver reports of estimated bandwidth and observed chunk sizes. The controller accumulates network and video statistics which are then passed on to the MDP model to enable it to update its strategy.

2.4.9 Segment Size and Duration

Work presented in [77] demonstrates the benefits of dividing segments into fixed-sized subsegments (for example, of size 100 kB) to achieve efficient bandwidth aggregation over multiple links. However, in order to increase performance and video quality, the client requires a buffer large enough (e.g. 5 segments) to compensate for link heterogeneity. Further work presented by the same authors in [123] considers the use of segment sections of variable size. The segment size is dynamically calculated based on the estimated throughput for all links. Links are allocated appropriate shares of a segment, where the slower links are assigned for smaller shares of data.

A study presented in [124] examines the relationship between segment durations and HTTP connection persistence. From a consideration of test results it was concluded that segment duration of between 5 and 8 seconds were optimal for typical network configuration scenarios without persistent HTTP connections (new TCP connection for every HTTP request/response pair). However, segment duration of between 2 and 3 seconds were identified as optimal in the case of persistent connections (single TCP connection used for multiple HTTP requests/responses).

A further study [119] demonstrated that the use of segments of longer duration produced smoother throughput measurements, but resulted in a slower rate of adaptation. Therefore, media segments of approximately 10 seconds duration were identified as sufficient to smooth out the varying instantaneous TCP transmission rate and thus produce a smoothed HTTP/TCP throughput measurement.

2.4.10 Comparison of HTTP Streaming Algorithms

This section provides a comparison of a selection of DASH client implementations in terms of network performance estimation, whether buffer level is considered in the bitrate selection scheme and startup delay considerations in the adaptation algorithms, as given in Table 2-2.

Solution	Performance Estimation	Buffer fill Level	Startup Delay
[111]	Changes of the available network throughput	Considered	Reduce by requesting the lowest bitrate first
[117]	Current measured bandwidth	Considered	N/A
[119]	Last segment fetch time (SFT)	Considered for request timing and step down decisions	Conservative step-wise switch up
[112]	Past history and throughput variance used	Not considered	Reduce by requesting the lowest bitrate first
OSMF [122]	Last segment download considered	Not considered	N/A
[118]	Last actual- measured throughput and previous estimated TP	Not considered	Initial bitrate determined based on bandwidth measured during the MPD download
QDASH [122]	Proxy measured network performance (RTT)	Considered	Starts with lower quality
[125]	measured Bandwidth	Considered	Starts with lower quality
QNOVA [126]	current estimate of mean quality, rebuffering, cost and other quality rate tradeoffs	Considered	Starts with lower quality

Table 2-2: Cross-comparison of DASH Implementations

The remainder of this section provides an overview of relevant studies of the comparative performance of adaptive solutions for HTTP-based video delivery.

A study reported in [106] used different test content (simulated bandwidth traces) to evaluate Microsoft Smooth Streaming, Adobe HTTP Dynamic Streaming, and the Netflix Player. Due to a lack of dataset consistency between the tests applied to the different systems, result comparison is difficult [127]. These commercial player comparisons focused more on their behaviour (e.g. a Netflix client is found to be more aggressive in bitrate change than a Microsoft client for large changes in connection throughput) than on the underlying control algorithms.

HTTP streaming in ***vehicular networks*** (a high-speed vehicular environment, wherein the wireless bandwidth varies significantly and rapidly) was also investigated. Real world bandwidth traces were evaluated using a proprietary client in [128], where HTTP streaming (using the authors' own system) was compared with non-adaptive HTTP streaming (progressive download). This study demonstrates that dynamic HTTP streaming is an effective solution for mobile networks and outperforms non-adaptive HTTP streaming. Testing [127] compared the performance of their proprietary MPEG-DASH system with Microsoft Smooth Steaming, Adobe HTTP Dynamic Streaming, and Apple HTTP Live Streaming and led to the conclusion that "DASH could potentially become a major driver for mobile multimedia streaming" [127, p. 37].

An experimental study using an MPEG-DASH player implementation presented in [129] compares three MPEG-DASH player algorithms using 4 s segments, a history of 6 segments, as well as minimum (12 s), optimal (30 s) and maximum (50 s) buffer levels. The authors report that the algorithm proposed in [130] performs better than the others in terms of response time, however, it is prone to buffer under runs. Algorithms proposed in [111] and [119] exhibit stability in buffer levels and available bandwidth utilisation, at the cost of a longer startup phase when tested under stable network conditions. Furthermore, the algorithm from [111] excels in maintaining stable buffer levels and smoother playout even under highly unstable network conditions, while the others exhibit a significant tendency to display oscillating video qualities.

2.4.11 QoE Aware HTTP Streaming

QoE-aware DASH (QDASH) [122] system measures available network bandwidth and deploys a QoE aware algorithm to determine video quality levels. QDASH deploys a bandwidth measurement module on a hardware proxy directly connected to the media server for accurate bandwidth measurements and uses probes to determine RTT. A QoE-aware switching algorithm, run prior to next segment request, calculates intermediate quality levels in case of down-switching. The intermediate level is chosen based on the buffer size in video seconds and current segment quality. The idea is to request the next segment in higher quality if the buffer fill level is sufficient. QDASH was evaluated using subjective tests [122].

QoE-enhanced adaptation algorithm over DASH (QAAD) [125] is a rate adaptation algorithm which considers current player buffer status and preserves minimum buffer size to cope with fluctuating network conditions and achieve seamless video streaming. The deployed rate adaptation algorithm preserves the minimum buffer length to avoid stalls and minimises the video quality changes during playback. Experimental evaluation indicates that QAAD outperforms QDASH in providing stabilised quality levels without playback interruption in the setting with periodic bandwidth fluctuations. The QAAD Bandwidth Estimation Scheme uses periodical estimation where bandwidth is calculated and then smoothed using weighted moving average. The QAAD Bitrate Selection Scheme considers the current buffer status, previous bitrate and the estimated available network bandwidth.

The Network Optimization for Video Adaptation (NOVA) [126] framework for multiuser joint resource allocation is based on user preferences and a simple QoE model, as well as capacity and video content variability. An online algorithm maximises QoE under rebuffering, cost and network constraints. The network controller carries out resource allocation (e.g., bandwidth) to maintain video quality levels and to reduce violations of rebuffering and cost constrains on the user side. The optimisation algorithm (i.e. QNOVA) for video adaptation at the client side chooses the quality of the next segment so that it is close to the current estimate of mean quality, and thus avoids high variance in quality. Furthermore, the algorithm penalises quality choices leading to large segment file sizes when there is increased risk of violation of rebuffering and cost constraints.

2.4.12 Solutions Aware of Previously Downloaded Content

The idea of downloading media content from peers that have previously downloaded the required content is commercially deployed by Spotify [131] providing significant reductions in infrastructure and bandwidth requirements while maintaining QoS levels.

The peer-assisted DASH (pDASH) system proposed in [117] modifies MPD files to allow use of parts of segments from randomly selected peers which have previously downloaded the segments. pDASH addresses three issues related to its deployment context. (a) It divides segments into chunks (e.g. 1/8 of the segment) to address issues associated with the limited upload capacity of Internet connections at peers, where the uplink capacity is typically one eighth of the downlink capacity. (b) As peer-stored chunks are requested randomly and the pDASH clients discard requests when a maximum number of concurrent connections is reached, the player's download algorithm needs to handle two segments in parallel (peer-chunks and server-segments). (c) pDASH clients' cache size (buffer) is required to be of sufficient size to serve content to other peers. pDASH deploys a central Segment Tracker which processes and logs each segment request made to the web server, and an MPD generator which generates MPDs that integrate BaseURLs of all clients having segments which are part of the requested video. This solution was evaluated in a simulated environment in terms of utilisation of the network link to the server and the amount of content requested from the peers. However, no findings about the quality of the video playout at the client side were indicated. The presented results indicated up to a 25% reduction in the server bandwidth could be achieved, which when converted to infrastructure cost, has a significant business impact. Furthermore, the simulations indicated that for some segments, more than 50% of content was downloaded from peers.

2.4.13 Approaches to Quality Evaluation for Adaptive HTTP Streaming

Historically, UDP streaming was the method of choice for video delivery. Numerous attempts have been made to study [132] and determine [133] the QoE as a function of objective-technical parameters for both the delivery network (e.g. throughput, delay, jitter, loss) and the delivered video (e.g. resolution, frame rate, bitrate, compression). The focus has been on the spatial aspects of the delivered video, and less often on the temporal aspects of the video.

The recent shift to TCP-delivered video content (e.g. MPEG-DASH), assures reliable, ordered delivery. In this case, while delivery network congestion may cause initial delays and possible buffering interruptions [134], the displayed content does not exhibit video quality degradation [135] due to missing packets. Viewers are, however, particularly sensitive to frequent interruptions due to starved buffers [71]. A recent study [136] measuring over 200 million video viewing sessions confirms that more than 20% of sessions suffer quality issues such as more than 10% of viewing time spent on buffering or a startup delay longer than 5 seconds. Consequently, rebuffering was identified as the principal causal factor underlying problems experienced with the QoE during adaptive HTTP streaming [136] and primarily responsible for QoE variability [134]. The focus has to be placed on the temporal aspects of video quality, and it can be argued that spatial metrics such as Peak Signal-to-Noise Ratio (PSNR), are not applicable in the context of HTTP streaming as dropped packets are retransmitted by TCP [134]. The relationships between the three levels of quality of service (QoS) of HTTP video streaming: network QoS, application QoS, and user QoE were investigated in a study [134], where the correlation between the application and network QoS was characterised by means of analytical models and empirical evaluation. Subjective experiments were used to qualify the relationship between application QoS and QoE, which led to the proposal of an application performance metric that includes initial buffering and mean rebuffering duration. The initial buffering duration had no effect on the perceived quality, emphasising that users are generally willing to tolerate a longer startup delay for an uninterrupted video viewing experience. Conducted experiments involved 13 subjects watching flash videos. The buffer capacity was set to 3 seconds. Regression analysis resulted in the relationship between QoE and QoS given in equation (2.4.13.1) where MOS is Mean Opinion Score and L_{ti}, L_{fr} and L_{tr} are the respective levels of Initial Buffering Time, Rebuffering frequency (how frequently the rebuffering events occur), and Mean Rebuffering duration (the average duration of a rebuffering event).

$$MOS = 4.23 - 0.0672L_{ti} - 0.742L_{fr} - 0.106L_{tr} \qquad (2.4.13.1)$$

The data transmission performance of adaptive streaming over HTTP could be measured in terms of: Join time (initial buffering time i.e. the time that lapses from the initiation of the connection until the client buffer reaches playout level); Buffering ratio (the relative time spent in rebuffering, calculated as the total time of buffer starvation over the total length of playout including pauses for rebuffering) (Buffering percentage used in [71]); Rate of buffering events (relative frequency of induced interruptions calculated as the number of buffering events over the playout time) (similar to Buffering frequency used in [71]); and Average bitrate (the average of bitrates played), as proposed in [51]. The authors measured engagement at two levels: (a) View level (play time metric - duration of a viewing session) and (b) Viewer level (metric - the number of views and the total play time by a viewer). Rebuffering was observed to be the most critical factor in determining user engagement [51] and hence the most important quality metric. The impact of JoinTime on view-level engagement is significantly lower compared to the other metrics, however it becomes critical for viewer-level engagement as it negatively impacts on customer retention.

A relationship between viewer QoE and MOS and HTTP adaptive streaming video quality has been drawn from a study [108] where 7-13 viewers were used per test condition. Burst packet loss (repeated requests for dropped packets) was identified as the most devastating condition leading to MOS values between 1 and 2. A non-reference approach was used to evaluate the quality of video based on the video bitrates, where a 3 mbps change in video quality resulted in a MOS score change of 1. The clients tested requested inappropriate video quality levels in highly congested or corrupted network scenarios, leading to the conclusion that TCP goodput was a better reflection of the QoE that determined the requested video quality levels. While the study involved commercial clients, its user relevance could reasonably be generalised: users are more concerned with the variance (variability) of video quality rather than the mean (average level) of video quality. Hence, the proposed MOS prediction formulas – PMOS for Apple (equation 2.4.13.2) and Microsoft (equation 2.4.13.3) HTTP adaptive streaming are based on the mean (μ) and standard deviation (σ) of video quality. The coefficients shown in equations (2.4.13.2) and (2.4.13.3) were determined on the basis of a least mean squares (LMS) approach.

$$ApplePMOS = 1.36\mu - 1.87\sigma + 1.86 \qquad (2.4.13.2)$$

$$MicrosoftPMOS = 0.91\mu - 1.95\sigma + 2.06 \qquad (2.4.13.3)$$

The trade-off between the initial delay (wait before service commencement) and “stalls” (the interruptions during service consumption) has been investigated in subjective laboratory and crowdsourcing studies [137]. The impact of initial delays on QoE depends on the application, and a number of mapping functions were proposed. For example, functions based on the laboratory evaluation of YouTube video streaming, mapping the initial delay (T_0) to MOS and mapping the duration of stalls (T_1) to MOS for 60 second videos, are given in equations (2.4.13.4) and (2.4.13.5).

$$MOS = -0.862 \cdot \log(T_0 + 6.718) + 5 \qquad (2.4.13.4)$$

$$MOS = 1.175e^{-0.334T_1} + 3.19 \qquad (2.4.13.5)$$

2.4.14 Server/Host Selection

Server selection is a frequent task in distributed environments and it is typically based on performance estimators. These estimators [138] (or a combination of) can be classified as: (a) *Static estimators* estimate resource capacity (hardware resources, number of hops, connection link bandwidths, etc.), but not the system availability, (b) *Statistical estimators* rely on past system performance (e.g. latency and bandwidth) and reflect typical resource availability (less reliable when variability is high), (c) *Dynamic estimators* determine current network and/or server conditions using probes (introducing overhead traffic) and closely track resource availability in the absence of rapid fluctuations.

Selection algorithms can be grouped into [138]:

Network-side: (a) Router-based (rely on router metric): (e.g. IPv6 [139] anycast), (b) Domain Name System (based on DNS parameters): for DNS-level load balancing (e.g. "DNS resolution is exploited by YouTube to route clients to appropriate servers according to various YouTube policies" [140, p. 2]);

Server-side (based on server load and context such as energy consumption, heat, etc.): frequently used for server workload/utilisation optimisation, data centre workload management (e.g. HTTP Redirect, IP address rewrite, etc.);

Client-side (utilising geography, hops, RTT, bandwidth, latency, prior response time, random, etc.): used for differentiation of servers/services based on non-functional properties such as QoS.

There are also hybrid deployments where both local and global constraints are considered (e.g. DONAR mapping nodes [141] consider both client performance and server loads among other factors).

Most prediction algorithms offering optimal solutions are based on off-line analyses. For example, genetic algorithms, convex optimisation algorithms [141], integer linear programming techniques [142], mixed integer programming (MIP) [143] may achieve a near-optimal prediction quality after a learning time. Heuristic selection algorithms [144] for combinatorial and graph models offer near-optimal solutions in polynomial time making them more suitable for run-time decisions. One key requirement of our solutions is that the analysis and the prediction algorithms must be executed with minimal computational overhead to meet real time prediction deadlines. While the learning capabilities and the accuracy of off-line models provide optimal solutions, the time limitations require solutions which achieve satisfactory (not necessarily optimal) predictions quickly.

QoS-based service selection uses Time Series Prediction models to extrapolate the future QoS based on monitored data (i.e., past observations). Typically, predictors include models such as windowed means (moving average) or exponential smoothing [145]. An empirical study [146] compared four different approaches for QoS forecasting: average value, current value, linear model and Box-Jenkins Auto Regressive Integrated Moving Average (ARIMA). The time series of recorded response times were collected for 10 services over 4 months. Evaluation results show that a more complex model ARIMA, exhibited a prediction error significantly lower than the average value model, however the benefit is limited in terms of magnitude as the effect size is almost trivial (i.e. d=0.11).

CDN selection, when the same content is hosted in multiple locations, remains an important topic, and very relevant for massive content providers utilising multiple CDNs such as YouTube and Netflix. While a number of selection algorithms are open and public, most remain vendor specific and undisclosed.

A detailed study [140] of the YouTube CDN involved two university campuses and three Internet Service Provider (ISP) networks. An analysis was performed of groups of related flows which provided an insight into the mechanisms and policies underpinning the selection of hosting servers. The results reveal that the factors that affect the server selection process include user proximity (RTT) to data centres, server load, the popularity of video content (diurnal effects, limited availability of rarely accessed video, and the need to alleviate hot-spots that may arise due to popular video content). The data collected over a week-long period indicate that, in a given network, most requests are directed to a preferred data centre (with the smallest RTT). This is in contrast to a study [147] conducted on the original YouTube infrastructure prior to its migration to Google which indicated that the direct requests from a network to a data centre were proportional to the data centre size.

Some studies show that video fetch time depends on the popularity of the requested video. For example [148], investigated YouTube as an example of the "best practices" in the design of a large-scale content delivery system and showed that the popularity of video content (e.g. the "video of the day") introduces the need for hot-spot alleviation which is achieved via redirections that can in turn cause delays [140]. Conversely, when sparse video content (not replicated across all data centres) is requested, the accessed server will redirect the request to a server hosting the content [140], or fetch it from some backend data centre [148] and hence introduce larger delays.

A general framework for high-quality video delivery - Control Plane framework proposed in [136] dynamically (ideally midstream) adapts CDN allocation based on global knowledge of network, distribution of active clients and CDN performance. The authors do not consider cases where the client chooses the CDN arguing that there is an inherent need for global coordination across multiple viewers under overload which cannot be achieved with client-side mechanisms. The Framework is a solution to optimal resource (CDNs in this case) allocation problem that uses measurement-driven performance feedback to dynamically adapt video parameters (e.g. CDN selection, bitrate) in order to improve the video quality (e.g. half the buffering ratio in normal scenarios and more than 10 times improvement under more extreme scenarios). This Framework in line with CDN and ISP management approaches benefits from network-wide views. However, the authors are aware of challenges, including: scalability, interaction with CDNs, multiple providers and controllers.

A study [149] investigating its architecture reports that Netflix [107] statically assigns CDN ratings dependent on user accounts. Netflix MPDs contain references to three CDNs and a rating for each network. The CDN ranking (fixed for several days) is agnostic to content, viewing device, time, location and available bandwidth. The players "stay attached to a fixed CDN even when the other CDNs can offer better video quality" [149, p. 1620]. The authors [149] propose using a small number of instantaneous bandwidth measurements, at startup, to dynamically assign to users the best-performing CDN. This would deliver more than a 12% improvement in average bandwidth over the static Netflix CDN assignment strategy.

2.5 Self-Adaptive Systems

The autonomic computing paradigm attempts to eliminate the need for human intervention in complex computing systems operation. Such systems are also known as self-managing, self-adaptive or self-* systems. The adaptation dimension of these systems is responsible for *altering the managed system* (e.g. modifying system behaviour) *in response to the system perceived* external environment and the internal state of the system *to achieve system goals* [150]. For example, FORMS [151], a self-* system reference model, specifies (a) base-level subsystem (application behaviour) and (b) meta-level subsystem (adapting the base-level subsystem behaviour). IBM have proposed Monitor-Analyze-Plan-Execute-Knowledge (MAPE-K) [152], which is a well-accepted adaptive architectural pattern that can be used at FORMS meta-level. While there are simpler adaptive patterns (e.g. Internal Feedback Loop) MAPE-K neatly separates a control loop from the application logic [153] as indicated in Figure 2-7. The control loop *monitors* the system operation through suitable sensors, *analyses* the measured readings, *plans* an adaptation strategy, and utilises effectors to *execute* adaptation of the managed sub-system.

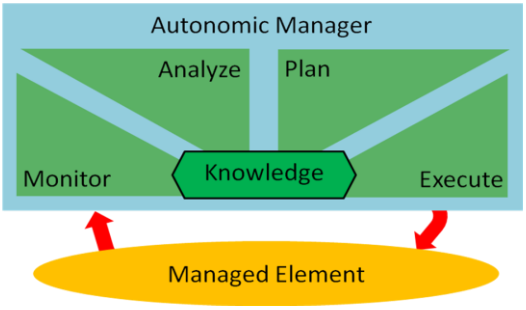

Figure 2-7: Structure of a MAPE-K Element Adapted from [152]

Adaptation reasoning, a core reasoning process in self-adaptive systems [150], links a particular context state to a planned action. It is frequently based on the monitored historical behaviour of the base–level system. Decision making typically employs time series runtime data processing. The task is to identify the most appropriate (possibly optimal) *action* given *the requirement constraints* (adaptation frequency, flexibility, time restrictions, etc.).

While there are a variety of planning techniques deployed in MAPE-K systems, most of them are rule-based, goal-based, and utility-based. There is strong argument that “utility-function policies are much more appropriate for autonomic computing than action policies” [154, p. 41].

2.6 End User Devices

Viewing device capabilities play a significant role in the overall viewing experience. This section outlines approaches to user device identification and classification.

Currently, Web users browse the Web using "devices ranging from mobile phones to domestic appliances" [155, p. 1], and at the same time, they "expect a usable presentation regardless of the device's capabilities or the current network characteristics” [156, p. 92]. The same expectation can be attributed to today's learners, where context-aware ubiquitous learning is an emerging trend in computer-supported learning. The main characteristics of such environments are accessibility (which fosters self-directed learning), immediacy (information is immediately available and permits immediate feedback and correction), and interactivity (instructional activities involving a variety of devices to obtain/interact with learning material). Such systems may be used in a variety of settings, e.g. learning languages [157].

The delivery context can be defined as a set of attributes that characterises the capabilities of the access mechanism and the preferences of the user. The access mechanism is a combination of hardware and software allowing users to interact with the Web using different modalities [158]. End-user device characteristics differ widely. Both hardware and software characteristics are important from educational and multimedia content delivery perspectives. Research in mobile learning reports students' discontent with the size and weight of their PDAs, their inadequate memory and short battery life [159]. Limited storage capacity [160] and slow connectivity [161] were also identified as inhibiting factors. The reasons that keep learners from browsing the Internet more frequently from a handheld device are related to device functionality [5] and are indicated in Figure 2-8. The same study indicates that 33.5% of users (sample size 4552) who daily browse the Internet from handheld devices identify the low speed of the network connection as a reason for not using handheld devices more often [5].

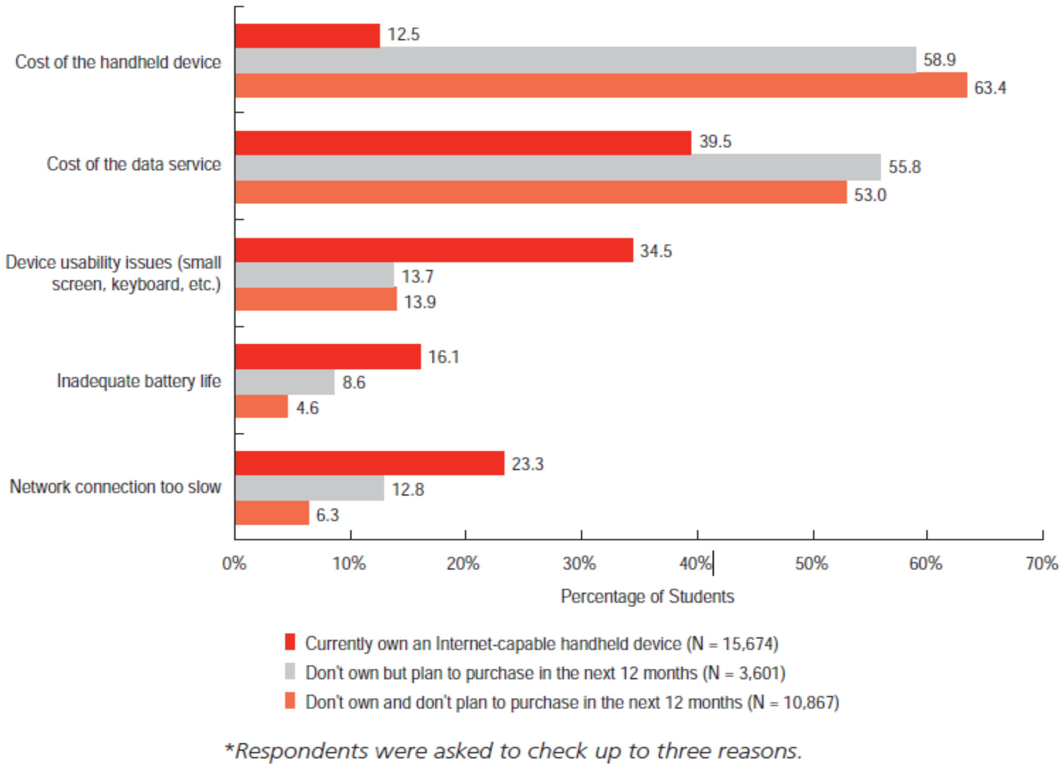

Figure 2-8: What Keeps Students from Using the Internet from a Handheld Device [5]

2.6.1 Device Identification

The delivery context of a device therefore includes its characteristics such as input (e.g. touch screen, mouse, keyboard, keypad, voice input, joystick, stylus, etc.) and output (e.g. visual display, speakers, projector, printer, etc.) capabilities, the level of language support, network connectivity capabilities, software supported by a device, etc. Functional and usable presentations should be provided to the learner regardless of the end-user device used. Therefore, a mechanism is required to allow devices to communicate their capabilities and user preferences to the server in order to tailor responses to cater for particular device limitations. Different approaches to device identification are outlined below.

The User Agent field in HTTP Request Header [20] contains information about the agent originating the request. This field should be included with all requests to allow for automated recognition of user agents in content negotiation. The User Agent field typically contains browser name, version, platform, and in some cases security level and OS/CPU description. An example, “User-Agent: Mozilla/5.0 (Windows NT 6.1)” indicates the version of the Web browser (Mozilla/5.0) and the OS (Windows NT 6.1). While this approach can be used to identify user agents (web clients), more information about device capabilities is required, and hence User Agent field information is not sufficient for more advanced adaptation.

The W3C's Composite Capabilities/Preference Profile (CC/PP) [162] is a description of device capabilities and user preferences and can be used to guide the adaptation of content presented to that device. The specification focuses on heterogeneous mobile devices for browsing the Web and is supported by industry (e.g. IBM, Ericsson). It addresses the lack of a standard way for a client to encode its delivery context and allows a device to identify itself, its capabilities and preferences to a server. The CC/PP framework defines client profiles (instances of a CC/PP vocabulary) as two-level trees of components and attribute/value pairs using Resource Description Framework (RDF) Schema [156]. Each component may be used to capture a feature of a delivery context and may contain one or more attributes. For example, a component that encodes a user's terminal hardware may contain an attribute to specify display width.

User Agent Profile (UAProf) [163] is a CC/PP-based framework developed by the Wireless Application Protocol (WAP) Forum (now OMA[4]) for capturing wireless device characteristics. UAProf device descriptions are stored in the UAProf profile repositories[5] and on the OMA web site. Capability and Preference Information may include five components: hardware characteristics (screen size, colour capabilities, image capabilities, manufacturer, etc.), software characteristics (operating system vendor and version, list of audio and video encoders, etc.), application/user preferences (browser manufacturer and version, markup languages and versions supported, scripting languages supported, etc.), WAP characteristics (WAP version, Wireless Markup Language (WML) script libraries, etc.), and network characteristics, such as latency and reliability.

---

[4] http://www.openmobilealliance.org/ [Accessed: 14-Dec-2015]

[5] Such as delicon.sourceforge.net/profiles.html [Accessed: 14-Dec-2015]

Although it can be used for adaptation to the device capabilities, not all devices have UAProfs, some profiles are inaccurate or poorly structured causing parsing problems, and retrieving and parsing UAProfs can be time consuming [164].

Microsoft proposed Universal Plug and Play (UPnP) [165] aims at device independent interconnection as a standard for universal connectivity between computers and mobile devices. UPnP is suitable to peer-to-peer network connectivity of smart appliances, wireless devices and PCs. Device descriptions can be found in UPnP database[6].

Wireless Universal Resource File (WURFL) Device Description Repository [166] identifies capabilities of the current viewing device in an XML file (devices grouped by manufacturer and browser software). This approach uses the user agent string (provided with the HTTP request) to index the file in order to obtain the device capabilities.

A cross-comparison of the above approaches is provided in Table 2-3 and extends a similar table provided in [167].

	CC/PP	**UAProf**	**UPnP**	**WURFL**
Proposer	W3C	WAP Forum	Microsoft	Luca
Standard used for device profile creation	RDF	RDF	XML	RDF/XML
Device profile format	XML	XML	XML	XML
Vocabulary in device profile	User-defined based on application	Designed and developed by WAP forum	Provided by vendors	Provided by vendors or users
Device profile transmission protocol	HTTP	WSP	HTTP	HTTP
Flexibility for application design	High, developers can create their own device profile vocabularies	Low, the UAProf can be viewed as an application of CC/PP	Low, the device profile description has to be provided by vendor	High, developers can create their own device profile vocabularies

Table 2-3: Comparison of Standards for Device Profiling

A framework for building a comprehensive learner device context profile proposed in [168] aims at providing information about the available functionalities/features (Internet connection types, existing sensors, camera, keyboard, touch screen, etc.) on a learner's device as well as the frequency of their use by the learner. This framework enables a "system to automatically identify, monitor and visualize the availability and usage of device functionalities/features in mobile devices and desktop computers" [168, p. 149].

The above approaches to device identification provide simple solutions, however they are not without issues: the queried databases may not be regularly updated and thus may contain

[6] http://www.upnp-database.info [Accessed: 14-Dec-2015]

obsolete information, descriptors for recently released devices might be missing, etc. Further overviews of device profiling approaches may be found in [169] and [167].

2.6.2 Device Classification

There are a number of approaches to the classification of end-user devices. For example they may be grouped using two orthogonal dimensions of personal vs. shared and portable vs. static [170]. Devices can be grouped on the basis of software and hardware characteristics. Software characteristics include the browser capabilities (e.g. standards supported such as WAP, HTML5, etc. and the markup language) or the platform capabilities. Hardware characteristics also affect interaction style and include device output capabilities (displays size, shape, colour support, etc.), input capabilities (keyboard, touch screen, stylus, etc.), processing power, storage capabilities (volatile and nonvolatile), data connection (e.g. standards supported, bandwidth and the time to connect), etc. Devices can be classified according to their bandwidth and support for wireless/wired modes of communication (IEEE 802.11x, IEEE 802.3, IEEE 802.15.1, etc). Mobile and conventional devices can be classified into five groups according to their display characteristics in terms of resolution, viewable display dimensions and the number of colour bits as in [171]. Device capabilities [172] can be grouped by other attributes that influence the media presentation, including supported media types, display capability, audio/video capability and operational capability defined by attributes including memory, CPU, operating system, etc.

Type	**Resolution (pixel)**	**Colour Depth (kilobytes)**	**Battery Power (mAh)**	**CPU Power (GHz)**	**MM support**
Handheld Devices	160 x 120	32	1100	0.1	70%
	320 x 240	64	1500	0.3	
	640 x 480	128	1800	0.5	
Portable Devices	640 x 480	128	2400	1	90%
	800 x 600	256	3200	1.5	
	1024 x 768	512	3800	2	
Large Screen Devices	1024 x 768	256	3800	2	100%
	1280 x 1024	512	5000	2.5	
	1600 x 1200	1024	5000	3	

Table 2-4: Device Characteristics Classification

A classification proposed in [173] considers a comprehensive set of device characteristics including display resolution, battery power, colour depth, CPU power and multimedia (MM) support. Three classes of devices are proposed, namely Handheld Devices, Portable Devices and Large Screen Devices. Considered device features (screen resolution and colour depth, battery and CPU power, as well as multimedia support) and corresponding classifications are summarised in Table 2-4.

Video Level	Bitrate (kbps)	Resolution (width x height)
l_0	300	320x150
l_1	700	640x360
l_2	1500	640x360
l_3	2500	1280x720
l_4	3500	1280x720

Table 2-5: Akamai Adaptive Streaming Video Levels [174]

A study [174] which investigated Akamai[7] adaptive streaming identified five bitrate levels and three resolutions as shown in Table 2-5. This association could be used for device classification into three classes based on the screen resolution.

2.7 Summary

In the context of video streaming, different approaches are used to achieve quality viewing experiences when heterogeneous viewing devices are used and/or where video content is transmitted over unreliable, best effort networks. The goal is to ensure an uninterrupted viewing experience, which is typically achieved by video bitrate adaptation in response to environmental conditions, such as user preferences, viewing device capabilities and/or delivery network context (e.g. network conditions such as bandwidth fluctuations). This chapter has introduced the technological background and context for the algorithms developed in this research work. The chapter provides an overview of technical requirements for the transmission of video files, with a particular focus on Quality of Service and end-user Experience (definitions and approaches to QoS/QoE evaluation have been provided). The proposed solutions are DASH-based, and hence a considerable portion of this chapter has been devoted to DASH-related issues. Self-adaptive systems have been introduced as the proposed solutions provide adaptation. The solutions to the problem of ensuring consistent high quality, uninterrupted video viewing consider both network conditions and the learner's device. Consequently, different approaches for device identification and classification have been introduced.

[7] https://www.akamai.com/ [Accessed: 4-Jan-2016]

3 Web-based Learning Systems

This chapter presents a comprehensive review of learning systems used in a university setting to describe a context for the solutions presented in this thesis. The focus is on adaptive learning systems that deliver content over the Internet. As "the demand for education is escalating around the world" [175] and free educational content becomes widely available, the Internet is seen as a means for affordable global distribution. Adaptive systems for learning support are investigated and described in particular detail and relevant educational content modalities and adaptation approaches are outlined. Adaptive Hypermedia Systems [176] provide tailored learning experiences allowing "anytime, anywhere" access where the adaptation process is primarily based on learner characteristics and learning context. The proposal in this thesis is to extend adaptation to consider the delivery network performance e.g. characteristics such as bandwidth, delay. Furthermore, learners may use smartphones, laptops, PCs or even TV to access the learning system, therefore consideration should be given to both end user device and underlying delivery network conditions to ensure high levels of Quality of Experience. As the viewing device and delivery network conditions are part of the learning context, context-aware adaptive systems are investigated and an overview of the research in the area is provided. Given the growing availability of open/free educational content, open and distributed educational systems are outlined.

3.1 A Brief History of Hypermedia

Early hypermedia systems (e.g. Xanadu[8]) dealt with both text and media (graphics, audio, video) and brought improvements in a wide range of application domains. However, they were limited to a closed corpus document base and only the emergence of the World Wide Web in the early 1990's brought hypermedia systems to a global audience where they today play an essential role in everyday life.

The rapid development in the Information Communication Technology (ICT) area and the use of ICT for education has inevitably given learners more flexibility and eased access to educational materials, which in turn has provided opportunities for non-traditional students to join mainstream education. As the Web matured, the demand for systems that considered end users increased and systems became more user-centred and personalised. The Adaptive Hypermedia (AH) approach was one attempt to meet emerging personalisation needs and to address problems with "one size fits all" systems. The first educational AH systems that emerged in the late 1980's and early 1990's employed technology-enabled learning that was akin to an electronic book. Although a number of Web-based systems [176], such as On-line

---

[8]Xanadu Project - http://www.xanadu.net/ [Accessed: 14-Dec-2015]

Information Systems, Information Retrieval Hypermedia, Institutional Hypermedia, On-line Help Systems applied user adaptation and personalisation, the majority of early AH systems were Adaptive Educational Hypermedia (AEH) systems. AEH systems provide training tailored for the learner's personal characteristics, goals, background knowledge, hyperspace experience, preferences, etc. Furthermore, these systems offer navigation support to guide learners through the learning content hyperspace.

3.2 Learning Objects, Content Modelling and Standards

Our solutions could be deployed to enhance content selection from remote servers hosting learning content and this section lists relevant standards organisations and outlines corresponding (most frequently used) standards. Organisations such as the IEEE Learning Technology Standards Committee (LTSC)[9], Advanced Distributed Learning Initiative[10] (ADL), and IMS[11] Global Learning Consortium (formerly called Instructional Management Systems Project) have produced a number of standards which cover issues relating to learning content metadata, packaging and learner profiles. These standards govern information storage, reusability and exchange in the area of eLearning through the definition of fixed data structures and communications protocols.

The IEEE Learning Object Metadata (LOM) [177] standard defines a learning object as "any entity, digital or non-digital, that may be used for learning, education or training". This standard specifies relevant learning object attributes (e.g. type of object; author; owner; format; pedagogical attributes, etc.) to support search, discovery, and retrieval.

ADL Sharable Content Object Reference Model (SCORM) [178], the most popular international standard for educational content is used to represent the modular, sharable learning objects that compose learning materials. SCORM defines a content packaging scheme that wraps the learning objects into standard teaching materials. SCORM separates learning content from its hosting system: Learning Management Systems (LMS) / runtime service (RTS) allowing content reusability. SCORM sequencing is a part of structured content which provides no support for open corpus/external content. Modular Adaptive Learning Systems (MALS) [179] are an example of adaptive systems based on the SCORM standard.

3.3 Adaptive Hypermedia Systems

Brusilovsky defines hypermedia as "a set of nodes or hyperdocuments (for the purpose of brevity we will call them "documents") connected by links" to related documents [176]. The user of a hypermedia system accesses documents in a nonlinear fashion. While document linking provides many advantages (e.g. navigational freedom), the complex task of "finding

---

[9] http://ltsc.ieee.org, https://ieee-sa.centraldesktop.com/ltsc/ [Accessed: 14-Jun-2015]

[10] http://www.adlnet.org [Accessed: 14-Dec-2015]

[11] http://www.imsglobal.org [Accessed: 14-Dec-2015]

one's way around" and selecting relevant material can pose a considerable challenge for the user, creating the feeling of being "lost in the hyperspace". The user must be facilitated in gaining efficient access to relevant information. Furthermore, each user is unique, having distinct characteristics, goals, preferences, background, and interests making personalisation necessary. Two kinds of hypermedia systems emerged to meet the personalisation requirement:

(a) *Adaptable Hypermedia Systems* tailor the presentation according to the user's presentation preferences, background, etc. where the user directly provides information, typically via a dialog or questionnaire. Here, the adaptation changes are performed once – at the time of the user's initial interaction.

(b) *Adaptive Hypermedia Systems (AHSs)* can be defined as systems which can alter their various visible aspects (i.e. their structure, functionality or interface) based on their user model in order to accommodate the differing needs of individuals or groups of users and the changing needs of users over time (this definition combines definitions from [176] and [180]). Adaptive systems provide personalised guidance and adapt the presentation (information, media types, etc.) based on a model of the interaction context (task, user, device, time, place, etc.). A user model profile is gradually developed using implicit inferences based on interaction with the user [181], i.e. it is based on the user's behaviour (browsing actions, page accesses, etc.). Users are unaware of this process in many cases, and apart from initial registration a user is not required to provide any further information.

3.3.1 Architecture and Components

This section provides an overview of a typical AHS structure. All AHS employ a ***User Model (UM)*** built from user knowledge, interests, preferences, goals and objectives, action history, type, style, skills and capabilities, individual traits, experience and other relevant properties that might be useful for adaptation.

A ***Domain Model (DM)*** is a knowledge space that defines the structure and organisation (links, relationships) of the conceptual representation of the application domain (sometimes called a content model). DM is typically a collection of elementary knowledge fragments of various sizes. An ***Adaptation Engine (AE)*** applies the UM to adapt the presentation, information content and navigation structure throughout the interaction with the user. An example of an AHS model is the LAOS [182] model given in Figure 3-1 (page 51), where the Adaptation Model (AM) contains the adaptation specification for the course, the Presentation Model (PM) contains information relating to the presentation of the course and Goal and Constraints Model (GM) contains pedagogical and structural information about the content.

In order to answer individual user requests a typical AHS, first retrieves the user model and subsequently retrieves the domain model to perform adaptation of the requested resources. A developer-oriented insight into the internal structure of AHS in education can be found in [183].

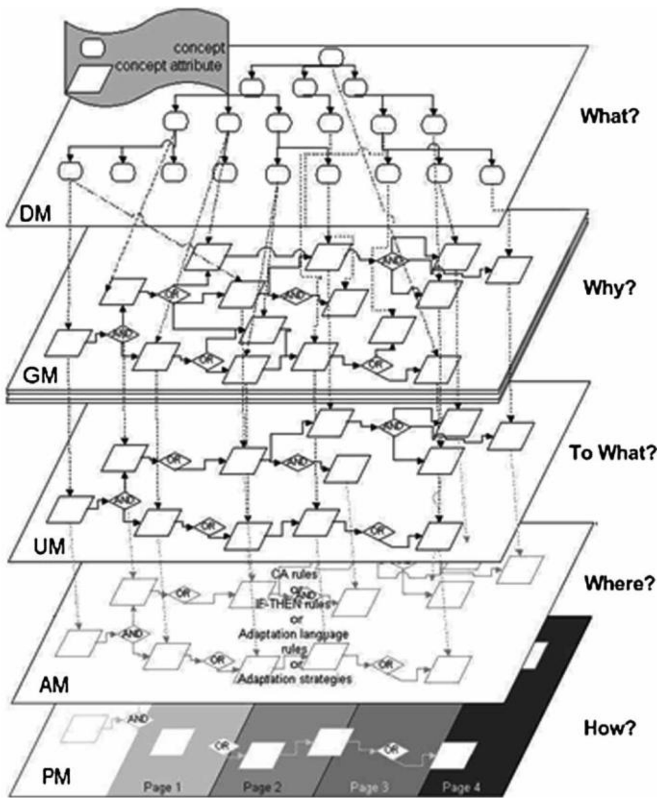

Figure 3-1 Five Layers of the LAOS Model [182]

An explicit ***User Model (UM)*** is a distinctive component of every adaptive system. The UM captures relevant user features, that are collected either implicitly (e.g. UM is updated based on user-AH system interaction) or explicitly (e.g. system requests direct input from the user). Many AEH systems use learners' knowledge to perform adaptation. AEH UMs are frequently called ***student models*** and represent users' existing knowledge within a specific domain. The first AH systems implemented their user models as group competency-based models (e.g. stereotypes, where a user can move to another group when conditions pertaining to the new group are met). Another approach is to employ a so called weighted overlay model to store information about the learner's knowledge levels about each domain item (e.g. a binary value: known/not known, qualitative value: good-average-poor, numeric value: 0-100, probability that the user knows the KE: percentage, etc.). Hence, the learner's knowledge is represented as an overlay of domain knowledge. Today's models are complex domain/skill matrices [184]. Furthermore, while such models were formerly components of a monolithic learning environment, they are now delivered as a service in line with the current trends towards distributed learning frameworks. For example, such a UM can harvest user data from multiple sources (e.g. learning systems) and may be owned and managed independently. An example is the CUMULATE server [185], [186] which has been successfully incorporated within a tutoring system [187]. Furthermore, there are personalised delivery environments such as WHURLE (Web-based Hierarchical Universal Reactive Learning Environment) [188] that support different user models. WHURLE [189] adapts to visual/textual preferences determined based on an online Inventory of Learning Styles [190].

The ***domain (knowledge space) model*** structures and describes the content and serves as the backbone of the AH system. The DM consists of Knowledge Elements (KE) that denote elementary fragments of domain knowledge (e.g. concepts, knowledge items, topics, knowledge elements, learning objectives, learning outcomes). DMs of current systems are of varying complexity ranging from simple set/vector models of unrelated KE (no internal structure) to complex ontology-based networks of interrelated KE. Most frequently used links between KEs are *prerequisite* links, *inhibitor* links and semantic links (e.g. IS-A, PART-OF) which lend themselves to adaptation and user modelling techniques.

Adaptation Model (AM) is set of generic and specific adaptation rules for the content adaptation, navigation adaptation and the user model updates. These rules, for example, can be Condition-Action rules where the rule's action is performed when its condition becomes true or IF-THEN rules as implemented in LAOS [182].

The ***Adaptive Engine (AE)*** tailors content based on the contents of both the DM and UM. The three most popular **adaptation technologies** include adaptive content selection, adaptive navigation support, and adaptive presentation [191]. AE acts as an interpreter for adaptation rules (in AM) and it is typically implementation-depended, while DM, UM and AM describe the adaptation and content at implementation-independent level. In general AHS interactions AE deploys a number of interfaces [10] for monitoring and controlling system usage as follows:

- *User Event Tracker* (e.g. featured in GenericLogDB layer in AHA! [192]) tracks and logs user interactions (e.g. mouse/keyboard events) with the system, which then can be used for UM updates.
- *Behaviour Monitor* uses data provided by the Event Tracker and applies AM rules to modify the UM.
- *Registration* gathers personal information (e.g. questionnaire/form data) used for the initialisation of the UM. For example ProfileDB layer creates new user profiles in AHA! [192].
- *Information Delivery* Interface produces Web pages (collections of DM units) tailored to the UM based on the feedback from the AM.

Furthermore, there are authoring modules for content management, e.g. ConceptDB layer in AHA! [192] creates/destroys concepts, allows concept/attribute searches and creates the adaptation rules associated with the attribute.

The solutions presented in this thesis could be deployed to extend the AH adaptation process, and hence a brief overview of each adaptation approach is given in Section 3.3.2.

Some existing AHS use a ***Presentation Model (PM)*** to provide ***adaptive presentation*** support that tailors information presentation to best suit the user's profile. This approach is particularly useful in educational AHS, where the content presented is adapted to the learner's current knowledge, knowledge growth, progression of competency, goals and other characteristics.

Although techniques for adaptive multimedia presentation exist, the techniques for text adaptation are most studied and used in fully-fledged systems. These techniques can be applied to fragments of information relative to a concept.

3.3.2 Adaptation Approaches

This section provides an overview of different adaptation approaches implemented in AHS. General adaptation issues are presented in Section 3.5. Comprehensive surveys can be found in [176], [193], [194].

Adaptive content selection is performed by restricting current access to learning content. The content can be changed (when content fragments are inserted, removed, summarised via statistical or linguistic analysis) or (de)emphasised (dimming, sorting, scaling text/images, changing text fonts, scaling segments to suggest relevant/important content fragments).

Adaptive navigation support (link-level adaptation) limits the browsing space to the most relevant documents by suggesting links or providing adaptive descriptions for visible links. The approaches to adaptive navigation include:

Guidance. Local guidance suggests the next step - the link to the most appropriate node leaving no other option to the user. Global guidance, aims at finding the shortest navigational path to the most desired information.

Orientation Support provides the user with their “location” in the hyperspace. Local orientation informs about the nodes directly linked from the current node while Global Orientation informs about the whole hyperspace.

Personalised Views allow the user to organise and manage hyperspace by maintaining a set of the most relevant links for a particular goal. A number of different techniques for adapting links were identified in [195].

Other approaches to adaptation include (a) ***structural adaptation*** that gives the user a spatial representation of the hyperspace environment, which in the educational setting may provide the learner with a sense of their position within the environment and an indication of the size of the environment. Structural aids include overview maps, local maps, filters and indexes; and (b) ***historical adaptation*** where history trails, footprints (logged by the system), landmarks (marked by the user) and progression cues are used to represent the user's path through the system, which in turn gives the learner a sense of their current progress.

3.3.3 Advantages and Development Trends

AEH systems offer numerous advantages. As Web based learning systems they provide general eLearning advantages, such as interactivity (simulations, experiments, on-line collaboration with other learners and instructors, video conferencing), media-rich content (searchable media rich learning material in different forms and presentation styles), just-in-time delivery, etc. Furthermore, they offer a personalised user-centric experience that boosts learning outcomes.

Despite the advantages, early AEH systems suffered from a number of shortfalls. For example, although educational, early systems ignored well established pedagogical and instructional design principles. An overview of eLearning platforms provided in [196] indicates three evolutionary generations of learning management systems, starting from monolithic, progressing through modular to service-oriented systems. The same applies to AEH, as initial architectures did not separate in many cases the teaching/pedagogical model, content/domain model and adaptation engine [196]. This approach inhibits reusability of teaching material, and forces instructors to opt for a pedagogical approach at design time. Later AEH systems were centralised by nature limiting the extensibility of the system. When distributed AEH emerged, they continued to deal with closed corpus content domain (content created at design time for the system in question) thus limiting system reusability. Third generation, service oriented systems, such as APeLS [197] are highly modularised, supporting addition of new modules.

Brusilovsky, one of the founders of AH "movement" has voiced concerns regarding AH usage [198], claiming that "almost 10 years after the appearance of the first adaptive Web-based educational systems, just a handful are used for teaching real courses, typically in a class led by one of the authors of the adaptive system." [183, p. 6] and "their inability to meet the needs of practical Web-enhanced education" [183, p. 7]. Almost exclusive focus on adaptive content delivery prevented personalised technologies (e.g. AH systems) from becoming high impact technologies [199]. Slow take-up by learners is due to lack of usability and to the low technical quality of the content delivery (long delays, frequent stoppages, etc.).

Two current trends in AH area can be identified. Firstly, attempting to mimic modern LMS systems by providing as many teacher/learner features and maintaining ability to adapt to the user such as SALMS [200]. Secondly, focusing on the integration of open corpus Web content while providing adaptive guidance for this content [201], [202].

3.4 Open and Distributed Adaptive Hypermedia Learning Systems

Early AH systems were stand-alone dealing with a limited number of well-structured resources known at system design time (so-called closed corpus systems) and although deployed in the Web context, provided no support to incorporate information from arbitrary Web locations. Open Adaptive Educational Hypermedia System (OAEHS) can be defined as "adaptive hypermedia systems which operate on an open corpus of documents, i.e. a set of documents that is not known at design time and, moreover, can constantly change and expand" [201]. By definition, such systems use an open corpus of documents and adapt hypermedia content to the individual needs of the user regardless of the origin of educational material. For example, the materials may be part of a tutorial, may refer to content from a personal Web page or a blog, they could be Learning Objects (LOs) that belong to an open Digital Educational Repository

(DER), video clips from a Massive Open Online Course (MOOC) or YouTube, or excerpts from scholarly research papers.

The first OAEHS [202] was proposed in 2001, since then research efforts shifted focus on personalisation of the access to distributed learning content at service level. Many of the OAEHS separate links from documents. Links are kept in centralised locations for easy maintenance and are processed separately from the media to which they relate [203]. Today, a number of successful personalisation services exist [204], including systems outlined below.

KBS-Hyperbook's [202] indexing approach treats all content units equally regardless of their origin (open/closed corpus). This system separates user knowledge from information resources, so once indexed, all information resources are fully integrated and adapted to the student's needs.

SIGUE [205] attempts to use existing open corpus content by combining parts (e.g. tutorial items) and adding relevant metadata, such as relationships and glossary terms, so that the compiled result will be an adaptive tutorial with accompanying navigation.

While both KBS-Hyperbook and SIGUE attempt to leverage existing content into adaptive resources, the annotation is performed by a teacher as Web items must be manually indexed with domain model concepts in order to be added to the system. Two approaches to automated classifying of open content are machine learning and social navigation which were adopted in Knowledge Sea [206].

Open Corpus Content Service (OCCS) [207] is a content discovery, harvesting and indexing service that deploys a focused Web crawler that traverses open digital repositories and the Web.

MAgAdI [208], [209] is an agent based domain-independent, open and adaptive learning platform for blended-learning, which is one of current trends in education [210].

Other systems that integrate content from multiple providers, while supporting interactivity and personalisation include the eXtensible Tutor Architecture - XTA [211] and MEDEA [212].

3.5 Delivery Context-aware Adaptation

The focus of this section is on AEH systems and learning content delivery context (underlying network conditions and learner's end device). This is an important issue, as handheld, mobile devices are growing cheaper and more powerful while existing learning systems provide media rich content designed with desktop computers and high speed network connections in mind. Such content is generally unsuitable for small screen devices with limited hardware. Therefore, to provide an effective learning experience, the quality of media presentation often needs to be adapted according to the user's preferences and to device capabilities as well as delivery network constraints. The quality of adaptive content delivered over heterogeneous environments depends on a number of factors, such as learner's device (e.g. PC, tablet, smartphone).

Characteristics of the current delivery network connection (e.g. bandwidth, delay, loss and jitter) significantly influence content delivery and are described in Section 2.3.1.

This section defines context and context-aware adaptation approaches. The focus is on adaptive and non-adaptive Personalised Learning (PL) Systems in distributed and mobile environments that consider performance related factors, such as file size, network conditions and user device (terminal). Adaptive learning multimedia content delivery for resource limited devices in an environment is subject to variable constraints and contexts.

3.5.1 Time-dependent Media Content

Video can be engaging, entertaining and thought provoking, replacing lengthy text passages and adding a professional look and feel [213]. Audio is also important for educational multimedia and necessary in some areas such as second language acquisition, music, reading, etc. The approach to the educational video changed over time and three phases can be identified [214]. Firstly, in the 1970's and 1980's, with the use of TV quality video broadcast as educational television (ETV) or delivered by post. In the 1990's and 2000's the Internet become more widely available and the shift to distance education (DE) was pronounced, however the quality of the delivery network limited the use of video. Today, with the improvements in Internet provision and the advent of Massive Open Online Courses (MOOCs) such as ALISON [215], Coursera [3], edX [2], MIT OpenCourseWare [216], Udacity [29], etc. traditional educational institutions "have suddenly embraced not only the use of online DE but also of the ETV medium that predated it" [214, p. 2].

A large-scale [217] study based on 6.9 million video watching sessions across multiple courses on the edX MOOC platform measured student engagement. They considered the time students spent on watching each video and whether the students attempted to answer post-video assessment problems. Video length was identified as the most significant indicator of engagement, where median engagement time is at most 6 minutes, regardless of total video length. The study indicated that the videos of less than 3 minutes have the highest engagement.

ETV seeded interest in incorporating features of entertainment in education: edutainment. The edutainment approach builds on the motivational aspects of a game to aid the learning process and has resulted in a number of serious games developed and deployed in an educational setting. Typical game resources consist of assets and code. In a distributed environment (e.g. online multiplayer gaming), assets can be streamed similarly to media streaming with a strict time requirement (just-in-time).

Educational video content is either streamed or downloaded at the user's request and played at the destination. It is continuous in nature and its delivery involves server and client applications. Technical aspects of video delivery are addressed in Chapter 2.

3.5.2 Context Definition and Components

Context can be defined as "any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." [218, p. 2]. User context is determined by the following components (adapted from [219] and extended to educational context):

- *Environmental context*, such as a user's surroundings, physical location, neighbouring objects or people, lighting, device capabilities [220], underlying communication network load, etc.
- *Personal context* including both physiological data (e.g. blood pressure, heartbeat, weight, glucose level, retinal pattern, etc.), and mental context (e.g. cognitive load, mood, expertise, stress levels, learning style and preferences, background knowledge, etc.).
- *Task context* e.g. explicit goals, actions, activities, events, content modality, etc.
- *Social context* e.g. the role that the learner plays in the context as well as class peers, friends, neutrals, enemies, neighbours, co-workers, relatives etc.
- *Spatio-temporal context* e.g. date, time of day, location, movement, etc.

Comprehensive surveys of context-related issues can be found in [221], [222].

3.5.3 Adaptation Approaches

A survey of approaches to context-aware content adaptation is provided in [223], a subset of which is listed below:

- **Modality Transformation** converts content to a mode that is most useful to the capabilities of a user device. Examples include video to text/audio/image, text to audio, table to plain text/list, image to text, speech to text, or even removal.
- **Data transcoding** converts the data format to match the client device capability including converting images (e.g. colour depth reduction, such as colour to gray scale conversion and format conversion, such as JPEG to BMP, GIF to JPEG, etc.), video (e.g. MPEG to QuickTime) and audio (e.g. Wav to MP3).
- **Information Abstraction** is the process of reducing bandwidth requirements by compressing data (reducing size, quality, data-rate) while preserving the most important information for the user. For example, video highlighting and key-frame extraction, scaling down video and audio streams (frame rate reduction, resolution reduction, region of interest identification) can be applied.
- **Data Prioritisation** applies different quality of service levels for items of different importance to the user. The more important parts have higher priority and are transmitted before less important ones. Examples include using layered coding and

multiresolution compression for images [33], or prioritising regions of maximum interest to video viewers [224], [225].

- **Purpose Classification.** All objects in a page are classified by their purpose using content analysis techniques. Thus, redundant objects (such as images of banners, logos, advertisements, etc.) can be identified and omitted from transmission to devices with limited capabilities (assuming related copyright issues have been addressed) [223].

3.5.4 Temporal Adaptation Factors

Adaptation approaches can also be grouped according to the time of adaptation as follows.

Static Adaptation (off-line): Multiple versions of multimedia information (different quality and processing requirements) are created at design time and stored on the server. As the learner's request arrives, the server selects the most appropriate version to match the user's context and available bandwidth. This approach eliminates processing overheads at presentation time. More storage (not an issue, as the cost of storage is reducing) and clever content management are required. Although server-side adaptation offers maximum author control over the delivered content the document author must predict typical contexts and create appropriate versions (e.g. colour depth for images, or bitrate for video) of the content. However, the quality of the connection (e.g. available bandwidth) to a viewing device may change considerably during a single session since it depends on the mobility of the user for mobile devices and the current load of the delivery network. Therefore, the technical quality of the initially selected document version may become too demanding for dynamic delivery conditions, and client controlled adaptation, such as that offered by DASH can provide better results. Current approaches to client-side adaptation choose bitrates (from a discrete set of bitrates) [12], [99], [100], [105] to match the current delivery context. Evaluations of commercial players suggest there exists scope for improvement in client-adaptation strategies [174][106].

Dynamic Adaptation (on-the-fly): A single video version is created and is modified at presentation (transmission) time. This approach adapts to current conditions (e.g. network load, end device capabilities, user preferences, etc.), however it introduces processing overheads. The World Wide Web Consortium (W3C)[12] defines three types of content adaptation, based on the adaptation location, namely server side, proxy side and client side adaptation as follows:

- *Server Based Adaptation.* The content server performs on-the-fly adaptation through the selection of an appropriate adaptation algorithm based on the client's context profile and current network conditions (e.g. available bandwidth). Such approaches (e.g. [63], [226]) typically employ device detection to send optimised content to the requesting device to match its capabilities. This imposes additional computational load and resource consumption on the server.

---

[12] http://www.w3.org/ [Accessed: 2-Jan-2016]

- *Proxy-Based Adaptation*. The adaptation (typically content distillation and transcoding) takes place between the server and the client and is performed at the intermediary, proxy server. The adaptation is geographically closer to the client, and does not modify settings at clients and servers, nor must the content be re-authored. However, it is only fully successful when based on both knowledge of the target device capabilities and author-provided metadata and adaptation hints [155].
- *Client Based Adaptation*. The final appearance and functionality of delivered content are determined by the client based on the data obtained in the response from the server. The adaptation code has direct access to the device's capabilities. Limitations in processing power, battery and connection bandwidth limit the possibilities for this type of adaptation. Despite the constant improvements in computing power of hand-held devices, this approach remains unpopular and the main responsibility for the adaptation resides with the proxy or server. Therefore, a combination of server-side static content adaptation (e.g. multiple document versions) and client controlled document version (quality) selection such as that offered by DASH provide better results.

Path-based Adaptation. Any node along a network path can participate in adaptation. Due to resource sharing, simulations have shown [227] that this approach can outperform other approaches providing the most robust performance under changing network configurations and across varying servers and clients.

3.5.5 Delivery Network-aware Adaptation in Personalised Learning Systems

Quality of Service (QoS) adaptation is a context-based adaptation, which considers the quality of the delivery network in order to ultimately improve users' Quality of Experience and consequently learning process in an educational setting. A comprehensive study of network conditions and QoS is presented in [226], where different network condition factors were analysed and used for adaptation.

The QoE-aware AHA system (QoEAHA) [63] considers the delivery network conditions. QoEAHA generates recommendations on learning content quality and media type based on the learner's perception of the network delivery performance in order to enhance viewing experience. It enhances the typical AM with performance-related rules to enhance the personalisation process. A QoE Layer is introduced consisting of two new components:

- *Performance Monitor (PM)* monitors and measures QoE-related performance metrics (e.g. download time, round-trip time, throughput, user tolerance to delay and the user's behaviour).
- *Perceived Performance Model (PPM)* provides a dynamic representation of user satisfaction related to the perceived delivery performance. It models user perceptions and generates constraints/suggestions related to the AHS-generated Web page (e.g. number of

components, size of components and overall page size). It is implemented as a set of stereotype user classes having similar performance features, where each class is described with a set of features (F) and a set of suggestions (S) for user perception optimisation. Each user is classified in one or more such classes with a degree of probability.

Furthermore, QoEAHA implements an adaptation algorithm that appropriately transforms (modifying or eliminating embedded components) Web page content based on the predicted impact on QoE to best meet user's expectations based on measured QoS parameters.

Whilst QoEAHA considers download time and RTT, it focuses on static content so transmission of video content is not considered in this solution.

Research presented in [228] uses an assessment of current network conditions by the adaptation delivery engine to calculate what type of content is most appropriate for delivery. Here the decision is made at the content provider's side.

3.5.6 User Device-aware Adaptation in Personalised Learning Systems

End-user device characteristics differ greatly and both hardware and software characteristics are important from the educational content delivery perspective. Software issues, include browser capabilities (e.g. standards, protocols and markup language supported, etc.) and operating system capabilities. Hardware characteristics also affect interaction style and include device output capabilities (display size, colour support), input capabilities (keyboard, touch screen plus stylus), processing power, storage capabilities (volatile and nonvolatile), data connection (e.g. standards supported, bandwidth and the time to connect), battery capacity and current charge, etc. Today's learners are using a wide range of devices to obtain and interact with learning material.

While portability of mobile devices brings many advantages (e.g. accessibility, immediacy, interactivity, etc.), there are issues. For example, students express dissatisfaction with device size, weight and battery life [159], as well as limited storage capacity [159], [160] and slow connectivity [161]. In terms of presentation, discontent with the need for horizontal/vertical scrolling and reduced visibility of images (e.g. diagrams appeared cramped) were reported [229].

The reason for such discontent may lie with the design of the learning content, where content was authored with large screen devices (PC, laptop) in mind. Mobile devices are limited in terms of screen size, network connection cost and quality, user input/output modalities, operating system supported, battery life and processing/storage power. This section describes existing terminal-aware adaptive hypermedia systems and authoring tools.

Adaptive Personalised eLearning Service (APeLS) [230][231] is a multi-model metadata-driven adaptive hypermedia system that is augmented with a number of context-aware features. The system is terminal-aware [232] and dynamically ("on a per session basis") tailors both the navigational structure and appearance of the learning experience to match the current environment of the learner. The terminal model is interpreted by the adaptive engine to select appropriate learning resources during the content selection process. The need for terminal adaptation in a mobile learning setting is emphasised in [231] where an architecture and implementation of dynamically composed eLearning courses for PDAs was proposed. Multiple versions of content for each concept exist, i.e. different types of media (e.g. images, text) and can be used to describe the same concept. The content selection process chooses candidate narratives based on their appropriateness for a given concept and for the device the learner is using. In addition, the most appropriate navigation paradigm is chosen based on knowledge of the learner's device. This approach thus focuses on both the content presentation and navigation issues. The system is extended with a context interpreter [233] to manipulate and translate contextual information.

MAS-SHAAD [220] is multi-agent modular implementation of the SHAAD [234] model that dynamically generates XHMTL pages from content stored in a closed corpus repository based on user preferences and device characteristics. This system was integrated [235] with dotLRN [236] to capture the user device profile and accordingly select the media types of the content resources, their resolution and size. A customised version of an HTML transcoder was used to re-codify pages for handheld devices. dotLRN considers the device screen resolution to choose a suitable resource from a set of resources that explain the same concept, so multiple content versions matching different resolutions must be maintained.

The MobiLearn [237] project is a context-aware generic mobile learning architecture, where the context state (location, activity, device capabilities and learner's input) [238] is used to exclude unsuitable content, while remaining content is ranked by its suitability to the current context. The system both personalises learning content: adapts to user preferences, locations and behaviours; and customises learning content: tailors Web content to the capabilities of the client device (e.g. laptops and tablets, PDAs and smartphones) and the network connection using transcoding.

Intelligent Distributed Cognitive-based Open Learning System for Schools (iClass) project (European Commission FP6 IST Project) [239] is a pedagogically-based system empowering both learners and teachers. In many ways, this system adopts approaches similar to APeLS. Both the chosen pedagogical strategy and the visual preferences of the learner are considered in the process of Learning Object (LO) generation (selecting learning assets from the learning object space and creating/modifying LOs). A repository of contextual data (information about environment, device type, etc.) is maintained.

Mobile Mathematics Tutoring (MoMT) [240] system performs contextual content adaptation using transcoding based on the learner and viewing device characteristics. However, this solution does not consider transmission of video content.

A2M recommender system with the OpenACS/dotLRN [241] identifies the user device by a proxy installed on the client side. This information is used by a device model server to retrieve the device capabilities (the screen size) to limit the number of recommendations obtained so that they fit within the screen.

The solutions presented above benefit from considering limitations of mobile devices, however, none of them consider the delivery network heterogeneity and the network characteristics in general. The following two solutions consider both the viewing device and network conditions.

The Adaptive Display Environment for Adaptive Hypermedia (ADE) [242] deploys a modular content presentation system to adapt to the device type at run-time. Display and contextual adaptation to different devices, screen sizes and connection speeds is supported by extending the LAOS [182] Presentation Model object with the following variables: device (set to the user agent variable in the HTTP request when accessing a Web page), bandwidth (returns an estimate of the network bandwidth, where text-only content is displayed for low bandwidth and videos and audio for high bandwidth) and screenwidth and screenheight (describe the size of the client device screen, and can be used to optimise the layout of the course) [243].

Content authoring is outside of the scope of this work, however it is worth noting that options for device adaptation could be a useful extension to authoring tools for adaptive systems as suggested in [244]. MediaMTool [245] is a simple authoring tool that automatically creates multiple versions of the multimedia clips based on a set of specified multimedia clip features to save battery power on the learner mobile device. QoE-LAOS [173], a performance-aware extension of the classic LAOS [182] authoring model, introduces three sublayers: QoE Content Features sublayer, QoE Characteristics sublayer and QoE Rules sublayer deployed at LAOS's DM, PM and AM, respectively to make the system aware of the viewing device and delivery network issues. Two main models: the Device Characteristics Model (dealing with performance and quality of display) and the Network Characteristics Model (dealing with performance of content delivery network) are introduced at QoE Characteristics sublayer.

3.5.7 Consideration of Social Knowledge (Community Wisdom)

Following in the footprints of others may be a path to more efficient learning experiences. Community wisdom can be used for both social search and navigation. Social navigation can be defined as movement from one item to another "provoked as an artefact of the activity of another or a group of others" [246, p. 1] such as selecting objects because others have been examining/recommending them. Learning systems follow this trend.

CoFIND [247] guides learners to relevant resources based upon the content of the resources and its usefulness (which is generated collaboratively by the users); EDUCO [248] visualizes the information space as clusters of closed corpus documents where currently viewed documents are marked and a user's navigation is made visible to all users at run-time; CRICOS [249] uses background colour of icons to indicate the utility of a resource for the active user, their friends, and users with similar interests .

Knowledge Sea II [206], [250], [251] leverages the collective knowledge and expertise of a large community of learners and past learners' interaction with the system for social adaptive navigation support of both open corpus (e.g. pages of several hierarchically-structured Web textbooks) and closed corpus (e.g. lecture handouts) systems.

3.6 Summary

This chapter sets the application scene for the solutions proposed by this research. It provides an overview of technology enhanced learning systems, with a particular focus on open, adaptive and distributed systems. These systems are well documented and researched. They consider a set of context characteristics, they adapt to match those constraints and as a result they provide an ideal setting for further enhancement with our proposed algorithms. Such an addition would make them less sensitive to unpredictable heterogeneous networked environments. Open adaptive learning systems are typically distributed and their content is stored on remote servers, which extends the geographical distance between the content host and content consumer introducing problems associated with distributed environments e.g. delays, jitter. Our solutions consider both network conditions and the learner's viewing device, so a considerable portion of this chapter was dedicated to context-aware and in particular end user device-aware solutions in the area. As this research focuses on time-dependent media in education temporal adaptation factors are investigated. Ideas behind social wisdom open new possibilities for further enhancement of our solutions and therefore it has been outlined.

Distributed PL systems, despite emerging technologies and continuous hardware improvements, remain vulnerable to congestion in the delivery network, as the rising number of Web users erodes the benefits of new hardware technologies.

4 Proposed Solution Architecture and Algorithms

This chapter introduces the DASH-based Performance Enhancement Architecture (DPEA) (and its related algorithms) proposed to reduce the negative effects of network congestion on the viewing experience in personalised video delivery systems providing external content to a campus area network. The solution is demonstrated using a Personalised Learning system (PL system) deployed in a university campus setting.

As discussed in Chapter 3, the tapestry of campus-based education is changing with increases in class sizes, expanding utilisation of portable networked viewing devices and a growing availability of free educational video material. Recent surveys suggest: “Although students rate network performance as generally good, projected increases in connected devices could soon challenge even the most robust campus networks.” [28, p. 14]. In this context, DPEA aims to enhance personalised learning content distribution systems by taking into account factors that are rarely considered in this research area. For example, Adaptive Hypermedia Systems (see Chapter 3) traditionally focus on learner and learning context characteristics and do not consider the technical aspects of the learning context i.e. device and network characteristics. An exception to this rule is QoEAHA [63], a solution for learning systems that suggests learning content adaptations based on the learner’s perception of the delivery performance in order to enhance their viewing experience (albeit for static content only). DPEA considers the performance of the delivery network and can be used to enhance the video content and the hosting server selection process and thus the quality of video content delivered in personalised systems.

A number of factors affect user experience while viewing video content over MPEG-DASH [12] and via adaptive HTTP streaming systems in general. These factors can be grouped into (a) ***viewer-related personal factors*** (e.g. user preferences, experience, education/training, expectations, cognitive load, etc.), (b) ***network-related factors*** (network throughput, delay and RTT, loss, jitter, etc.), (c) ***viewing device hardware factors*** (device type: PC/TV, laptop/tablet, smartphone, etc., battery capacity, processing power, connection type: wired, WiFi, 3/4G, etc.), (d) ***viewing device software factors*** (operating system, media player and buffer characteristics, etc.), (e) ***content-related factors*** (content genre, spatial and temporal complexity, technical features: frame rate, colour depth, resolution, codec, bitrate, segment length, locality of content, etc.), (f) ***hosting server-related factors*** (server availability, connection quality, response delay, etc.). The subset of factors considered in the DPEA solution is presented in Table 4-1 (page 67).

The DPEA architecture and related components are presented next in terms of their purpose, context of application, block-level architecture and algorithms. The deployment and future extensions are also indicated. Furthermore, conclusions are drawn for each solution.

4.1 DPEA Architecture and Components

The target deployment context for DPEA is a university campus where downloaded learning content ultimately becomes distributed across various nodes within the campus network as depicted in Figure 4-1.

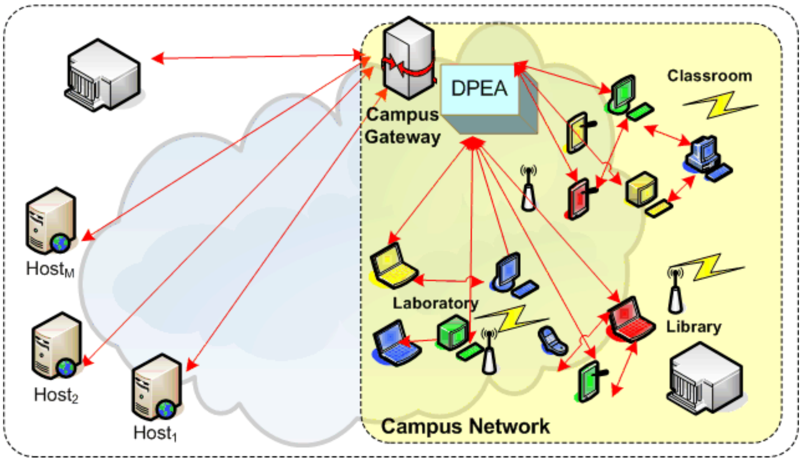

Figure 4-1: DPEA University Campus Setting

The DPEA architecture includes two major novel components: (a) a ***Performance Oriented Adaptation Agent*** (**POAA**), (b) a ***DASH-based performance oriented Adaptive Video distribution solution*** (**DAV**). A high-level illustration of the DPEA architecture is presented in Figure 4-2. The proposed solution does not modify content at remote servers and hence can potentially be used in conjunction with any remote host storing MPEG-DASH content.

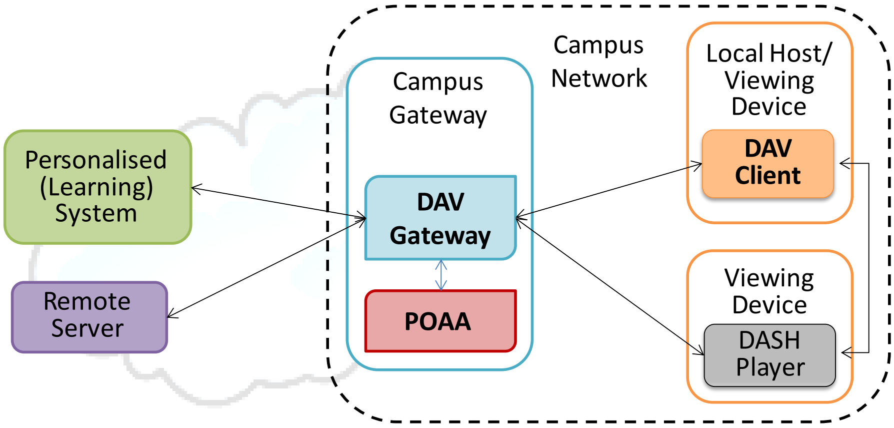

Figure 4-2: High-level DPEA Architecture

4.1.1 POAA

The idea behind the POAA is to select for each Learning Object (LO) request the best performing network path to a server hosting the requested content, based on each server's past performance. This chapter introduces two flavours of the Performance Oriented Adaptation Agent (POAA). Open POAA (oPOAA) [15], [16] was developed for Open Personalised

Learning (oPL) systems such as Open Adaptive Educational Hypermedia Systems (OAEHS) [202], [205] (see Section 3.4). The standalone oPOAA module directly communicates with the associated oPL system to enhance the content selection process. It deals with a variety of media types (e.g. text, images, animations, audio, and video) transported over UDP. It adds network performance aware adaptation to existing adaptive PL systems dealing with open corpus content residing on remote servers/repositories. The oPOAA architecture and underlying algorithms are described in Section 4.2.

As technology-enhanced learning systems matured, and with the growth in the number and types of inexpensive viewing devices and high throughput networks, educational video content has become the medium of choice for online systems. Furthermore, the MPEG-DASH [12] standard for HTTP-based video transmission supports bitrate adaptation at the client side. Therefore, given the shift towards educational video content and the recent MPEG-DASH standardisation, a DASH-aware POAA (dPOAA) [17], [18] is proposed. It focuses exclusively on TCP-transported video content for personalised systems where it performs intelligent selection across remote servers storing identical MPEG-DASH content. MPEG-DASH content is delivered as a sequence of video content segments (see Section 2.4.2) and is described by an MPD file (see Section 2.4.4) containing all required information for video playout, such as location and bitrate for each video segment. MPD files are downloaded by DASH players (see Section 2.4.5). Video segments (see Section A.2.1) may reside on multiple servers, in this case the MPD file contains multiple location URLs and a DASH player can choose between them. While the standard allows specification of multiple URLs, it does not dictate client-side selection algorithms. The dPOAA solution along with its selection algorithms are presented in Section 4.3. dPOAA calculates remote server ratings based on historical server performance information (e.g. throughput) and can be deployed as a Server Reputation Generator (see Section 4.4.2.2) to aid remote host selection during the DAV MPD creation process as described in Section 4.4.2.4. In the proposed setting, dPOAA is based on the campus gateway as indicated in Figure 4-1 and its output is used in the MPD creation process. However, dPOAA could also be used as a DASH player plugin independently of the DPEA architecture. In this case, dPOAA is deployed at the DASH player as described in Section 4.3.5.

4.1.2 DAV

The DASH-based performance oriented Adaptive Video distribution solution (DAV) considers user device characteristics and user profiles as well as the content already available locally to improve the content delivery process thereby increasing the overall viewing experience. While dPOAA aids remote server selection, DAV introduces access to locally stored content. DAV consists of two components: a DAV Gateway based on the campus gateway, and a DAV Client installed on high performing nodes within the campus network as indicated in Figure 4-1. The DAV Gateway dynamically constructs MPD files that contain URLs that point to both local and remote hosts. The DAV Client component informs the DAV Gateway about its availability and about stored video segments. Each DAV Client is equipped with a simple, standard web server that serves cached MPEG-DASH content to other local nodes. Crucially, even devices without a DAV Client installed benefit from the DAV solution, as once provided with modified MPD files, they can request content both from remote servers and local hosts (nodes with DAV Client installed). DAV is presented in Section 4.4.

4.1.3 Solution Summary

Factors affecting playback under MPEG-DASH are indicated at the beginning of this chapter. A subset of these factors is considered in the DPEA solution and these factors are listed in Table 4-1. The emphasis was placed on the important factors that are either already available (e.g. user preference and user class are provided by the associated open PL system) or can be unobtrusively collected (e.g. throughput and RTT are measured by the DAV Gateway, device characteristics are determined based on the HTTP User-Agent header).

Factor	**Unit**	**Source**	**Consumer**
(a) User-related Personal			
User preference and user class	N/A	User Profiler (External PL System's User Model)	MPD Builder
(b) Network Performance			
Connection throughput	bits/second	DAV Gateway oPOAA	dPOAA PL System
Delay (RTT)	milliseconds	DAV Gateway oPOAA	dPOAA PL System
(c) Viewing Device Hardware			
Device Type	Device Class	Device Profiler	MPD Builder
Connection characteristics	wired/wireless	Viewing Device	MPD Builder
Screen resolution	ranges	Viewing Device	DASH Player
Request rate	number	DAV Client (Heartbeat)	DAV Gateway
Availability	Boolean	DAV Client (Heartbeat)	DAV Gateway
(d) Viewing Device Software			
Buffer content	seconds	DASH Player	DASH Player
(e) Content			
Segment bitrate	bps	Server (original MPD)	MPD Builder
Segment length	seconds	Server (original MPD)	MPD Builder
Segment ID	segment list	Server (original MPD)	MPD Builder
Content available locally	segment list	DAV Client	Local Content Elicitor

Table 4-1: DPEA Factors Affecting Video Playout

Table 4-2 summarises the link between proposed solutions and the service each provides.

Solution	Service provided	Utilised by	Introduced in
oPOAA	Server Rating	oPL System	Section 4.2.3
dPOAA	Server Rating	PL System, DPEA	Section 4.3.3
DAV Gateway	Host Selection	MPD Builder	Section 4.4.3.1
DAV Gateway	MPD Building	DAV and DASH Clients	Section 4.4.3.2
DAV Client and Gateway	Heartbeat Updates	Local Content Elicitor	Section 4.4.3.3

Table 4-2: DPEA Component, Mechanism, Consumer and Reference

4.2 open Performance Oriented Adaptation Agent (oPOAA)

open POAA was developed for open PL (oPL) systems. Such systems use existing LOs residing at remote servers e.g. Digital Educational Repositories (DER). Open POAA was proposed to address issues that may arise as a consequence of poor performance of the network connections between the oPL system and the servers hosting LOs addressing research question 1.2.1. The delivery network conditions change frequently and sometimes significantly even during a single learning session. Metrics such as delay, jitter, loss, download time, etc. reflect the state of the network and can be monitored in order to determine performance-based adaptation measures. Such measures include guiding the selection of LOs and/or hosting servers in response to current network conditions. Typically, an oPL system selects LOs that coincide with the given learner's profile. Due to network performance issues, a user might perceive an unacceptable download delay, poor quality of delivered content (e.g. jerky, blocky images, frequent interruptions in the video playback, etc.), as contributory factors in degrading their overall viewing/learning experience to an unacceptable level. Thus a performance-aware enhancement for oPL systems is required that selects the most suitable LOs and hosting servers based both on performance and on the learner specific characteristics.

In this context, oPOAA was introduced to enhance oPL systems by considering network delivery conditions along with personal learner characteristics in the content selection process. A literature review of personalised Web-based learning systems was presented in Chapter 3. The oPOAA extension architecture and components are presented below.

4.2.1 Context

The learning content selection process is triggered at every user request for learning content. During this process a list of suitable LOs is generated by an oPL system. The PL system first selects those LOs that match learning objectives relevant to the learning outcome (relevant LOs). Next a subset of these latter LOs that best match learner characteristics is chosen as the basis for a presentation suitable for the learner before it is finally delivered it to the learner's device. When the oPOAA solution is deployed in conjunction with the oPL system, the content selection process is enhanced to select the best connected remote hosts in order to minimise the content download latency. The LO selection process is illustrated in Figure 4-3.

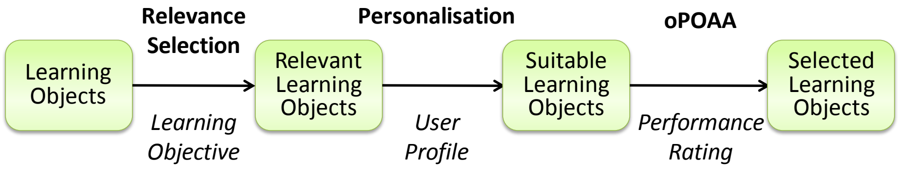

Figure 4-3: Performance-aware Learning Objects Selection Process

The first step is relevance selection where the PL system identifies the learning objective and selects LOs matching the learning outcome. The second step is personalised selection where the oPL system shortlists a number of the most suitable LOs based on the user's learning profile. The oPL system also assigns a LO suitability rating for the requesting learner. LOs may be distributed across several remote DER servers. oPOAA-enhanced selection introduces the third step – network performance-aware assignment where the oPOAA agent estimates a performance rating for each hosting server (for each suitable LO). This performance rating is based on the performance history of the DER hosting the selected LOs.

4.2.2 Architecture and Components

oPOAA continuously monitors network conditions between the oPL system and DER servers to determine network performance without employing an agent at the DER side. Network parameters considered relate to content delivery performance and include download time and delay. They are inferred from historic performance information gathered across a number of recent sessions with the DERs in question. The block-level architecture for the oPL system incorporating oPOAA is shown in Figure 4-4. In addition to the typical components of an oPL system, such as the Adaptation Engine, User Model, Domain Model, Figure 4-4 shows the three new oPOAA components, namely the ***oPOAA Performance Model*** (oPOAA PM), ***oPOAA Domain Model*** (oPOAA DM) and ***oPOAA Performance Engine*** (oPOAA PE).

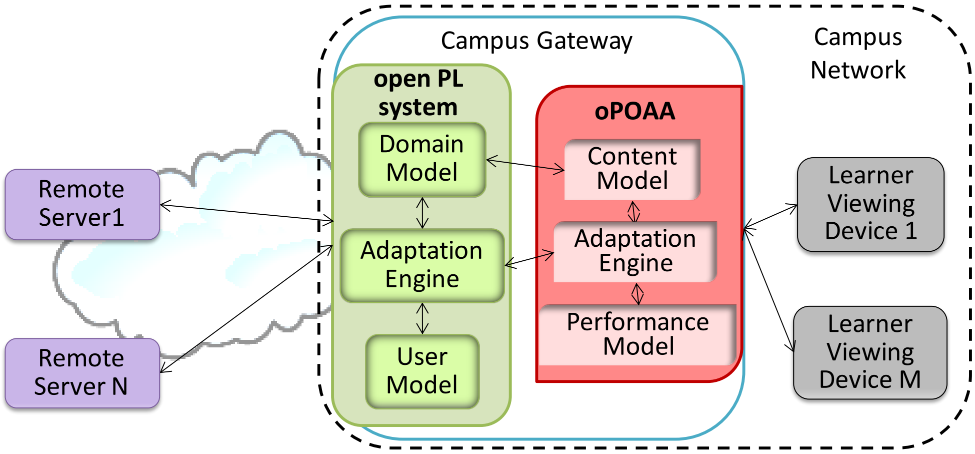

Figure 4-4: oPOAA Block-level Architecture

oPOAA Performance Model (oPOAA PM) is a passive component of the oPOAA. It stores information used by the oPOAA Performance Engine. Each DER is assigned a unique identifier. The oPOAA PM maintains a history log for each connected DER (DER log). The log is a sliding-window structure that contains network performance-related readings for the most recently requested content from a given DER. The following readings are maintained for a number (X) of the most recently delivered LOs from each DER:

- LO_ID: LO identifier, unique within the oPL system Domain Model;
- Delivered: delivered content that reached the learner device (measured in Kb);
- RTT (Round Trip Time): the time required to send a message over a link to this DER and receive a response (measured in milliseconds);
- Duration: the time interval between the content request and the completion of the content delivery (measured in milliseconds);
- Time Stamp: the date and time of the LO request.

Sample content from a DER log is given in Table 4-3. The throughput is calculated as *Delivered* over *Duration* and it is measured in Kbps.

LO_ID	Delivered	RTT	Duration	Time Stamp
Mat980	4480	45	2500	2014-10-30 08:30
Mat344	59	35	300	2014-10-30 10:45

Table 4-3: DER Log - Sample Content

oPOAA Content Model (oPOAA CM) is the other oPOAA passive component. It acts as a link between the oPL system Domain Model and oPOAA PE (as shown in Figure 4-4) and provides information about the LOs from the Suitable and Relevant LOs (SRLO) list (described in Section 4.2.3). LO details, such as ID, size and locations (URLs) are required to perform performance aware selection. Sample LO information is provided in Table 4-4.

LO_ID	LO_SIZE
Mat980	4500
Mat344	60

LO_ID	DER_ID	URL
Mat980	DITDER1	http://www.dit.ie/~lejlar/video/Diff.mpg
Mat980	DCUDER3	http://www.dcu.ie/~lejlar/video/Diff. mpg

Table 4-4: LO Details (a) and Locations (b)

oPOAA Performance Engine (oPOAA PE) is the active component of oPOAA that calculates performance ratings for all suitable and relevant LOs suggested by the oPOAA CM at each learner request. Furthermore, oPOAA PE selects DERs to be contacted and schedules requests for each LO in the SRLO list.

Performance ratings are based on network conditions, therefore the oPOAA PE requires data on the state of the links to the DERs. The quantity of additional traffic introduced by a monitoring solution should be minimised to avoid consuming valuable network bandwidth resources. The proposed solution collects as much information as possible without employing software agents on the DER and learner sides. oPOAA was proposed to cater for end users (learners) who are typically reluctant to install third-party software on their devices. User behaviour has significantly changed with the increased popularity of smartphone applications. However, DER owners and administrators remain reluctant to install and run third party software. Therefore oPOAA PE collects data (request time, requested size, delivery time) for each LO requested and delivered, and calculates DER performance information that is then recorded by the oPOAA PM to DER logs.

4.2.3 Performance-aware Selection Algorithm

This section describes the oPOAA's performance-aware selection process as indicated in Figure 4-5. For each learner request, the oPL system typically generates a list of Suitable and Relevant LOs – a SRLO list. The LO's suitability is based on the oPL system User Model while the LO relevance depends on the characteristics of the current learning request (learning objective).

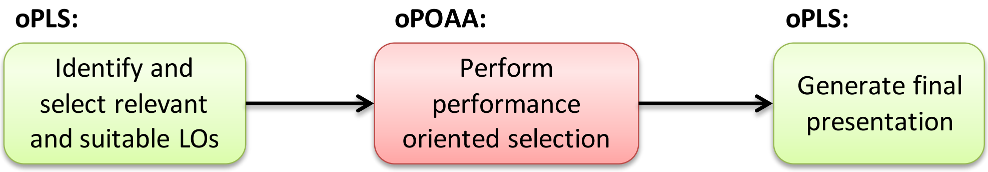

Figure 4-5: oPOAA Algorithm Steps

The SRLO list is then forwarded to oPOAA. Figure 4-6 illustrates the sequence diagram for the subsequent performance-aware selection process.

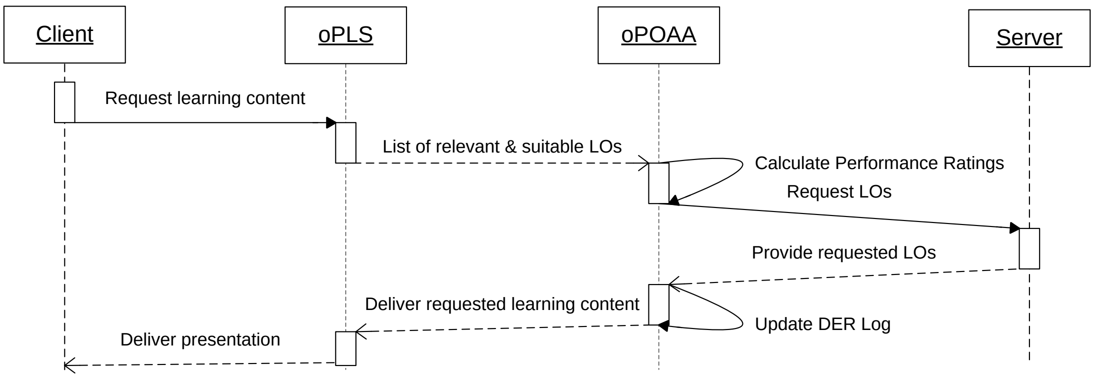

Figure 4-6: oPOAA Selection Process Sequence Diagram

As the oPL system is aware of servers (DERs) containing different LOs, it is assumed that the SRLO list provided by the oPL system contains the following information for each suitable LO:

- LO_ID - LO's identification code (unique within oPL system Domain Model);
- LO_URL - LO's Uniform Resource Locators (URLs);
- LO_SR - LO's suitability rating, ranging from 0 (not suitable at all) to 100 (perfect match to the learner's profile) as provided by oPL.

The oPOAA performance adaptation process begins upon receipt of the SRLO list from the oPL system. The list is processed in order from the most suitable LO to less suitable ones. The oPOAA Performance Engine calculates performance ratings and generates a performance data enriched SRLO (PSRLO) list. This is the SRLO list extended with LO performance data: object media type, object size and a list of alternative locations as follows:

- LO_TYPE - LO's type, namely Text, Image (Graphics) and Multimedia (Audio or Video) determined based on the file extension;
- LO_SIZE - LO's size in kilobytes (kb);
- LO_LOCS - A list of alternative locations (DERs that store the LO).

Sample content of a PSRLO list is given in Table 4-5, where LO_LOCS03 and LO_LOCS04 are lists containing alternative URIs for LO1 (Mat980) and LO2 (Mat344) respectively.

LO_ID	**LO_TYPE**	**LO_SIZE**	**LO_LOCS**
Mat980	Video	4500	LO_LOCS03
Mat344	Image	60	LO_LOCS04

Table 4-5: PSRL List: Sample Content

The content of a LO_LOCS list is derived based on an oPOAA CM table containing data about LO_IDs and associated URLs as given in Table 4-4 (b).

The oPOAA adaptation algorithm selects the LOs from the currently most efficient servers by considering the performance of the DER-oPL system network link. The link performance is calculated based on the logs collected over a number (X) most recent transactions with each DER. The logs are stored for each DER in a sliding window-like structure (in oPOAA PM as indicated in Table 4-3). The DER's sliding window log is updated with new performance information every time a learning object is delivered from the DER. The measured *Throughput* is calculated as the quantity of delivered content (*Delivered*) over the measured delivery time (*Duration*). All log readings are considered to be of equal importance. Therefore, the estimated RTT and throughput of a server Y ($estRTT_{DERy}$ and $estTP_{DERy}$) are calculated as the average of previous recordings of RTT and Throughput. It is calculated for each LO requested from a DER. For each LO_j within the provided SRLO List, beginning with the most suitable, POAA PE calculates expected delivery times ($expDelivTime_{LOjDERi}$) for each DER_i on which the LO resides, based on the size of the LO ($size_{LOj}$), on estimated throughput of the hosting server DERi ($estTP_{DERi}$) and on estimated delay ($estDelay_{DERi} = estRTT_{DERi}/2$) along the network link. The expected download time is calculated based on formula (4.2.3.1).

$$\exp DelivTime_{LO_jDER_i} = \frac{size_{LO_j}}{estTP_{DER_i}} + estDelay_{DER_i} \qquad (4.2.3.1)$$

The DER_S with the shortest expected delivery time for the particular LO is sent a request for that LO. The oPOAA algorithm in pseudo-code is provided in Algorithm 4-1.

```
Input:     LO_LOCSj: List of servers (DERs) hosting content (LOj)
           SRLO: List of suitable LOs
           DER logs: Collated historic DER-oPL link performance data
Output:    SDER: List of DERs that can provide best delivery of the
           LOs under the current network conditions
Algorithm:
    for LOj ∈ SRLO
       for DERi ∈ LO_LOCSj
          expDelivTimeLOjDERi = (sizeLOj/estTPDERi + estDelayDERi)
       endfor
       expDelivTimeLOjDERs = min {DERi ∈ LO_LOCSj : expDelivTimeLOjDERi}
       SDER <- (LOj, DERs)
    endfor
```

Algorithm 4-1: The oPOAA Algorithm in Pseudo-code

This simple algorithm is of low computational complexity. It uses small logs collected over time. The scalability of the solution depends on the number of remote hosts storing learning content utilised by the associated oPL system. The oPL system could deploy content scattered across millions of web servers. However, it could be argued that it is not likely to have copies of a single LO stored on more than M (M < 100) different servers. Most are so far away that they do not need to be considered. A threshold can be introduced where servers with RTT twice as large as the average RTT are not considered. Therefore, while logs about thousands of servers are maintained, the performance calculation considers a small subset (e.g. M) of these.

However, there are limitations as the solution depends on the oPL system to provide LO details and locations (URLs). Furthermore, all recorded performance metrics are considered equally important, so stale logs could affect accuracy of the estimated delivery time which is used for server selection.

4.2.4 Deployment

The proposed delivery performance-aware solution – oPOAA – can be deployed with existing distributed PL system such as Knowledge Tree [198] to augment the current adaptation process. oPOAA could also enhance performance of systems that enable personalised access to distributed heterogeneous knowledge repositories, an example of which is Smart Space for LearningTM (SS4L) [252]. Furthermore, it could enhance tools and their underlying algorithms for selection of LO from DERs. An example of such an algorithm was given in [253]. This classroom-constrained selection algorithm considers class time constrains when selecting learning content from DERs. With the oPOAA enhancement, the algorithm would request LOs that would be delivered without interruptions and within the time allocated for the class.

4.2.5 Future Work

Over the past decade, viewing devices technology and network connections have improved so that downloading and rendering still graphics and text information causes little delay. At the same time, educational video content has grown increasingly prevalent, which still puts a strain on both the viewing device and the delivery network. This was the time of the shift from UDP-based video streaming solutions to HTTP-based streaming, which culminated in the introduction of a new HTTP-based streaming standard, MPEG-DASH [12] [13] (see Section 2.4). Therefore, oPOAA solution that considers the server link performance for various types of content delivered over UDP needed consideration. A new version which focuses on MPEG-DASH video content – dPOAA was developed. Details of dPOAA solution are provided in Section 4.3.

4.2.6 Summary and Conclusions

This section presented the oPOAA extension for open corpus PL systems. oPOAA enhances the existing selection process of learning objects by taking into consideration network delivery conditions. The use of oPOAA in an oPL system brings significant performance improvements in terms of requested content download time. The effects of long download delays are discussed in Section 2.3.2. A reduction in download latency reduces study session time and information processing time per page [75] and is expected to improve the overall learning process.

4.3 DASH-aware Performance Oriented Adaptation Agent (dPOAA)

dPOAA supports informed selection of remote servers storing identical MPEG-DASH content and it can be used to enhance the performance of PL systems utilising such content.

4.3.1 Context

Each MPEG-DASH video is associated with a DASH Media Presentation Description (MPD) document specifying URLs for servers hosting the video. The MPEG-DASH standard and related details were presented in Section 2.4. The purpose and structure of MPD documents and their components were described in Section 2.4.4. Handling of multiple alternative base URLs is addressed in Section 5.6.5. of the MPEG-DASH standard [12]. The standard supports the specification of alternative base URLs through the BaseURL element at any level (i.e. MPD, Period, Adaptation Set or Representation) of the MPD document (see Section 2.4.4). When alternative base URLs exist, identical video segments are provided at multiple locations (remote hosts). When multiple BaseURLs exist at the same level, their order is not relevant as no priority or preference is encoded across the URLs provided. Whilst the standard does not dictate the URL selection process, it states that the client: (a) may use the first BaseURL element as the base URL "in the absence of other criteria" and (b) "may implement any suitable algorithm to determine which URLs it uses for requests" [12, p. 58]. In keeping with the standard, we propose the use of dPOAA to generate server ratings which can be used for specification in MPDs of the best performing server where the requested content is available from multiple servers. dPOAA could also be deployed as a plug-in for DASH players in order to help choose between servers when multiple BaseURLs are listed. dPOAA aims to improve the quality of delivered video in terms of rebuffering and initial delay on the viewer side.

4.3.2 Architecture and Components

The learning content selection process in a Personalised Learning (PL) system is triggered at each learner's request for learning content. At that time the PL system identifies the video content that best suits the learner characteristics. Once videos with matching learning objectives are identified, the PL system builds a presentation suitable for the learner before finally delivering the presentation to the learner's device. The presentation contains links to relevant MPDs. Once a DASH player downloads an MPD file it will parse it and request media segments from the specified remote server(s). Multiple server URLs are specified in the MPD file when the chosen video resides on multiple servers.

dPOAA enhances the remote host selection process by providing server ratings, so that segments are requested from the currently most efficient remote host. This in turn will result in better quality of video playout and reduced initial download latency. To calculate its ratings, dPOAA depends on information about network connections to remote hosts. This information can be either collected from the local viewing device (illustrated in Figure 4-11) or provided by DAV Gateway (illustrated in Figure 4-10). Throughput is inferred from historic performance information gathered during the most recent sessions with the remote servers. Network parameters considered include download time and segment size. The block-level dPOAA architecture given in Figure 4-7 illustrates two dPOAA components, namely the ***dPOAA Performance Model*** (dPOAA PM) and the ***dPOAA Performance Engine*** (dPOAA PE). The server URLs from the original MPD are provided by the *DAV Gateway*. The URL of the best performing server is then passed to the *DAV Gateway* and the new MPD is built accordingly.

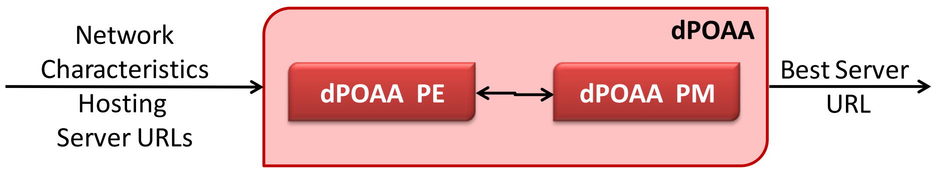

Figure 4-7: dPOAA Block Level Architecture

dPOAA Performance Model dPOAA PM is the passive dPOAA component. It stores information used by the dPOAA PE. The dPOAA PM maintains a log for each contacted hosting server. The log is a sliding-window structure that contains readings for the X most recently requested segments from a given server. Each server is identified by its URL (its unique ID - Server_URL). The following data are maintained for the X most recent segments delivered by each server:

- *TPut*: measured throughput calculated as *download size/ duration* where
  - *Download size* is the size of the content delivered measured in kilobits
  - *Duration* is the difference between the segment delivery completion and request times in milliseconds

- *RTT*: measured time calculated as the difference between the times of the first byte of the segment arriving and the request time in milliseconds
- *Time Stamp*: the date and time when the segment is requested

Sample content from a server log is given in Table 4-6 .

Server_URL	TPut (Kbps)	RTT (ms)	Time_Stamp
http://dbq.multimediatech.cz[13]	4480	45	2014-12-30 08:30
http://streaming.polito.it	6900	25	2014-12-30 10:45
http://emmy9.casa.umass.edu	5000	35	2014-12-30 10:46

Table 4-6: dPOAA PM Server Log: Sample Content

dPOAA Performance Engine (dPOAA PE) is the active component of dPOAA. It calculates performance ratings for all remote servers hosting requested video when a learner requests it. Server performance ratings are based on network conditions. Thus, the dPOAA PE requires continuous updates on the state of the links to each server. Our aim is to collect as much information as possible without introducing additional traffic or deploying agents on the remote server side. Therefore the dPOAA PE bases the selection process on the details provided by DAV Gateway collected for each segment requested and delivered from the hosting server.

Note how the dPOAA differs from the oPOAA approach. The differences between parameters collected under oPOAA vs. dPOAA include: (a) oPOAA logged download size (*Download*) and duration (*Duration*) values separately, dPOAA logs throughput (*TPut*) so that only a single value is logged; (b) oPOAA logged LO IDs, dPOAA does not, as it is not relevant for link throughput estimation; (c) dPOAA utilises time stamps; (d) dPOAA is not as tightly coupled with the associated PL system.

4.3.3 Host Performance Calculation Algorithm

This section describes the process behind the host performance calculation. The hosting servers differ in response time and availability (varying performance, load, etc.) as well as in the quality of connection (e.g. throughput, delay). dPOAA considers the quality of the connection and generates a Server Rating - R_n, based on the utility function using normalised throughput and RTT as given in equation (4.3.3.7).

- Tp_{nL} denotes the throughput for the most recently downloaded segment from server *n*, whilst
- Tp_{nX} denotes the weighted average throughput over the X most recent requests. Tp_{nX} is added to make the approach less sensitive to short term fluctuations in the server connection throughput. It is calculated using equation (4.3.3.1).

In equation (4.3.3.1), w_i is the weight factor for a throughput measurement Tp_{ni} ($1 \le i \le X$), and it reflects the freshness of the recorded throughput for server *n*. The variable t_c denotes current

[13] URLs obtained for redbull_6sec.mpd [Accessed: 22-Dec-2014]

time and t_d denotes download time. The value of w_i equals 1 for throughput recorded within the past ε seconds, and is lowered as time passes so as to reduce the impact of stale measurements.

$$Tp_{nX} = \frac{\sum_{i=1}^{X} w_i \cdot Tp_{ni}}{\sum_{i=1}^{X} w_i} \qquad w_i = \begin{cases} 1 & |t_c - t_d| \le \varepsilon \\ \dfrac{\varepsilon}{t_c - t_d} & |t_c - t_d| > \varepsilon \end{cases} \tag{4.3.3.1}$$

Using previous throughput logs to smooth the current server throughput estimate may slow reaction to major drops in available bandwidth, which could cause problems (e.g. assigning a high rating to a server that is currently overloaded). Reaction time is controlled by a factor w_1 determined using an exponential function that produces values from 0 to 1 and is sensitive to changes in throughput. The factor w_1 is calculated using the formula given in equation (4.3.3.3). The normalised throughput - *normTp* is calculated in (4.3.3.2) where *maxTp* is the highest throughput recorded for any remote server.

$$normTp_n = \frac{(1-w_1)\cdot Tp_{nX} + w_1 \cdot Tp_{nL}}{\max Tp} \qquad 0 < w_1 < 1 \tag{4.3.3.2}$$

$$\Delta = \frac{Tp_{nL} - Tp_{nX}}{Tp_{nX}} \qquad w_1 = \frac{1}{1+e^{\Delta}} \tag{4.3.3.3}$$

The exponential function ensures the server rating is quickly adjusted to decreasing throughput, where Δ is a normalised throughput difference. Drops in measured throughput will make w_1 larger than 0.5, which favours the most recent throughput measurements. Figure 4-8 plots w_1 against Δ.

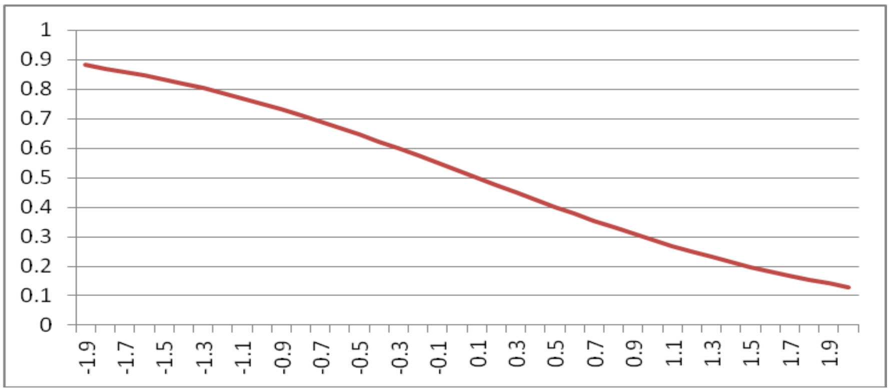

Figure 4-8: Sample Values of w_1 when Δ Ranges from -1.9 to 1.9

The same calculation is performed for RTT.

- RTT_{nL} denotes the RTT for the last downloaded segment from server *n*, whilst
- RTT_{nX} denotes the weighted average RTT over the X previous requests. RTT_{nX} is introduced to render the algorithm less sensitive to short lived fluctuations in the server connection RTT. It is calculated using equation (4.3.3.4).

$$RTT_{nX} = \frac{\sum_{i=1}^{X} w_i \cdot RTT_{ni}}{\sum_{i=1}^{X} w_i} \qquad w_i = \begin{cases} 1 & |t_c - t_d| \le \varepsilon \\ \dfrac{\varepsilon}{t_c - t_d} & |t_c - t_d| > \varepsilon \end{cases} \tag{4.3.3.4}$$

In equation (4.3.3.4), w_i is the weight factor for a RTT measurement RTT_{ni} ($1 \le i \le X$), and it reflects the freshness of the recorded throughput for server *n*. The variable t_c denotes current time and t_d denotes recorded time. The value of w_i ranges from 1 for RTT recorded within the past ε seconds, and it reduces as time passes.

Using previous logs to smooth the current server RTT estimate may slow reaction to significant increases in RTT, which could have a negative effect (e.g. giving a high rating to a server that is currently overloaded). Reaction time is controlled by a factor w_2 determined using an exponential function that produces values from 0 to 1 and is sensitive to changes in RTT. The factor w_2 is calculated using the formula given in equation (4.3.3.6). The normalised value of RTT – *normRTT* is calculated in (4.3.3.5) where *minRTT* is the smallest RTT recorded for any remote server.

$$normRTT_n = \frac{\min RTT}{(1-w_2) \cdot RTT_{nX} + w_2 \cdot RTT_{nL}} \qquad 0 < w_2 < 1 \tag{4.3.3.5}$$

$$\Delta = \frac{RTT_{nX} - RTT_{nL}}{RTT_{nX}} \qquad w_2 = \frac{1}{1+e^{\Delta}} \tag{4.3.3.6}$$

The Server Rating - R_n, is then calculated using equation (4.3.3.7).

$$R_n = (1-w) \cdot normRTT_n + w \cdot normTp_n \qquad 0 < w < 1 \tag{4.3.3.7}$$

Figure 4-9 illustrates the sequence diagram for the proposed performance-aware selection process.

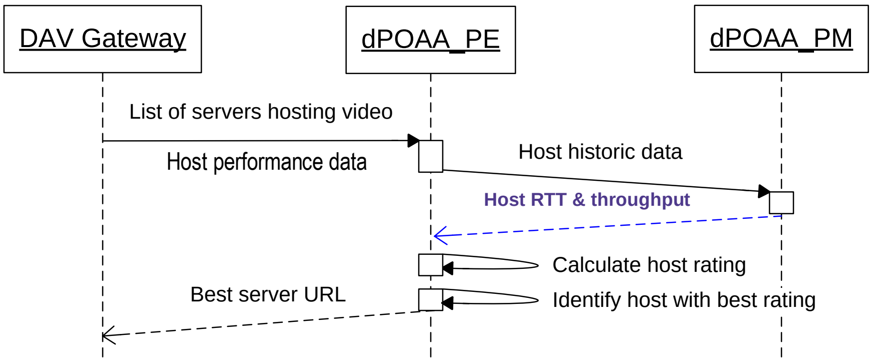

Figure 4-9: dPOAA Sequence Diagram

An outline of dPOAA PE selection algorithm is provided in Algorithm 4-2.

```
Input:     DER_URLj: List of servers (DERs) hosting video (vj)
           DER logs: Collated historic host performance data
Output:    DER_URLS: URL of the currently best connected host
Algorithm:
    for DERj ∈ DER_URLj
       Tpjx calculated based on (4.3.3.1)
       RTTjx calculated based on (4.3.3.4)
       w1 calculated based on (4.3.3.3)
       w2 calculated based on (4.3.3.6)
       normTpj calculated based on (4.3.3.2)
       normRTTj calculated based on (4.3.3.5)
       Rj calculated based on (4.3.3.7)
    endfor
    RS = max {DERj ∈ DER_URLj : Rj}
    return DER_URLS
```

Algorithm 4-2: The dPOAA Algorithm in Pseudo-code

4.3.4 Deployment

dPOAA is deployed as a Server Reputation Generator (see Section 4.4.3) to aid remote host selection in the proposed DAV architecture. In this case, dPOAA is installed on the campus gateway together with the DAV Gateway. Network conditions between the DAV Gateway and remote video servers are monitored to determine network performance without employing specialised software either at the remote server or on the client's viewing device. The block-level architecture for this deployment is presented in Figure 4-10.

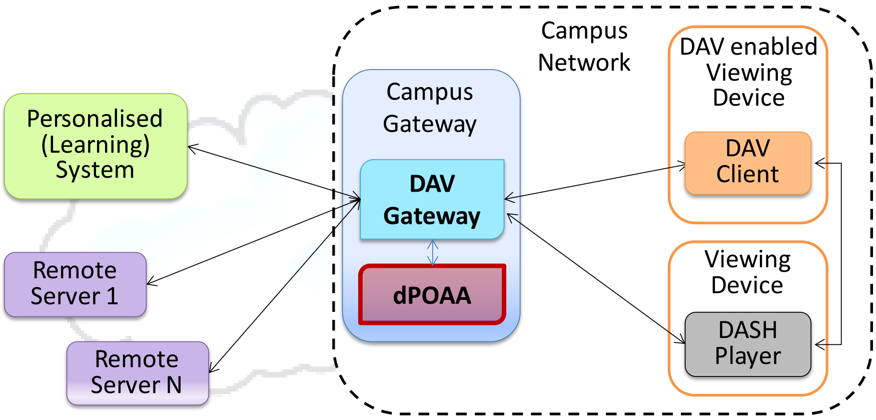

Figure 4-10: dPOAA Deployed with DAV Gateway

4.3.5 Future Work

In this work dPOAA was evaluated as a Server Reputation Generator to the DAV Gateway. However, when DAV is not deployed, dPOAA could be installed independently as a DASH player plug-in on the client DASH player allowing it to choose the best performing server which will in turn minimise download latency and rebuffering. In the latter case, network conditions between the client and remote servers hosting video content are monitored to determine network performance without employing any additional software module at the remote server. The download time is estimated from historic performance information gathered during recent sessions with servers. The gathered information is used to choose between the remote servers specified in the original MPD file. The MPD is not modified in this case. The block-level architecture for this deployment is shown in Figure 4-11, and it can be used when the DAV solution is not deployed.

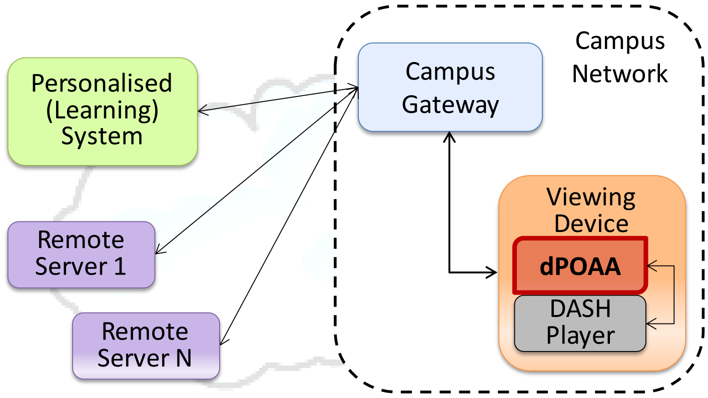

Figure 4-11: dPOAA Deployed at the Viewing Device

While the DAV deployment requires the dPOAA solution to be deployed in conjunction with DAV at the gateway level, it provides more recent information about the remote servers than the player deployment described here. The gateway deployment utilises server throughput data about segments requested from all DAV Clients in the network, whilst in the player deployment data is based on the download history of the client (where dPOAA is deployed).

Similar to approaches to service selection/ranking [254], dPOAA treats older observations as less relevant than more recent ones (weights observations based on their age). However, the prediction is based on the weighted averages and the periodicity of the time series (e.g. response times decrease during weekends or increase during class hours) is not considered. The proposed solutions could be extended to deploy models to identify trends and periodicity (seasonality) in the recorded readings, which would further enhance server selection.

4.3.6 Summary and Conclusions

This section describes the architecture and design of dPOAA for MPEG-DASH enabled PL systems where video content may reside on multiple servers. In this case the relevant MPD files contain multiple host URLs and the player can choose any of them. While the standard allows specification of multiple URLs, it does not specify corresponding selection algorithms. The hosting repositories differ in the quality of connection (throughput, delay, etc). dPOAA considers the quality of connection to rank remote servers. Remote host rating is inferred from historic performance information gathered by DAV Gateway over a number of recent sessions with servers. dPOAA selects the best performing server and the DAV Gateway updates the MPD accordingly before it is forwarded to the client (as presented in Sections 4.4.2.2, 4.4.2.4 and 4.4.3). Thus the client requests the video from the best performing server.

4.4 DASH-based performance oriented Adaptive Video distribution solution (DAV)

This section introduces the DASH-based performance oriented Adaptive Video distribution solution (DAV). DAV utilises the content already available locally to improve the content delivery process thereby improving the overall viewing experience. In this section, DAV is presented in terms of its deployment context, architecture, principal components and algorithms.

4.4.1 Context

Many current adaptive multimedia streaming solutions are HTTP-based [105]. Video servers can host multiple versions of a video of varying bitrate, resolution, colour depth and level of detail in order to cater for low- up to HD-quality renditions. The Dynamic Adaptive Streaming over HTTP (DASH) [12][13] standard supports video streaming based on successive downloads of short video segments and addresses the problems associated with traditional approaches to web streaming such as RTP/RTSP-based streaming (introduced in Section 2.2.1) and progressive download (outlined in Section 2.4.1). The DASH standard was reviewed in detail in Section 2.4 wherein a description of the MPEG-DASH Media Presentation Description (MPD) file format was presented. The proposed solution – DAV – dynamically generates new MPD files, combining information provided in the original video server's MPD file with network performance metrics collected over time. DAV also utilises segments downloaded by nodes in the campus network.

The newly created MPDs include both remote server URLs and generated URLs pointing to campus network nodes hosting relevant versions of video segments. The original MPDs provided by the remote host are used to identify remote servers hosting the video. Subsequently, a host rating algorithm is applied to determine the best performing remote and local hosts which are then incorporated into the generated MPDs. The structure of the latter fully complies with the standard, and all local DASH-enabled user devices benefit from the system. Local nodes, providing downloaded segments, act as DASH-based video servers where all that is required to serve DASH video is an off-the-shelf web server.

4.4.1.1 DASH Video Content Structure

The DASH standard and associated structures were presented and reviewed in Section 2.4. DASH video consists of a number of periods (temporal sections). Each period is associated with a number of adaptation sets (components/tracks), which in turn come in different representations. Representations typically differ in various aspects, in the case of video components they typically differ in terms of spatial resolution, video quality level, number of frames per second, etc. which is ultimately reflected in the bitrate. The MPD file lists the sets of available representations. Each representation is composed of segments. The standard defines the segment duration as "the duration of the media contained in the Segment when presented at normal speed" [12, p. 10], in other words, the number of seconds of video playback. We assume that all segments within one representation have the same duration (which is a typical setting indicated in the standard [12]) and that the MPD is structured with one segment per representation.

4.4.2 Architecture and Components

The high level DAV system architecture presented in Figure 4-12 is composed of diverse networked client devices which request video streams, a campus network to which users are connected, a campus gateway (at which level the DAV adaptive solution is deployed), distributed servers which store and serve DASH video content and the Internet which enables connectivity between servers and DAV. DAV operates in conjunction with a Personalised Learning (PL) system such as WHURLE 2.0 [255] that provides learner-specific information (e.g. WHURLE 2.0 UMS – User Modeling Service). DAV aims to enhance video delivery by performing DASH-based adaptive video delivery which selects the best performing source for the delivery of each segment. The source can be either one of the remote servers storing video segments belonging to the requested video or a client device located within the campus network. Only client devices that serve cached content must run DAV Client software to support this new functionality. Consequently, much of the required DAV Client functionality is provided by a standard web server. Simple user viewing devices will not serve content but will benefit from locally cached content.

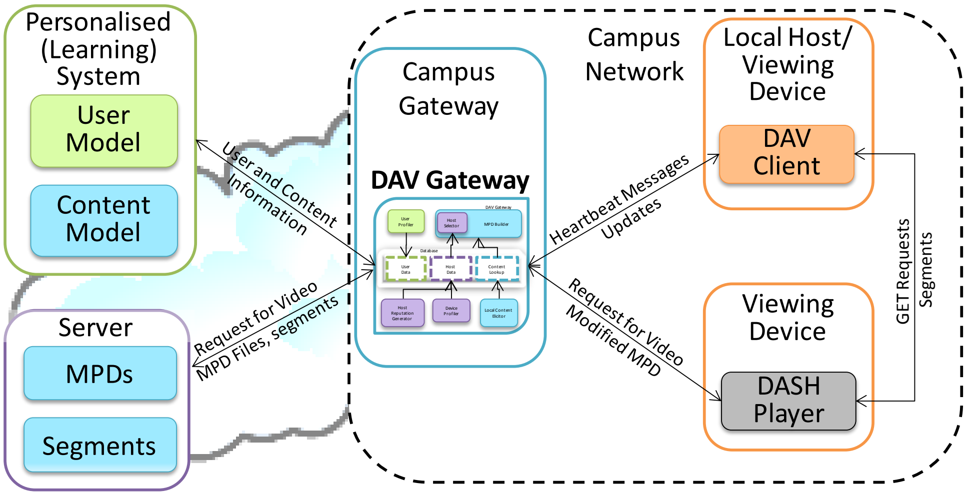

Figure 4-12: Component High-level Architecture and Components Communication

As DAV is located at the level of the campus gateway and deals with all requests emanating from campus network users for video-based learning content, it has access to a variety of information, including network-related data (performance characteristics related to the links connecting the remote servers and the campus network), video content-related data (e.g. information about segments available locally), and user context-related information (i.e. viewing device characteristics and user preferences provided by PL system). DAV calculates performance metrics from the observed video data flow, so no additional network traffic overhead is incurred. No measurement software is installed on the remote server side, nor does the monitoring impose additional load on the remote servers hosting the requested video.

The DAV Gateway architecture is expanded in Figure 4-13 and each of its major components is described in detail below.

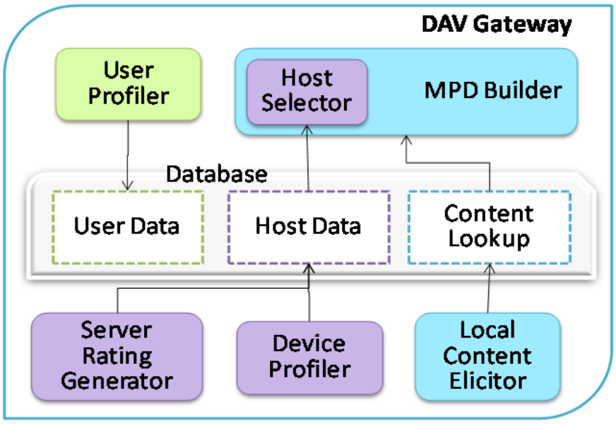

Figure 4-13: DAV Gateway Architecture

4.4.2.1 DAV Gateway: User and Device Profilers

Proposed profilers group users by their learning characteristics and viewing devices.

User Profiler. Group adaptation takes into account actions undertaken by users belonging to the same (manually or automatically created) group [194]. Here, partial group-based adaptation is performed. The User Profiler performs user clustering based on (a) the enrolling course and (b) the learner's profile as provided by the PL system. Each user cluster CUx, $x \in [0, M]$ (M is the maximum number of groups) groups users from the same course and sharing the same learning profile. PL systems maintain learner profiles in user model components, where user profiles are typically grouped into classes (e.g. stereotypes) based on their characteristics, such as learning style, goals, background knowledge, etc. Students having similar learning profiles require and are interested in similar learning content while students enrolled in the same course are more likely to require and consequently request the same video content around the same time. This node filtering reduces the number of local nodes that are considered for potential content retrieval and leads to more efficient decision making and ultimately increases average video quality across all clients.

Device Profiler. DAV identifies the capabilities of the requestor's viewing device (display resolution, supported media formats) using the Wireless Universal Resource File (WURFL) Device Description Repository [166]. An example of WURFL device information for the Samsung Galaxy Note 4[14] is given in Figure 4-14. This is a static approach to user device identification, where the user agent string embedded in the HTTP request is used to query a WURFL database to obtain the corresponding device capabilities. Alternatives for retrieving

[14] Extracted from http://sourceforge.net/projects/wurfl/files/WURFL/ [Accessed: 23-Dec-2014]

device profiles include W3C's Composite Capabilities/Preference Profile (CC/PP) [162], User Agent Profile (UAProf) [163] and Universal Plug and Play (UPnP) [165]. They are outlined in Section 2.6.1.

```xml
<device id="samsung_sm_n910a_ver1"
        user_agent="Mozilla/5.0 (Linux; Android 4.4.4; SAMSUNG-SM-N910A Build/KTU84P)
           AppleWebKit/537.36 (KHTML, like Gecko) Chrome/35.0.1916.141 Mobile Safari/537.36"
        fall_back="generic_android_ver4_4" actual_device_root="true">
    <group id="product_info">
      <capability name="uaprof" value="http://wap.samsungmobile.com/uaprof/SM-N910A.xml"/>
      <capability name="model_name" value="SM-N910A"/>
      <capability name="brand_name" value="Samsung"/>
      <capability name="marketing_name" value="Galaxy Note 4"/>
      <capability name="release_date" value="2014_september"/>
    </group>
    <group id="wml_ui">
    <group id="display">
      <capability name="physical_screen_height" value="127"/>
      <capability name="columns" value="25"/>
      <capability name="physical_screen_width" value="71"/>
      <capability name="rows" value="21"/>
      <capability name="resolution_width" value="1440"/>
      <capability name="resolution_height" value="2560"/>
    </group>
    <group id="image_format">
    <group id="wap_push">
    <group id="mms">
    <group id="sound_format">
    <group id="chips">
  </device>
```

Figure 4-14: Samsung Galaxy Note 4 WURFL Information

4.4.2.2 DAV Gateway: Server Rating Generator

The Server Rating Generator operates under the assumption that at any point in time, a number of videos (learning objects) with the same learning objectives exist and are stored on distributed remote servers. These hosting repositories differ in response time and availability (varying performance, etc.) as well as in quality of connection (e.g. throughput). This rating module generates for each remote server a rating R_n which is inferred from historic network performance data gathered over a number of recent sessions (segment downloads) with those servers. The DASH-aware Performance Oriented Adaptation Agent (dPOAA) [15] (described in Section 4.3) can be deployed in this setting as a module that will provide server rating information to the DAV Gateway.

4.4.2.3 DAV Gateway: Local Content Elicitor

The gateway-located Local Content Elicitor is aware of campus network nodes that have recently downloaded video segments. This component collects information about segment requests in terms of the request time, requestor (local node ID), contents (video and segment IDs) and destination server (remote host URL). This data together with the segment quality (bitrate) and the time of segment download (timestamp) are stored in the Content Lookup database. The Local Content Elicitor communicates with DAV Client nodes that recently downloaded or are currently downloading segments of the requested video to determine their availability. This is achieved with the deployment of the Heartbeat Mechanism described in Section 4.4.3.3.

Whilst DAV facilitates sharing content among peers, it is not a classical peer-to-peer system [256] and peers are not aware of each other in the sense that there is no distributed hash table, nor do nodes implement gossip/flood mechanisms. Here, information about local content is centralised and communicated to peer nodes via MPDs.

4.4.2.4 DAV Gateway: Media Presentation Description (MPD) Builder

The MPD Builder module creates a new MPD for the requested video based on the original MPD documents provided by the remote servers and on the host performance information. The MPEG-DASH BaseURL element is described in detail here, whilst the general structure of DASH MPD documents was covered in Section 2.4.4. The BaseURL is a component of MPD syntax which specifies a location (URL) where DASH content resides. Critically for DAV, this optional element can be specified at multiple levels in the MPD XML hierarchy. The MPDs composed by DAV contain BaseURLs specified at multiple levels. When all segments are available on multiple servers (i.e. at alternative locations), the relevant URLs are specified with multiple BaseURL elements. A sample MPD file containing multiple BaseURLs at the representation level is presented in Figure 4-15.

```xml
<MPD xmlns="urn:mpeg:DASH:schema:MPD:2011">
    <Period start="PT0S">
      <AdaptationSet bitstreamSwitching="true">
        <Representation id="0" codecs="avc1" mimeType="video/mp4" width="320"
           height="240" startWithSAP="1" bandwidth="45514">
           <BaseURL>http://www-itec.uni-klu.ac.at/Bunny/bunny_6s/</BaseURL>
           <BaseURL>http://download.tsi.telecom-paristech.fr/Bunny/bunny_6s/</BaseURL>
           <SegmentList duration="6">
              <SegmentURL media="bunny_6s_50kbit/bunny_6s1.m4s"/>
              <SegmentURL media="bunny_6s_50kbit/bunny_6s2.m4s"/>
              <SegmentURL media="bunny_6s_50kbit/bunny_6s3.m4s"/>
           </SegmentList>
        </Representation>
      </AdaptationSet>
   </Period>
</MPD>
```

Figure 4-15: Section of a Sample MPD with Multiple BaseURLs at the Representation Level

The standard allows multiple BaseURLs at the same level, but does not dictate metrics to guide DASH clients on how to choose a server, as there is no priority, or preference between the URL alternatives indicated in the MPD. The alternative servers may be used as fallback when one server becomes unavailable or too slow [124]. The order of URL specification is not relevant.

Figure 4-16 shows a sample MPD document containing multiple BaseURL elements specified at different levels. Three consecutive dots (...) are used to indicate that some attributes/elements have been omitted for brevity. A relative URL is specified at the representation level (indicated by the red box) and an absolute URL is specified at the MPD level (surrounded with the blue box). In this case, the URLs from different levels are merged using a resolution algorithm [257].

The relative URLs are concatenated with the BaseURL specified at the level above, and so on until an absolute URL is built or the BaseURL at the MPD level is reached. The BaseURL element has two optional attributes (a) byteRange (a template to construct URLs for requesting a byte range of the segments) and (b) serviceLocation (specifies a relationship between Base URLs, e.g. a common CDN).

```
<MPD xmlns="urn:mpeg:DASH:schema:MPD:2011" .. >
  <BaseURL>http://www-itec.uni-klu.ac.at/</BaseURL>
  <Period ...>
    <AdaptationSet ...>
      <Representation id="0" codecs="avc1" mimeType="video/mp4" ... >
        <BaseURL>Bunny/bunny_6s/</BaseURL>
        <SegmentList duration="6">
          <SegmentURL media="bunny_6s_50kbit/bunny_6s1.m4s"/>
          <SegmentURL media="bunny_6s_50kbit/bunny_6s2.m4s"/>
          <SegmentURL media="bunny_6s_50kbit/bunny_6s3.m4s"/>
            ...
        </SegmentList>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```

Figure 4-16: Sample DASH MPD with Multiple BaseURL Elements at Different Levels

When video content is available at multiple remote hosts (specified with multiple MPD level BaseURLs), the best performing server is chosen by the DAV Gateway Host Selector. The Host Selector uses the remote hosts indicated by the *Server Rating Generator* (such as dPOAA) which selects the currently best performing host and sets the host's URL as the BaseURL at the MPD level. For each segment within a video, the MPD Builder takes the N best performing local hosts (based on the information provided by the Device Profiler and Local Content Elicitor) and adds the local node URLs as Representation level BaseURLs to the new MPD file. Multiple BaseURLs are provided as fallback, if one host fails the others are consulted.

The new MPD file structures are compatible with the DASH standard [12] [117] and specify one segment per period, providing different representations for the segment and specifying the URLs of the hosting nodes. When a segment is unavailable locally the client reverts to the remote server (URL is specified in the BaseURL at MPD level).

The DAV Gateway produces two types of MPDs, static and dynamic. Static MPDs are typically used for on demand streaming of prerecorded video. In this case, all segments are available at the time specified in the MPD@availabilityStartTime attribute (if not specified, at the time the MPD becomes available). Dynamic MPDs are used for live video content, such as lectures or speeches, which is commonly streamed in real time as the recorded event takes place. A live feed, such as video is encoded and the resulting stream is published on a web server, which then serves the live stream to clients. In this case, media content can only be prepared for transmission after the content has been recorded and encoded. Typically, a DASH client fetches a dynamic MPD from a server to join a live session. Consequently, for each selected representation, the client determines the latest segment availability time and the segment availability start time of the next segment. Furthermore, the client determines the segment playout start time and when to fetch an updated MPD.

An indicative extract from a DAV-generated static MPD (static DAV MPD) is given in Figure 4-17.

```
<MPD xmlns="urn:mpeg:DASH:schema:MPD:2011" ... >
   <BaseURL>http://www.dcu.ie/Video1/Vid1_6s/</BaseURL>
   <Period id="0" ...>
      <AdaptationSet ...>
         <Representation id="v0"... bandwidth="50012" >
            <BaseURL>http://LocalNodeA_IP/Video1/Vid1_6s/</BaseURL>
            <BaseURL>http://LocalNodeX_IP/Video1/Vid1_6s/</BaseURL>
            <SegmentList duration="6">
               <SegmentURL media="v_50kbit/segment01.m4s"/>
            </SegmentList>
         </Representation>
         <Representation id="v1" ... bandwidth="100012">
            <BaseURL>http://LocalNodeB_IP/Video1/Vid1_6s/</BaseURL>
            <BaseURL>http://LocalNodeY_IP/Video1/Vid1_6s/</BaseURL>
            <SegmentList duration="6">
               <SegmentURL media="v_100kbit/segment01.m4s"/>
            </SegmentList>
         </Representation>
               ...
      </AdaptationSet>
      ...
   </Period>
   <Period id="1" ...>
      <AdaptationSet ...>
         <Representation id="v0"... bandwidth="50012" >
            <BaseURL>http://LocalNodeA_IP/Video1/Vid1_6s/</BaseURL>
            <BaseURL>http://LocalNodeX_IP/Video1/Vid1_6s/</BaseURL>
            <SegmentList duration="6">
               <SegmentURL media="v_50kbit/segment01.m4s"/>
            </SegmentList>
         </Representation>
         <Representation id="v1" ... bandwidth="100012">
            <BaseURL>http://LocalNodeB_IP/Video1/Vid1_6s/</BaseURL>
            <BaseURL>http://LocalNodeY_IP/Video1/Vid1_6s/</BaseURL>
            <SegmentList duration="6">
               <SegmentURL media="v_100kbit/segment01.m4s"/>
            </SegmentList>
         </Representation>
               ...
      </AdaptationSet>
      ...
   </Period>
   ...
</MPD>
```

Figure 4-17: Indicative Elements of the Static DAV MPD

A sample of dynamic DAV-produced MPD file is given in Figure 4-18 (page 88).

While there are similarities between dynamic and static DAV MPDs, there are differences in the number of periods specified, identification and timing attributes.

```xml
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  type="dynamic"
  minimumUpdatePeriod="PT9S"
  timeShiftBufferDepth="PT30M"
  minBufferTime="PT4S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011"
  publishTime="2014-12-17T12:30:00"
  availabilityStartTime="2014-12-17T12:30:00"
>
   <Location>http://DAV_Gateway_IP/Video1/upMPD_Vid1_10s.mpd</Location>
   <BaseURL>http://www.dcu.ie/Video1/Vid1_10s/</BaseURL>
   <Period id="0" start="PT0S"...>
     <AdaptationSet ...>
       <Representation id="v0"... bandwidth="50012" >
         <BaseURL>http://LocalNodeA_IP/Video1/Vid1_10s/</BaseURL>
         <BaseURL>http://LocalNodeX_IP/Video1/Vid1_10s/</BaseURL>
         <SegmentList duration="10">
           <SegmentURL media="v_50kbit/segment01.m4s"/>
         </SegmentList>
       </Representation>
       <Representation id="v1" ... bandwidth="100012">
         <BaseURL>http://LocalNodeB_IP/Video1/Vid1_10s/</BaseURL>
         <BaseURL>http://LocalNodeY_IP/Video1/Vid1_10s/</BaseURL>
         <SegmentList duration="10">
           <SegmentURL media="v_100kbit/segment01.m4s"/>
         </SegmentList>
       </Representation>
                    ...
     </AdaptationSet>
   </Period>
 </MPD>
```

Figure 4-18: Indicative Elements of a sample Dynamic DAV MPD

Dynamic DAV MPDs are similar in structure to static DAV MPDs. However, while static DAV MPDs (excerpt provided in Figure 4-17) contain information for all periods of a presentation, dynamic DAV MPDs define one period at a time as shown in Figure 4-18. The specified period contains an adaptation set with various representations of a single segment. MPDs are extended for one period in each MPD update.

The MPEG-DASH standard specifies attributes that are used for live content playout timing. Table 4-7 (page 89) lists a subset of these attributes. The content of this table is drawn from various sections of the MPEG-DASH standard [13].

A video content host may advertise times (in wall-clock time) at which segments of media content will be available. Synchronisation methods by which client devices synchronise their local clocks with wall clock times are also advertised by the source server. For example, the Network Time Protocol (NTP) [258] or HTTP Date header [20] (date and time that the message was sent) can be used as synchronisation mechanisms. Alternatively, synchronisation information can be provided in the MPD. For example, an MPD may specify values for MPD@availableStartTime and Period@start attributes. For dynamic MPDs the sum of these two values and the duration of the media segment may specify the availability time of the period (the first media segment of each representation) in UTC [259]. The DASH player requires MPD updates in order to continue playing video. The time between two updates is specified by the MPD@minimumUpdatePeriod attribute value. A Location element, a child of MPD element, specifies the location (URL) at which the updated MPD is available.

Attribute	**Description**	**Level**
availabilityStartTime	The anchor for the computation of the earliest availability time (in UTC) for any Segment in the Media Presentation.	MPD
publishTime	The wall-clock time when the MPD was generated and published at the origin server. MPDs with a later value of @publishTime shall be an update to MPDs with earlier @publishTime.	MPD
minimumUpdatePeriod	The minimum update period of the MPD.	MPD
suggestedPresentation Delay	Suggested presentation delay as delta to segment availability start time. A fixed delay offset in time from the presentation time of each access unit that is suggested to be used for presentation of each access unit. The client chooses a suitable value when this parameter is not specified.	MPD
minBufferTime	Minimum buffer time, used in conjunction with the @bandwidth attribute of each representation.	MPD
timeShiftBufferDepth	The duration of the time shifting buffer for any Representation in the MPD that is guaranteed to be available for a Media Presentation.	MPD
start	The start time of the Period relative to the MPD availability start time.	Period

Table 4-7: MPD Timing Attribute, Attribute Description and Level from [13]

In terms of frequency of dynamic MPD production, the DAV Gateway behaves similarly to HLS [100]. In HLS, the client, after joining a live session fetches a new MPD after each segment. However, HLS does not provide information on the exact time schedule of their MPD and media segment creation [259]. Furthermore, timing and synchronisation in the DAV case are simplified, as the duration of the video and the time of the next MPD update are known. A typical live streaming server provides live content regardless of when different clients join the live session; each sees the same point in the stream at the same time. The DAV Gateway provides content where the clients request content from the point of their interest (typically the beginning of the video clip).

A further difference between static and dynamic MPDs is in the use of MPD identification attributes. The MPEG-DASH standard specifies attributes that are used for MPD components identification in the updated MPDs. Table 4-8 (page 90) lists attributes and descriptions for a subset of these attributes in the context of dynamic MPDs. The content of this table is drawn from the MPEG-DASH standard [13, p. 61]).

	Attribute Description	Level
id	An identifier for the MPD (recommended to use an identifier that is unique within the scope in which the Media Presentation is published, e.g. the URL to the MPD); If present, MPD@id shall be the same in the original and the updated MPD.	MPD
id	An identifier for this Period. The identifier shall be unique within the scope of the Media Presentation. This attribute shall be present in case of dynamic MPDs. The values of any Period@id attributes shall be the same in the original and in the updated MPD, unless the containing Period element has been removed.	Period
id	The values of any AdaptationSet@id attributes shall be the same in the original and in the updated MPD unless the containing Period element has been removed.	AdaptationSet
id	Any Representation with the same @id and within the same Period as a Representation appearing in the previous MPD shall provide functionally equivalent attributes and elements, and shall provide functionally identical Segments with the same indices in the corresponding Representation in the new MPD.	Representation

Table 4-8: MPD Component Identification Attribute, Attribute Description and Level from [13]

The MPEG-DASH standard provides template-based Segment URL construction which is used to indicate the location of following segments. This MPD format reduces the size of MPD file. However, this approach cannot be used in the DAV setting, as the DAV Gateway is not aware of the location of a segment in advance.

Where a typical DASH player downloads a segment from a local node and subsequently places a request to a remote server based on its most recent throughput estimate, an unrealistically high bitrate may be requested. The player's throughput estimate is, in this case, of the local network throughput which is typically very high (significantly higher than the throughput of the link to the remote server). Consequently, the player may over-optimistically request a segment of unrealistically high bitrate assuming an uncongested connection to the remote server. This phenomenon is also known as a "proxy effect" [260]. To alleviate this potential problem, the DAV Gateway MPD builder introduces a bitrate ceiling based on the current estimated throughput to the chosen remote server. In this case the player may not request content from the server at a bitrate higher than this ceiling. This ceiling is implemented by limiting the representation options in the new MPD file for segment versions (bitrates) not available locally.

DAV utilises the DASH live streaming functionality in a novel fashion to provide requesting nodes with access to the most recently downloaded segments by DAV Clients. In a static MPD setting a fixed MPD is provided at video request time, and segments subsequently downloaded to other local nodes during the course of video playout remain inaccessible. Therefore, the DAV Gateway also provides clients with dynamic MPDs. In this case, a new MPD is provided for each period (containing a segment). New locally available content downloaded between segment requests is reflected in updated MPDs. In effect every video viewed over DAV is treated as if it were a live stream in order to force clients to request regular MPD updates. Each MPD update takes into account the latest locally cached content. The client retrieves the latest MPD, analyses the playlist and, if needed, it can access the segments downloaded since its initial request. The client device requests the next segment and plays out the current segment under the expectation that it can continually access the next segment in time. Before fetching a new segment, the client device requests a new MPD providing the location of the subsequent segment. This process may impose some delay as it requires at least one MPD fetch round-trip time. However, the DAV-generated MPD is provided within the campus network and therefore this delay will be negligible. Furthermore, the updated MPD may specify URLs to locally available segments and, in this case, the overall segment download delay is significantly reduced.

4.4.2.5 DAV Gateway: Databases

DAV Databases store all data necessary for the MPD Builder. Some information originates from the external personalised system (i.e. PL system), while the remainder is inferred or collected by DAV components. The PL system informs which video matches the current learner request (e.g. MPD URLs on the external servers hosting the video based on the PL system domain/content model) and learner profile details (required for user-based clustering). DAV does not maintain detailed information about user profiles, as these evolve over time (e.g. when a user learns a new topic). Instead, DAV proposes to make use of sophisticated user models already implemented within current PL systems and stores only information necessary for user clustering in the ***User Data database***. Alternatively, DAV could independently maintain user information but this approach increases computational complexity and adaptation time.

The ***Host Data database*** stores information about the local active nodes including well-resourced nodes. This information is provided by the *DAV Gateway Device Profiler* and through the Heartbeat Mechanism (Section 4.4.3.3). Remote server ratings as inferred by *Server Rating Generator* (e.g. dPOAA) are also stored here.

The ***Content Lookup database*** contains information regarding video segments stored on local nodes. This information is collected and maintained by the *Local Content Elicitor*.

4.4.2.6 DAV Client

The ***DAV Client*** side module (a) accepts requests and sends the requested content to other nodes in the network, (b) reports information to the *DAV Gateway* (periodically with the Heartbeat Mechanism and upon downloading segments, e.g. upon receiving a segment from another DAV Client) and (c) plays video content.

Much of the required ***DAV Client*** functionality is provided by a simple web server which accepts HTTP requests from other local nodes and sends requested content (video segments) to the requester. (Running a web server requires port 80 be open. Since DAV clients are locally administered, configuring their firewall to allow traffic on port 80 is not an issue.) There are a number of software libraries that allow running a web server as part of another application.

GNU libmicrohttpd[15] is one such library implemented in C. It is HTTP 1.1 compliant and supported on multiple platforms (Linux, Android, OS X, W32, etc.). Alternatively, a light-weight standalone web server can be used, such as Mongoose[16].

The DAV Client indexes all downloaded segments and stores them (for β seconds) to a predefined directory structure accessible to the local web server. The *DAV Client* keeps the *DAV Gateway* informed about its availability via a Heartbeat message.

The DAV Client player is made aware of local and remote content locations through the new MPD, and locally available segments are utilised when available. The DASH player selects the bitrate of the next requested segment (segment version) using the bandwidth estimation formula given in equation (2.4.8.2). The player then parses the MPD file, to find the URL for the matching representation (containing that segment version). The new MPD contains local URLs for segment versions available locally, if a selected segment version is not found locally, the remote server URL is used.

Those users interacting via personal devices (without a DAV Client installed) connect with the DAV Gateway when requesting learning content from the PL system. They receive a version of the relevant MPD file that depends on the time of their request. However, their cached content cannot be accessed (referenced from new MPDs) by other nodes without installing full DAV Client functionality.

4.4.3 Host Selection, MPD Building and Heartbeat Algorithms

The sequence of a typical learner login request is illustrated in Figure 4-19. DAV enhances user interaction with a PL system, by intercepting requests to the PL system.

(1) During user login, DAV (Device Profiler) identifies the user's device capabilities and connection type and forwards the request to the PL system. The PL system provides information on the user's enrolled course and profile details (maintained by the PL system User Model). This information is used for user-based clustering.

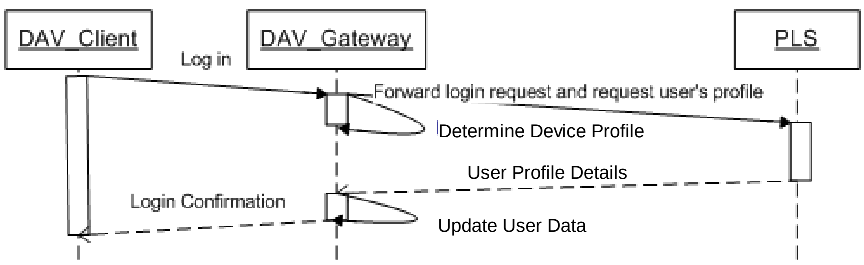

Figure 4-19: Log-in Sequence Diagram

(2) On the initial content request, the PL system provides information about the content i.e. relevant and suitable video based on: student learning objective (relevancy), enrolled course and

---

[15] http://www.gnu.org/software/libmicrohttpd/ [Accessed: 2-Jan-2016]

[16] http://cesanta.com/mongoose.shtml [Accessed: 2-Jan-2016]

profile (suitability). All videos recommended by the PL system may reside on a number of remote servers and this is reflected in the list of MPD URLs provided by the PL system (Domain Model). Once the *DAV Gateway* receives the list of MPD URLs for the recommended video, it obtains the relevant MPDs from the hosting servers.

For initial requests (before any content has been locally cached) from the campus network for a particular video, the original remote server-supplied MPDs are forwarded to the requesting nodes, and the content is requested from the remote servers. For subsequent requests from the campus network for the same video, the *DAV Gateway* (specifically the *MPD Builder* component) constructs a new MPD document using the best performing local and remote nodes (chosen by the Host Selector, see Section 4.4.3.1). The new MPD is then forwarded to the user requesting the video. Consequently, the user will transparently request video segments based on the provided MPD (using local content when available).

Two variants of DAV are proposed, static DAV creates one new MPD which is forwarded to the client as indicated in Figure 4-20; dynamic DAV creates a number of MPDs that are dynamically updated and forwarded on client request as indicated in Figure 4-21. The dynamic MPDs may be updated during the video playout which leads to better utilisation of local content downloaded during the video playout.

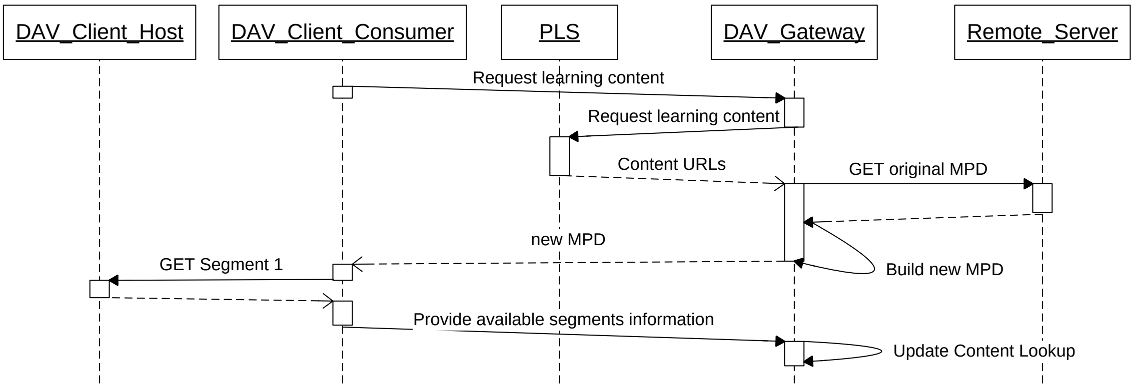

Figure 4-20: DAV (static MPD) Content Request Sequence Diagram

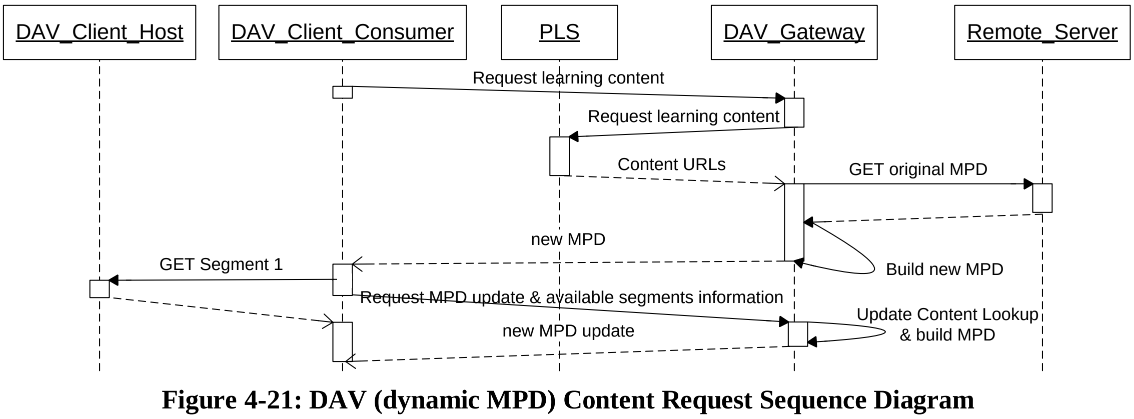

Figure 4-21: DAV (dynamic MPD) Content Request Sequence Diagram

Static MPDs, typically used for video-on-demand applications, are valid for the whole presentation and are there is no need for updates. Alternatively, dynamic MPDs are typically intended for live presentations but are used in the DAV context to force regular MPD update requests by clients. The adoption of dynamic MPDs was chosen in this setting, as the standard supports the update of dynamic MPDs during the presentation playout and the clients periodically request updated dynamic MPDs. It is our novel application of dynamic MPDs that makes local content accessible to all viewing devices.

4.4.3.1 DAV Host Selection

MPD Builder dynamically composes MPD constituents, selecting video content from the hosts that are currently the most efficient providers of the requested segments. These providers can be local nodes or remote servers and are determined by the *Host Selector*.

Remote Hosts

When requested video resides on multiple remote servers (i.e. there are multiple BaseURLs specified in the original MPD), the DAV Host Selector selects the server based on the server recommendation generated by the Server Rating Generator component (e.g. dPOAA).

Local Hosts

The Host Selector uses a utility function-based approach to calculate scores for local hosts - L_{ij} of equation (4.4.3.1.1) for the given segment version *i* stored at the node *j*. L_{ij} is a product of the normalised form (values ranging from 0 to 1) of each parameter and it has values in the [0,1] interval and no unit. Each video segment is available in a range of bitrates (as indicated in the original MPD). Here, "segment version *i*" refers to a given segment in one of the supported bitrates.

$$L_{ij} = nC_j \cdot fC_{ij} \tag{4.4.3.1.1}$$

$$nC_j = \begin{cases} 0 & n > dL_j \\ \dfrac{1}{n} & n \le dL_j \\ 1 & n = 0 \end{cases} \tag{4.4.3.1.2}$$

In equations (4.4.3.1.1 and 4.4.3.1.2), nC_j is the utility function component for node j ($1 \le j \le M$), and reflects the number of recent segment download requests to node *j*. The value of *n*, the segment request rate, is included in the Heartbeat updates from node *j*. The request rate depends both on the type of device and on the device connection type. Device class is determined based on the device profile and related to the overall processing power of the device and the connection types (wired vs. wireless). This device classification builds on the three classes proposed in [173]: i.e. *Large Screen*, *Portable* and *Handheld*. Two connection types considered in device classification are wired and wireless. The two attributes are combined into five device classes: $dC_j \in$ {*Handheld–Wireless*, *Portable–Wireless*, *Portable–Wired*, *LargeScreen–Wireless*, *LargeScreen–Wired*} and a maximum request rate (device Level - dL) is assigned to each class, $dL_j = f(dC_j)$ as indicated in Table 4-9. The aim is to evenly spread the delivery of downloaded segments among the DAV Clients. Network connection is considered as the limiting factor, so devices with a wired connection are assigned a higher dL value.

Device Class - dC	Device Level - dL
Handheld–Wireless	0
Portable–Wireless	0
Portable–Wired	20
LargeScreen–Wireless	10
LargeScreen–Wired	30

Table 4-9: Device Class and Device Level (Maximum Request Rate)

$$fC_{ij} = \begin{cases} 1 & |t_c - t_d| \le \varepsilon \\ \dfrac{\varepsilon}{t_c - t_d} & \varepsilon < |t_c - t_d| \le \beta \\ 0 & |t_c - t_d| > \beta \end{cases} \qquad (4.4.3.1.3)$$

fC_{ij} is a utility function component that considers the freshness of the given segment version *i* stored at the node *j*. The value of fC_{ij} ranges from 1 for nodes downloading the segment within the past ε seconds, and it reduces as time passes as indicated in (4.4.3.1.3). DAV Clients keep the downloaded segments for a predefined period of β seconds (e.g. video duration). Segments older than β seconds are considered stale and not referenced in the newly created MPDs.

L_{ij} is obtained over all candidate local hosts (within a user-based cluster) storing video segment version *i*. The host with the highest rating is selected as the target host and inserted in the new MPD. The node URL is placed within the BaseURL element at representation level (as in Section 4.4.3.2).

4.4.3.2 MPD Building

The best rated remote server is used in the new MPD. The local hosts with the highest scores are used for MPD composition, as given in Algorithm 4-3.

```
Input:  Original MPD file and collated host statistics
Output: New MPD file
1. If multiple remote servers exist, use the best performing one according to
   Server Rating Generator in order to set the BaseURL element at MPD level
2. For each segment in the video sequence
    For each bitrate supported in the original MPD
      Select best performing local host within the user cluster according
      to equation (4.4.3.1.1)
      If not available locally specify remote server imposing bitrate
           ceiling based on the server throughput
      Add the chosen host URL to the BaseURL at Representation level.
```

Algorithm 4-3: The MPD Generation Algorithm Outline

4.4.3.3 Heartbeat Mechanism

Lightweight Heartbeat messages are used to verify which campus nodes hosting video segments are currently online and available. The Heartbeat Mechanism is based on a periodic message transmitted between a DAV Client and the DAV Gateway. This message transmits the current state of the device and the request rate. The transmission frequency of the Heartbeat message can be adjusted at the DAV Client. The Client sends an initial Heartbeat message to the DAV Gateway when it comes online, on shutdown and periodically every Heartbeat period.

A Heartbeat Mechanism is also employed by the Stream Control Transmission Protocol (SCTP) [46]. In SCTP, a Heartbeat Consumer sends Heartbeat Requests to monitor Heartbeat Producers and expects an acknowledgement within a specified timeframe – the Retransmission Timeout (RTO). Unacknowledged requests cause the error count for the corresponding Producer to be incremented. The Producer is considered as inactive when the value in the error counter exceeds HeartMaxRetrans. The error count is cleared when the Producer next contacts the Consumer. This process is simplified in DAV.

***DAV Client*.** While online, DAV Clients periodically (when Heartbeat period expires) send Heartbeat messages to the associated DAV Gateway. The latter messages contain information about the number of requests handled since the previous Heartbeat message was sent (request rate). The request rate is used for load balancing among clients. This process is illustrated in Figure 4-22.

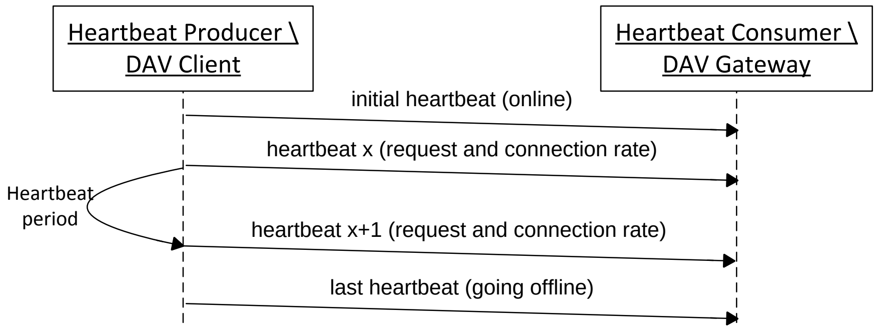

Figure 4-22: Heartbeat Sequence

***DAV Gateway*.** The reception of the initial Heartbeat message indicates that a DAV Client has been registered on the network. The DAV Gateway assumes a DAV Client is offline when N Heartbeat periods lapse without a Heartbeat message having been received from the Client.

This lightweight option minimises additional network traffic. However, this mechanism could be extended to report content freshness and network performance when downloading from or streaming to other clients in line with the Heartbeat messaging utilisation proposed in [261].

4.4.4 Deployment

DAV is a DASH-based solution that can be deployed in a campus setting as depicted in Figure 4-12 (page 82).

This solution scales well as the nodes in the campus network are grouped into clusters based on learners' properties. The similarity factor used for clustering in this case is provided by the associated PL system. The PL system provides information about each learner's enrolled course and learning preferences. The PL system's User Models maintain learner-related information, typically grouping learners with similar properties. The same mechanism is adopted in DPEA (DAV component), as the PL system recommends similar/identical video content for students enrolled in the same courses and having similar profiles. Furthermore, all devices are grouped based on their hardware characteristics (e.g. screen resolution) and type of network connection. Local devices connected to the wired network are favoured over wirelessly connected devices for DAV Client hosting purposes as indicated in Table 4-9.

The MPD building process requires selecting local nodes hosting required video segment versions. The local hosts (DAV Clients) are selected based on a utility function. It could be argued that building MPDs on-the-fly requires time and further extends the initial (startup) delay. However, this delay is offset by the improved user QoE that comes with access to local content.

4.4.5 Future Work

While the current solution utilises content available locally without a strategy for prefetching segments, it could be extended to request segments of higher quality and store them on well-resourced local nodes. Furthermore, a "Request Prioritisation" strategy could be put in place to prioritise local content sharing by reserving bandwidth for requests issued by well-resourced nodes hosting DAV Clients, so that they would download segments of a higher bitrate. These segment versions will then be used by other nodes, thereby improving the viewing quality for a larger number of users.

Finally, users could be incentivised to install DAV Client software on their machines. For example, a "quid pro quo" strategy could be deployed where access to local content could be limited to users that are providing access to their own local content.

4.4.6 Summary and Conclusions

This section describes the architecture and components of DAV for MPEG-DASH enabled PL systems where the overall viewing experience is enhanced by recruiting groups of active (watching) users within the campus to share their partial copies of the video stream with other nodes in the campus network. The adaptation process in DAV considers available bandwidth information collated over a number of the most recent segment downloads, and locally available content within the campus network to achieve the highest quality of video that can be delivered over the current network conditions. It should be noted, that this solution requires no modification of the HTTP servers hosting video content and that both DAV Client-enabled devices and devices with a simple DASH player (i.e. without a DAV Client installed) benefit from the proposed approach.

4.5 Solution Overview

The proposed context-aware solutions are light-weight as the underlining algorithms are deterministic and a simple context model has been applied. The solutions deploy end-to-end measurements in a non-intrusive manner without probing (no overhead traffic) or requesting information from the hosting server. The metrics for server-client link performance are measured RTT and throughput calculated based on the downloaded content size and measured time, as these two parameters are often used as a direct indicator of network performance (see Section 2.4.8).

A key-value model is used to represent delivery network context information, so the oPOAA PM and CM, dPOAA PM, DAV databases are represented as lists of attributes with their corresponding values. While this approach has deficiencies [221], it is sufficient to model a number of context types (e.g. [162]).

The proposed solutions use information about the past state of its context entities (i.e. RTT and throughput). Management of historical information imposes challenges and summarisation techniques (e.g. historical synopsis of data) must be utilised when the number of updates is high [221]. Therefore, the proposed solutions use time-windowing to reduce the amount of data saved at any given time. A context-aware system should also express information about quality policies such as confidence, freshness or resolution of captured data [262], therefore dPOAA and DAV consider freshness of the collected performance data based on timestamps. A sliding window discards data that are older than a given age. It is straightforward to implement, yet is an effective technique that keeps the selection in sync with the currently observed network parameters. Shorter windows lead to faster reaction to the changes in network conditions. However, short sliding windows may result in oscillations. So a weighted sliding window approach was adopted to react faster to more recent observations, while still considering historic observations albeit with smaller weight.

In order to evaluate the quality of a link to a server a utility function is used. The function maps the quality vector (e.g. Qs={normRTT, normTp}) to a single real value, to enable server ranking. A Simple Additive Weighting, one of the Multiple Attribute Decision Making approaches, is used in the utility functions. The QoS attribute values are normalised (transformed into a value between 0 and 1) to allow a uniform measurement independent of the units and ranges and finally, the weighting process is applied.

One key requirement of the proposed solution is that the analysis and the prediction algorithms must be executed with minimal computational overhead. The solutions proposed here may lack the learning capabilities and the accuracy of off-line models (which typically require considerable time/computing power), but they achieve satisfactory (not necessarily optimal) predictions efficiently.

Solutions, limitations and future work are described in Sections 6.4 and 6.6, respectively.

4.6 Summary

The ongoing diversification of video viewing devices and growing network connectivity are increasingly addressed by making use of adaptive HTTP streaming technologies (such as MPEG-DASH). Communities of video consumers that have similar demands (e.g. for the same video) and that are in close geographical proximity (e.g. campus network), may impose needless demands on the video hosting servers and on the communication link between the servers and the campus network. In this case, identical/similar video content is requested and delivered multiple times to local nodes. In this setting, the proposed DASH-based Performance Enhancement Architecture (DPEA) will enhance the overall viewing experience by selecting the best performing remote servers and utilising content available in the campus network. The DPEA architecture is presented and all components are described in detail in this chapter. This chapter details the Performance Oriented Adaptation Agents (POAA) which enhance selection of Learning Objects residing on multiple remote servers for Open Personalised Learning systems. Open POAA (oPOAA) deals with a variety of media types transported over UDP, adding network performance-aware adaptation to PL systems dealing with open corpus content. DASH-aware POAA (dPOAA) performs selection across remote servers storing identical MPEG-DASH content. Furthermore, the DASH-based performance oriented Adaptive Video distribution solution (DAV) utilises DASH content already available locally to improve the video delivery process and is one of the DPEA components presented here. Access to locally available content is achieved through DAV-generated static MPDs. This idea is however further extended through a novel application of dynamic MPDs whereby static video content is treated by DAV as if it were a live stream in order to cause DASH players to periodically request MPD updates. It is these DAV-generated updated MPDs that identify the location of locally available content that has been downloaded by other clients since the previous MPD update. These solutions are evaluated in a simulated environment and results are presented in Chapter 5.

5 Evaluation Setting, Results and Analysis

This section presents the evaluation of the solutions proposed and developed in this research.

5.1 *oPOAA Evaluation*

This section presents the evaluation of the proposed open Performance Oriented Adaptation Agent (oPOAA). Proof of concept evaluation tests were performed in a simulated environment, using Network Simulator version 2.29 (NS-2) [24]. NS-2 is a discrete event simulator, with substantial support for simulation of protocols at various levels of the TCP/IP networking model over heterogeneous networks. The evaluation objective is to determine if deployment of oPOAA leads to reduction in latency (see Section 2.3.2 for discussion on latency) and thus achieve improved content delivery. A number of learning objects of varying size are requested from multiple mirrored servers (of differing link quality).

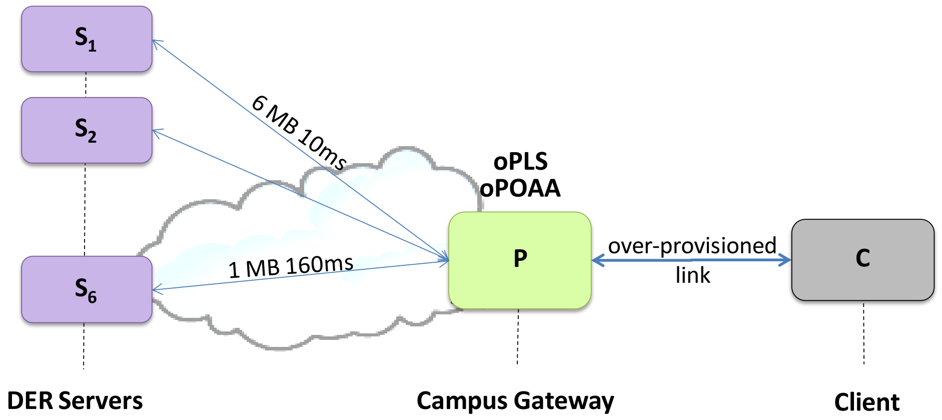

Figure 5-1: oPOAA Simulation Topology

5.1.1 Test Bed

The test setup considers the evaluation setting used in [263] as presented in Figure 5-1. The client C and DER servers (S_1, S_2, ..., S_6) are connected to a PL system server (P) on which oPOAA was installed. In order to evaluate the efficiency of the oPOAA deployment a simulation that models a university campus setting was implemented, where oPOAA resides on the university gateway server and learners use personal computers within the campus LAN. The network connections from DER servers to the server P (S_i-P) differ in terms of bandwidth and propagation delay. The network link between the PL system (P) server and the client (P-C) is over-provisioned such that no loss or significant delays are expected. This model deals with homogeneous clients in terms of the end-user device and network connection. Certain delays will occur while sending the content from the gateway server (P) to the clients, however, it is assumed that these delays will be constant and similar due to the homogeneity of the clients and are therefore not considered in this setting. Assuming that the last leg (P-C) has no major impact on the delivery performance, the calculated performance rating is based on the measurements gathered by monitoring the communication between the server (P) and the DER servers (S_i).

Characteristics of links. Six hosting servers (DERs) are considered, and measurements are taken for the relevant connecting links (P-S_i, i = 1…6). These links are of different bandwidth and delay as presented in Table 5-1. The link (P-S_1) has the best characteristics, while the quality of the other links is gradually decreased. The values used are smaller than typical bandwidths, to represent the slice of the bandwidth available for the client under some background traffic. The link between the server (P) and client is over-provisioned, assuming on-campus use. UDP is used as a transport protocol.

Link	Bandwidth	Delay
P-S_1	6MB	10ms
P-S_2	5MB	40ms
P-S_3	4MB	70ms
P-S_4	3MB	100ms
P-S_5	2MB	130ms
P-S_6	1MB	160ms

Table 5-1: Server P to DERs Link Characteristics

5.1.2 Test Scenarios

The sequence of the testing process is presented in Figure 4-6 (page 71). When a learner requests some learning content, oPOAA acting as a broker, contacts the oPL system requesting the learning content. It is assumed that the oPL system is aware of the content stored on distributed DERs. The oPL system sends back a list of relevant and suitable LOs and their sources – the SRLO List. The relevance of the LO is determined based on the current request for learning content, while the suitability is based on the user's model maintained by the oPL system. Once provided with the list of suitable LOs and the DERs where they reside, then based on the DER's performance ratings oPOAA assigns performance rating to each provided LO as described in Section 4.2.3. Finally, oPOAA requests LOs from the most efficient DERs, which guarantees that selected LOs are delivered with reduced latency. The learning content is delivered by the server (P). The aim of these tests is to compare the delivery performance in terms of download time for a system that deploys the proposed oPOAA against those measured for a system that does not employ selection of the content based on performance. Historic performance readings for five (X= 5 in Section 4.2.3) most recent deliveries from the DER are used. The simulation involves three different scenarios with three different DER selection approaches:

- Scenario 1 **Random**: oPL system randomly selects source DERs;
- Scenario 2 **oPOAA**: oPL system deploys oPOAA to select source DERs;
- Scenario 3 **Best**: oPL system gets all requested LOs from the most efficient DER.

5.1.3 Test Case

The corresponding simulation in NS-2 is depicted in Figure 5-2.

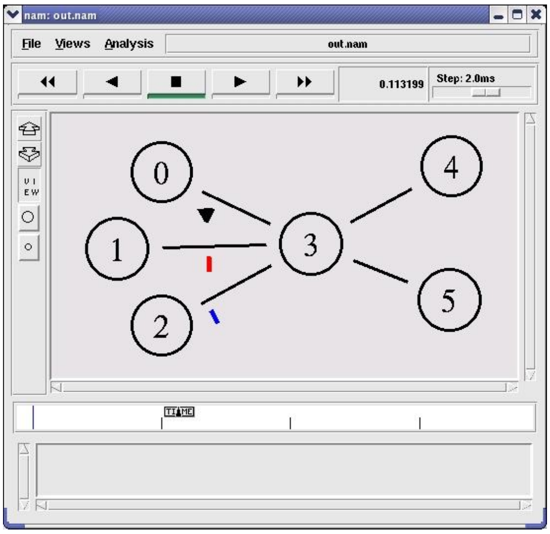

Figure 5-2: oPOAA Simulation in NS-2

Characteristics of LOs. In the test configuration, it is assumed that copies of LOs (matching learning output and suitability rating) reside on all servers. The learner requests varying numbers of LOs. Each LO is represented as a file. NS2 [24] TCL random-uniform function is used to generate files which sizes are distributed according to the uniform distribution. The minimum value of the distribution is set to be 1KB, while the maximum value is 100KB. These sizes match typical sizes of lesson reading materials (e.g. notes in PDF), or low bitrate video segment files.

Characteristics of requests. All LOs selected by oPL system – SRLO List (defined in Section 4.2.3) are requested. All requests originate from a single client as indicated in Figure 5-1. In each simulation, the number of LOs requested is varied from one to twenty. Delivery time of the requested LOs is measured in order to compare systems performance. Current adaptation aims at delivering every LO given in the SRLO List.

5.1.4 Test Case Results and Analysis

The recorded download times are presented in Figure 5-3.

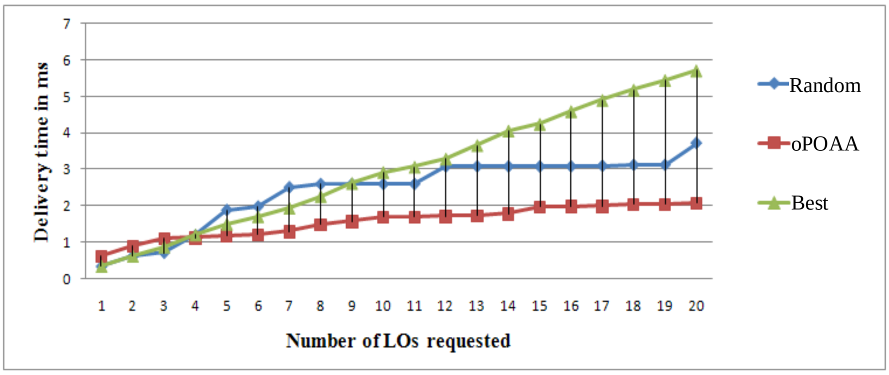

Figure 5-3: Delivery Latency with Increasing Number of LOs

These results indicate a significant improvement in performance reflected in the reduced download times when using the oPOAA-based system in comparison with the other two cases.

There is no significant difference in download times when the number of requested LOs is low (less than 5). However, the relative reduction in delivery time grows as the number of LOs increases. This is indicated with a subset of readings for the case of 10 and 20 requested LOs provided in Table 5-2.

LOs	Case 1	Case 2	Case 3
10	1.73	2.68	2.95
20	2.06	3.77	5.79

Table 5-2: Delivery Times (milliseconds) for 10 and 20 LOs

For example, when 10 LOs are requested, oPOAA enhanced system delivers requested LOs 35% faster than system with random selection of DERs and 41% faster than a system using a single DER. The difference in download times is even more significant for 20 requested LOs, namely 45% for randomly selected DERs and 64% for single DER systems.

5.2 dPOAA Evaluation

This section presents the evaluation of the proposed DASH-aware Performance Oriented Adaptation Agent (dPOAA). The aim is to illustrate that dPOAA deployment results in improved video delivery when a number of campus-based clients are requesting content of varying duration and segment length from a number of remote servers (storing identical content, but with different link characteristics). The dPOAA algorithm is evaluated in a simulated setting. The Network Simulator version 3.14 NS-3 [25] is used for modelling and simulations. Simulation objectives, setting and results are provided in the following sections.

5.2.1 NS-3 Simulator

NS-3 [25] is a free discrete-event network simulator built in C++ with the use of Python scripts for binding, that superseded the NS-2 [24]. NS-3 adopts open source (GNU GPLv2) licensing and development model, so the code can be edited to implement different network topologies and protocols. It provides alignment with real systems (e.g. sockets, device driver interfaces) and alignment with input/output standards. This was the reason for moving to NS-3, despite the fact that the simulations developed in the first part of this research were developed in NS-2. However, implementation of new modules takes time and expertise, as the simulator is complex consisting of hundreds of C++ files, for example the 802.11 module consists of 50 files. The simulator currently supports a number of TCP variants, Reno is used for the simulations. A DASH evaluation [264] using different TCP variants (Reno, Vegas and Cubic) indicated no significant performance difference (average streaming bitrate, congestion window and rate estimation) between the variants.

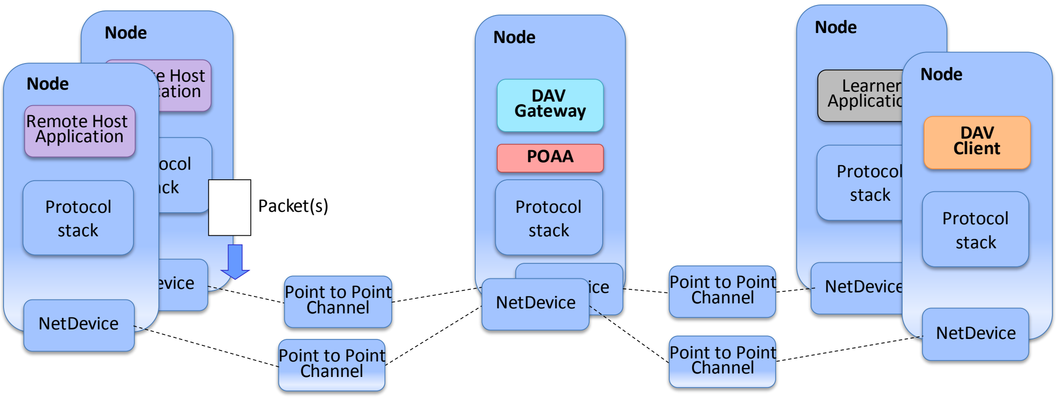

Figure 5-4: NS-3 Simulator Structure for dPOAA

Applications developed for the simulation include Remote Host (Server), Learner (DASH Client), DAV Client and DAV Gateway.

Learner Application - DASH Client has been developed to mimic behaviour of a typical DASH player. It consists of the adaptation logic and buffer modules. The adaptation logic determines the bitrate for the next requested segment. The algorithm is based on the throughput measured during the previous segment download. The quality (bitrate) of the next segment requested is calculated based on the formula used is DASH-JS [118] (see Section 2.4.8 for other approaches). The buffer class is in line with the implementation proposed in [118] and it models the fill level of the buffer. Whilst it does not store actual byte chunks, it keeps track of the content received and it is updated with every received packet. The buffer fill level is represented in buffered media time and measured in seconds. This approach is better than measuring buffer size in terms of buffered bytes as the content bitrate changes over time, e.g. segments of different bitrate are requested. The video playout starts when the buffer contains sufficient amount of data (e.g. 10 seconds of video) (see Section 2.4.7). Once the initial buffering is completed, the buffer is updated periodically every 0.2 seconds. The progress of the media being played back is modelled by subtracting 0.2 seconds from the buffer.

Remote Host Application - Server has been developed to act as a server. It accepts TCP messages with requests and it responds to the requester by sending the requested amount of data back.

DAV Client has been developed to mimic behaviour of a DAV Client. It consists of a DASH player and a server which responds to requests for content from the local nodes.

Proxy Application (DAV Gateway and ***POAA)*** has been developed to mimic behaviour of a DAV Gateway with POAA. It is aware of the content available locally and it produces new MPDs.

5.2.2 Test Bed

The test setup is presented in Figure 5-5. The client and remote servers (Server 1, Server 2, ..., Server N) are connected to the DAV Gateway on which dPOAA is deployed. In order to evaluate the efficiency of dPOAA a number of simulations were developed. These simulations model a university campus situation, where dPOAA resides on the university gateway and learners are using personal computers within the campus local area network. The network connections from the gateway to the remote servers (DAV Gateway – Server i) differ in terms of bandwidth and propagation delay. This model deals with homogeneous clients in terms of the end-user device and network connection. Here multiple clients are requesting video clips in contrast to oPOAA test bed (see Section 5.1.1).

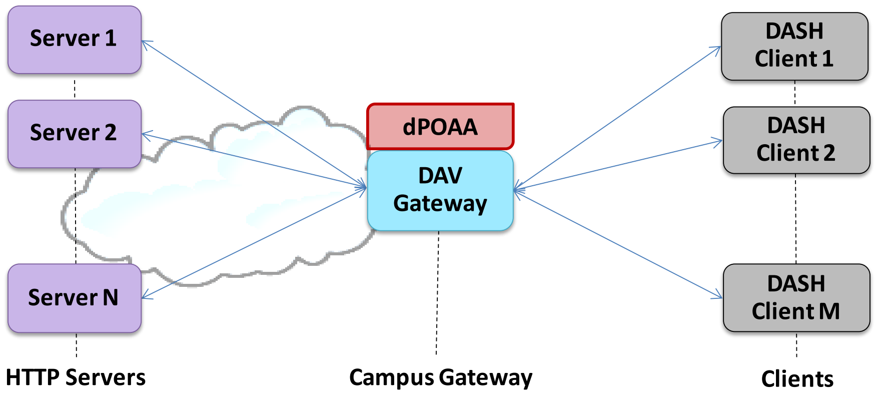

Figure 5-5: dPOAA Simulation Topology

Characteristics of Clients. Tests involve varying the number of clients from 6 up to 42 and gradually increasing the number in steps of three (as used in [126]). These limits were introduced as lower numbers of users impose no pressure on the delivery infrastructure, whereas larger numbers negatively impact the quality of delivered video (e.g. clients requesting the lowest supported bitrate). These users are students in a simulated classroom setting, being asked to watch the same video content. All students are within the campus network using well resourced devices with either wired or excellent wireless connection. The network link between the gateway and the students are hence over-provisioned such that no loss or significant delays are generated, and therefore no major impact on the delivery performance is expected.

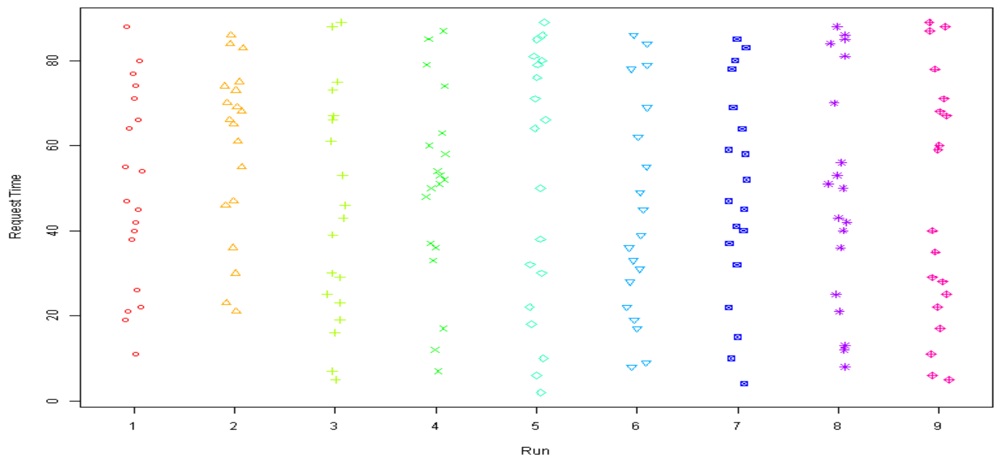

Figure 5-6: Sample Random Request Times for 20 clients (9 runs)

Characteristics of Requests. The readings are collected in a number of scenarios, where all clients were requesting all video segments from the remote servers (classical DASH situation). The clients are sending their initial request either (a) sequentially or (b) at randomly generated request times. The times are generated using a uniform random number generator provided in NS-3 [25]. An example of such a distribution where requests are made in the first 90 seconds is given in Figure 5-6.

Video Clip Characteristics. The simulations are performed with video clips of varying duration. Short videos (less than 3 minutes in duration) were identified in [217] (see Section 3.5.1) as the most engaging. Therefore, short video clips (from ~100 to ~200 seconds) are used for evaluation. It is assumed that copies of the requested videos reside on all remote servers. The requested video is delivered in segments of predefined duration. Whilst this is a simulated environment, to stay close to the real life situation, the file sizes are taken from the first freely available DASH dataset [124] that provides various full-length videos in a variety of genres, resolutions, bitrates and segment length. The Big Buck Bunny [265] test set is chosen. The Big Buck Bunny is frequently used in DASH evaluations including studies presented in [125], [127], [266], [267]. The original animation files are in AVC format. Segment duration is addressed in Section 2.4.9. Segment durations between 5 and 8 seconds were identified in [124] as the optimal segment size for similar network configuration scenarios without persistent HTTP connections. Furthermore, segments of around 10 seconds were identified as sufficient to produce the smoothed throughput measurement in [119]. Therefore evaluation tests are conducted with files of segment lengths of 6 and 10 seconds in duration. Segments in twenty bitrates ranging from 50 to 8000 kbps are provided by ITEC[17]. A study [174] investigating Akamai adaptive streaming identified the use of video encoded in five versions at different bitrates stored in separate files as given in Table 2-5. Whilst, HEVC DASH dataset [268] provides encodings with bitrates appropriate for UHDTV display resolution (e.g. 3840x2160), they are not used in our simulations, as the focus is placed on portable devices with smaller screen resolutions. It is in line with the DASH evaluations presented in [127], [267] that used bitrates range from 100 to 4500 kbps. The bitrates chosen to map the categories of end-user devices as illustrated in Table 5-3, where the simulations consider portable devices.

Device Category	**Display Resolution**	**Bitrates (Kbit)**
Handheld	320×240 & 480×360	**50**, 100, 200, 300, 400, **500, 600**
Portable	853×480 & 1280×720	**500, 600,** 700, 900, 1200, 1500, **2000**
Large-Screen	1920×1080	**2500**, 3000, 4000, 5000, 6000, **8000**

Table 5-3: Device Types and Resolutions and Corresponding Bitrates

[17] http://www-itec.uni-klu.ac.at/dash/?page_id=207 [Accessed: 2-Jan-2016]

The same source (ITEC) provides SNR_Y, SNR_U and SNR_V values for each frame of the animation video, as well as overall averages. Average SNR_Y values for each bitrate are given in Table 5-4. However, the provided average SNR values are relatively high for all bitrates. For example, for a ten fold increase in bitrate (from 50 to 500 kbps) the average SNR changed for ~6 (33.2353-27.1699), and it reflects the MOS values ranging from 4 to 3. Therefore the SNR values are not used for rating of the video quality as perceived by viewers and other methods of measuring quality were explored as outlined in Section 2.4.13. The quality metrics used for evaluation of the proposed solutions are given in Section 5.2.4.

Bitrate	50	100	200	300	400	500
Avg PSNR	27.1699	29.4168	31.1906	32.3202	32.9042	33.2353
Bitrate	600	700	900	1200	1500	2000
Avg PSNR	34.729	35.1217	36.1243	37.1951	37.8457	38.4605
Bitrate	2500	3000	4000	5000	6000	8000
Avg PSNR	40.406	41.1345	42.1054	42.6801	43.0224	43.3682

Table 5-4: Bitrates and Average SNR_Y Values from ITEC

Link Characteristics. Three different servers are considered, and measurements are taken for the appropriate links (Server i - Gateway, i = 1, 2, 3). Links are of different bandwidth and delay, the values used for the simulations are presented in Table 5-5. The link (Server1-Gateway) has the best characteristics, while the quality of the other links is gradually decreased. A good spread of network conditions is deployed ranging from good (10Mbps/15ms) to poor (2Mbps/85ms) in terms of bandwidth and delay. Links between the Gateway and Clients are over-provisioned, assuming on-campus use. The assigned delays are in line with typical DASH evaluation settings, for example the evaluation presented in [267] considers RTTs ranging from 0 to 150 ms. The links are under-provisioned to reflect the amount of bandwidth available for DASH streaming, and hence no other background traffic was utilised.

Link	Bandwidth	Delay
Server1 – DAV Gateway	10 Mbps	15 ms
Server2 – DAV Gateway	6 Mbps	50 ms
Server3 – DAV Gateway	2 Mbps	85 ms
DAV Gateway – Clients	100 Mbps	0.5 ms

Table 5-5: Remote Server – DAV Gateway Link Characteristics

5.2.3 Test Scenarios

The sequence of the testing process is presented in Figure 5-7 where Client represents viewing devices, dPOAA represents dPOAA deployed at the DAV Gateway, PLS represents Personalised Learning (PL) system and Remote_Host represents server hosting selected video.

When a learner requests learning content, the DAV Gateway intercepts the request for the PL system. The PL system identifies the video relevant for the learner's learning objective and sends the MPD URL back to the DAV Gateway. dPOAA then calculates the rating for each remote host storing the recommended video based on the BaseURL element in the provided MPD file. The DAV Gateway then modifies the MPD file so that only the URL of the remote server with the highest dPOAA rating is left. This in turn forces the client to request the segments from the remote server "selected" by dPOAA. To find a balance between being sufficiently responsive to past observations (large ε) and being overly responsive to latest events (small ε) and thus being perturbed by short lived noise and fluctuations, the value used for the tests is ε = 3 seconds (all readings fresher than 3 seconds get weight 1, weight of older readings is calculated based on equation (4.3.3.4)). The value of w is set to 0.5 equally weighting the contribution of throughput and RTT.

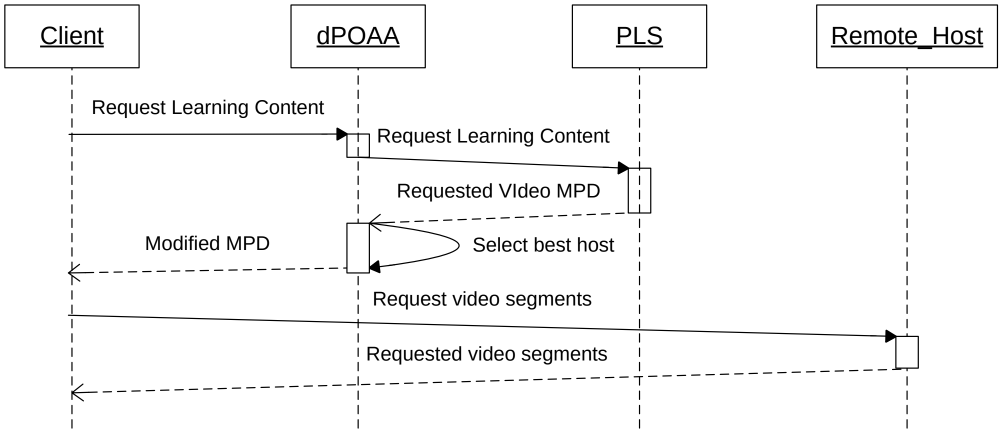

Figure 5-7: dPOAA Evaluation Sequence Diagram

To the best of our knowledge, there is no other similar DASH-based solution for selection of remote servers based on statistical estimators, so the simulations involve four cases using scenarios with different remote host selection approaches. A simple algorithm name is provided in brackets, and it is used for test result identification in the graphs provided.

The readings are collected for four scenarios. Scenario 1 – (dPOAA) involves informed hosting server selection - dPOAA selects the hosting servers according to equations (4.3.3.2) and (4.3.3.3) with X = 5 (results indicated in red - **o**). Scenario 2 - (oPOAA) involves informed hosting server selection - oPOAA selects the hosting servers based on the past server throughput and round trip times in order to reduce initial delays (results indicated in green - Δ). Scenario 3 – (RandS) involves a uniformly distributed random hosting server selection based on the uniform random number generator provided in NS-3 [25] (results indicated in turquoise - +). Scenario 4 – (BestS) involves a setting where all segments are requested from the same server – Server 1 which is the best connected server (results indicated in purple - **x**).

The clients are sending their initial request for the recommended video at randomly generated intervals ranging from 0.1 to 90 seconds.

5.2.4 Test Evaluation Metrics

The algorithms evaluated are compared according to the metrics proposed in [51] (see Section 2.4.13). So the following readings are recorded and compared:

- ***Join Time*** – the pre-buffering time (startup time), calculated as the time that lapses from the initiation of the connection until the client buffer reaches the playout level;
- ***Rebuffering Ratio*** - the relative time spent on rebuffering, calculated as the duration (total time) of buffer starvation over the total length of playout including pauses for rebuffering;
- ***Rebuffering Rate*** – the relative frequency of induced interruptions, calculated as the number of buffering events over the video playout time;
- ***Average Bitrate*** - the average of segment bitrates requested.

Rebuffering Ratio is expressed as a percentage, while Rebuffering Rate is given in number of rebuffering events per minute of video playout. Furthermore, estimated Mean Opinion Scores (***MOS***) were calculated and compared. The MOS is determined based on an equation that considers respective levels of Initial Buffering Time (Join time), Rebuffering Frequency (how frequent the rebuffering events are), and Mean Rebuffering duration (the average duration of a rebuffering event) as proposed in [134] (see Equation (2.4.13.1) in Section 2.4.13).

Stalls are points of buffer starvation during video playout. They take place when the video buffer level drops below a certain level (e.g. 0.4 seconds of available video data) and the player halts playout and waits for the video buffer to be replenished. All stalls are recorded, and the ***Average Stall*** is the average time spent in rebuffering for all clients. TCP is a reliable transport protocol, and all requested content will be delivered even when the delivery network is overburdened. Video playout is interrupted with rebuffering events (due to delays and retransmissions) during congestion periods; however the ***Average Bitrate*** of the delivered video does not reflect these disruptions in playout. Furthermore, high video bitrate values may force clients to switch down the bitrate because of buffering induced by poor network conditions [51] and thus stalls are more noticeable for higher bitrates [71]. There is a tradeoff between some evaluation criteria. For example, a longer join time (initial buffering) typically results in fewer interruptions of playback due to rebuffering at later stages, to which viewers are sensitive [19] (see Section 2.4.13). However, prolonged initial delays negatively impact on viewer retention as their tolerance drops at a certain point (around 15 seconds for join times) [51]. The same players i.e. DASH Clients (bandwidth estimation formula is given in equation (2.4.8.2), see Section 5.2.1 for other simulation modules) are used in all scenarios. The player requests the first segment at the lowest bitrate available for the device in question and it begins playout when a buffer level of 10 seconds (e.g. ~ segment duration) is reached. Therefore, the initial waiting times are short (under 15 seconds).

More information about QoE/QoP and latency is provided in Section 2.3.2, while Section 2.4.13 presents various approaches for evaluating HTTP streaming quality. The results are presented with line graphs. The X axis values represent the number of clients requesting the video (ranging from 6 to 42 in steps of 3). The Y axis values specific for each metric are given in Table 5-6.

Reading	Min Value	Max Value	Units
Join Time	0	15	Seconds
Rebuffering Ratio	0	30	N/A (percentage)
Rebuffering Rate	0	60	Number of events per minute
Average Bitrate	450	1800	Kbps
Average Stall	0	150	Seconds
Average MOS	0	5	N/A

Table 5-6: Y Axis Values for Result Graphs

5.2.5 Results and Analysis

All client buffers were monitored to determine interruptions when less than 0.4 seconds of video data is stored in the buffer. Data transmission performance during the simulations is measured in terms of the metrics described in Section 5.2.4. The results of the evaluation show reductions in both the rebuffering rate and join time with improvements particularly evident when longer video clips are streamed to a larger number of clients. These tests were performed to determine if dPOAA deployment depends on the requested video clip duration and the video segment duration when a varying number of clients (nodes) are requesting video at randomly generated intervals as indicated in Table 5-7.

Test Case Impact investigation	Segment Duration	Server-DAV Gateway		Request intervals	Segments Requested
		Bandwidth Mbps	Delay ms		
TC1 POAA_6s	6 s	10, 6, 2	15, 50, 85	1 – 90 s	16, 25 and 33
TC2 POAA_10s	10 s	10, 6, 2	15, 50, 85	1 – 90 s	10 and 20

Table 5-7: dPOAA Evaluation Test Cases

5.2.5.1 TC1 POAA_6s - Results and Analysis

Simulations were performed with clients requesting at random intervals, in order to determine if dPOAA performance depends on the number of clients concurrently requesting the same video in segment sizes of 6 seconds under the settings given in Table 5-8.

Property	Segment Duration	Server-Gateway		Random Request Time	Video Duration	Number of Segments Requested
		Bandwidth	Delay			
Value	6 s	10Mbps 6 Mbps 2 Mbps	15ms 50ms 85ms	1 – 90 s	96, 150 and 198s	16, 25 and 33

Table 5-8: TC1 POAA_6s Test Setting

The evaluation metrics are specified in Section 5.2.4. The results are presented in graphs in Figure 5-8 to Figure 5-13. Additionally, a representative subset of the evaluation results is provided in tabular form. The evaluation results suggest that the join time is reduced as indicated in Figure 5-8 and Table 5-9. When the number of clients requesting video is small (e.g. 6 clients) oPOAA outperforms dPOAA, and the BestS scenario produces the shortest join times. The oPOAA algorithm selects the remote server based on initial delay estimation, and consequently succeeds in reducing join times. The BestS scenario utilises the best server provisioned link and marginally outperforms (e.g. 30 ms longer waiting times as given in Table 5-9) dPOAA when 6 clients are requesting video. In all other cases, the best performing scenario is dPOAA, and the reductions in join times are more significant with increasing numbers of requesting clients.

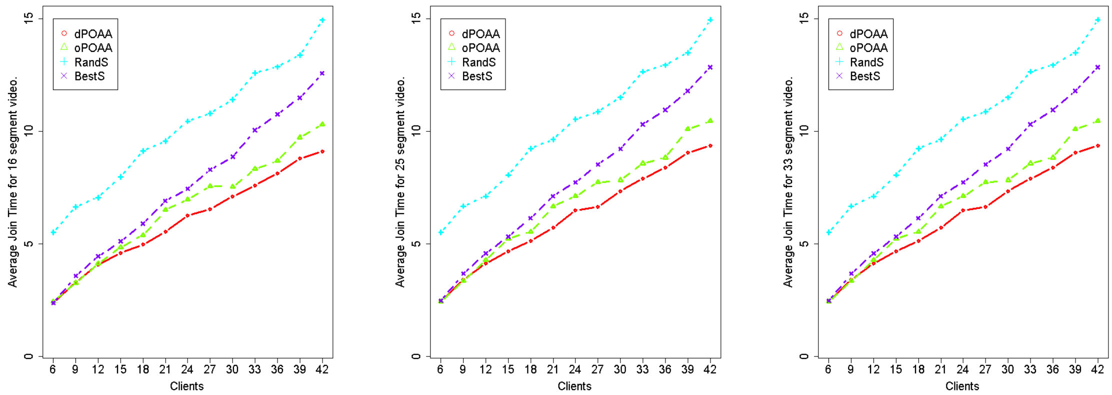

Figure 5-8: TC1 POAA_6s Average Join Time (seconds)

The subset (cases with 6, 24 and 42 clients only) of evaluation results provided in Table 5-9 indicate that, due to the random initial request timing taking place within 90 seconds, the video duration does not affect the join times when the video is longer than 150 seconds.

No. of Segments	No. of Clients	dPOAA (seconds)	oPOAA (seconds)	RandS (seconds)	BestS (seconds)	dPOAA vs RandS (%)	dPOAA vs BestS (%)	dPOAA vs oPOAA
16	6	2.40	2.44	5.59	2.37	57.08	-1.16	1.71
	24	6.27	6.99	10.27	7.45	38.92	15.82	10.21
	42	9.12	10.31	14.60	12.58	37.52	27.51	11.52
25	6	2.47	2.44	5.61	2.49	56.06	1.00	-1.16
	24	6.50	7.13	10.37	7.74	37.37	16.10	8.86
	42	9.36	10.47	14.64	12.86	36.05	27.16	10.55
33	6	2.47	2.44	5.61	2.49	56.06	1.00	-1.16
	24	6.50	7.13	10.37	7.74	37.37	16.10	8.86
	42	9.36	10.47	14.64	12.86	36.04	27.16	10.55

Table 5-9: TC1 POAA_6s Average Join Time

Utilisation of a dPOAA algorithm improves the average bitrate compared to the BestS setting as indicated in Figure 5-9. However, there is no improvement compared to the random (RandS) and oPOAA scenarios. As discussed in Section 5.2.4, the average bitrate does not reflect the interruptions in playout. While high average bitrates are maintained in the RandS scenario, playout suffered from stalls as extreme as 51.33 seconds for 42 clients requesting 96 seconds of video (indicated in Table 5-13 on page 114). The evaluation results are summarised in Table 5-10 which contains average bitrates for the cases with 6, 24 and 42 clients only. The negative values indicate that the dPOAA average bitrate was lower than that in the RandS/oPOAA scenarios.

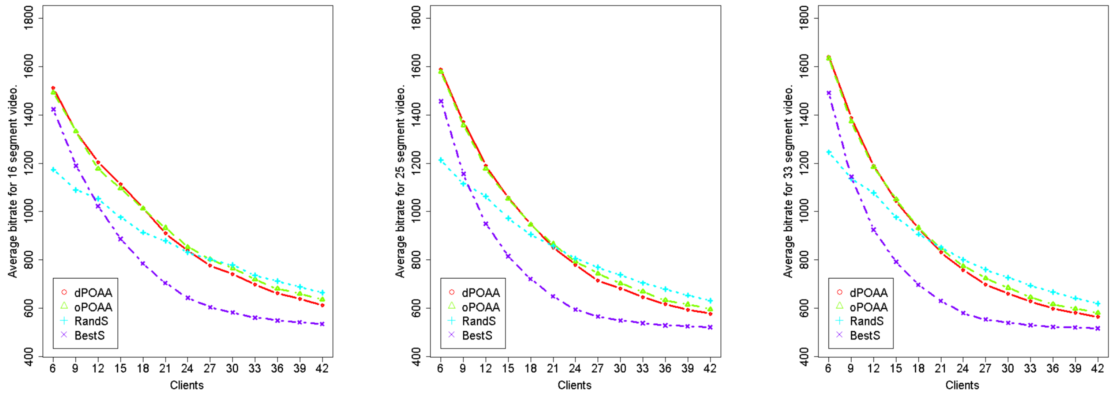

Figure 5-9: TC1 POAA_6s Average Bitrate (kbps)

No. of Segments	No. of Clients	dPOAA (kbps)	oPOAA (kbps)	RandS (kbps)	BestS (kbps)	dPOAA vs RandS (%)	dPOAA vs BestS (%)	dPOAA vs oPOAA
16	6	1510.83	1492.75	1174.15	1423.33	28.67	6.15	1.21
	24	837.79	853.08	830.49	643.08	0.88	30.28	-1.79
	42	612.21	635.52	665.17	534.82	-7.96	14.47	-3.67
25	6	1588.33	1577.92	1213.65	1456.67	30.87	9.04	0.66
	24	779.75	793.02	806.47	594.56	-3.31	31.15	-1.67
	42	576.98	595.27	631.32	521.25	-8.61	10.69	-3.07
33	6	1640.00	1634.75	1246.20	1491.67	31.60	9.94	0.32
	24	758.58	774.06	800.99	579.31	-5.29	30.95	-2.00
	42	564.32	580.63	619.71	516.69	-8.94	9.22	-2.81

Table 5-10: TC1 POAA_6s Average Bitrate

Deployment of dPOAA significantly reduces the average rebuffering ratio compared to oPOAA and RandS as depicted in Figure 5-10. Tabular data (Table 5-11) indicates that dPOAA stays within 3.5% of playout time even in the extreme case of 42 clients. In contrast, BestS exceeds this threshold (3.5%) at 36/33/30 clients for videos of 16/25/33 segments, respectively. Furthermore, dPOAA outperforms BestS starting from 33/30/27 clients (16/25/33 segment video). While the negative values indicate that dPOAA introduces a degree of rebuffering, dPOAA stays within the 1% threshold (40% of viewers experience at least 1% rebuffering ratio [136]) for up to 39/36/33 clients (16/25/33 segment videos).

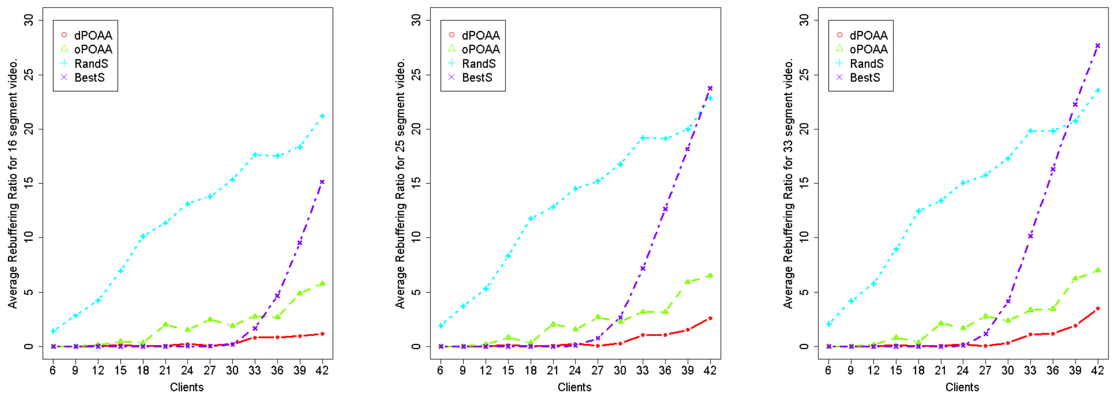

Figure 5-10: TC1 POAA_6s Average Rebuffering Ratio

No. of Segments	No. of Clients	dPOAA	oPOAA	RandS	BestS	dPOAA vs RandS (%)	dPOAA vs BestS (%)	dPOAA vs oPOAA
16	6	0.00	0.00	1.43	0.00	N/A	N/A	N/A
	24	0.21	1.52	13.12	0.00	98.41	N/A	86.87
	42	1.16	5.77	21.19	15.13	94.51	92.32	60.16
25	6	0.00	0.00	1.89	0.00	N/A	N/A	N/A
	24	0.21	1.59	14.54	0.06	98.57	-259.30	86.87
	42	2.59	6.51	22.83	23.76	88.65	89.09	60.16
33	6	0.00	0.00	2.06	0.00	N/A	N/A	N/A
	24	0.18	1.68	15.06	0.08	98.81	-116.92	89.35
	42	3.47	6.98	23.56	27.69	85.27	87.47	50.29

Table 5-11: TC1 POAA_6s Average Rebuffering Ratio

Similarly, Figure 5-11 depicts rebuffering rates where dPOAA significantly reduces the rebuffering rate compared to oPOAA and RandS scenarios. The evaluation results are summarised in Table 5-12 which presents average rebuffering rates for the cases with 6, 24 and 42 clients only. The negative values indicate that dPOAA introduces a degree of rebuffering. However, dPOAA stays within the bounds of 0.4 stops per minute up to 24 clients and peaks at 6.85 stops per minute for 42 clients. In contrast, BestS exceeds this threshold at 36/33/30 clients for videos of 16/25/33 segments, respectively.

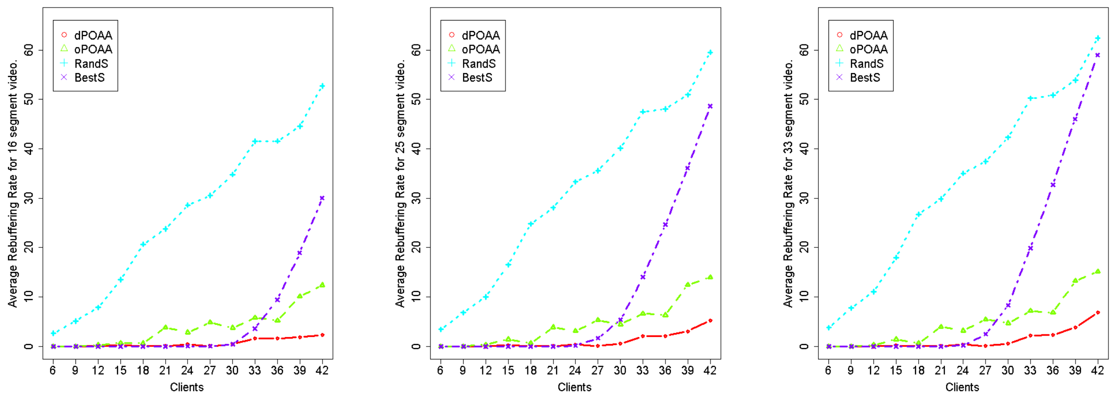

Figure 5-11: TC1 POAA_6s Average Rebuffering Rate (per minute)

No. of Segments	No. of Clients	dPOAA	oPOAA	RandS	BestS	dPOAA vs RandS (%)	dPOAA vs BestS (%)	dPOAA vs oPOAA
16	6	0.00	0.00	2.65	0.00	N/A	N/A	N/A
	24	0.39	2.82	28.63	0.00	98.64	N/A	86.16
	42	2.31	12.41	52.73	30.01	95.61	92.29	81.36
25	6	0.00	0.00	3.44	0.00	N/A	N/A	N/A
	24	0.39	3.11	33.32	0.17	98.84	-130.14	87.56
	42	5.20	14.01	59.56	48.65	91.27	89.31	62.88
33	6	0.00	0.00	3.78	0.00	N/A	N/A	N/A
	24	0.32	3.19	35.05	0.20	99.08	-57.60	89.93
	42	6.85	15.14	62.45	58.97	89.03	88.39	54.78

Table 5-12: TC1 POAA_6s Average Rebuffering Rate

It should be noted (see Figure 5-10 and Figure 5-11) that oPOAA, which focuses on the reduction of the initial delay, introduces some level of rebuffering from 15 requesting clients onward. However, this is compensated for by reductions in join times for large numbers of requesting clients (more than 30) compared to RandS and BestS.

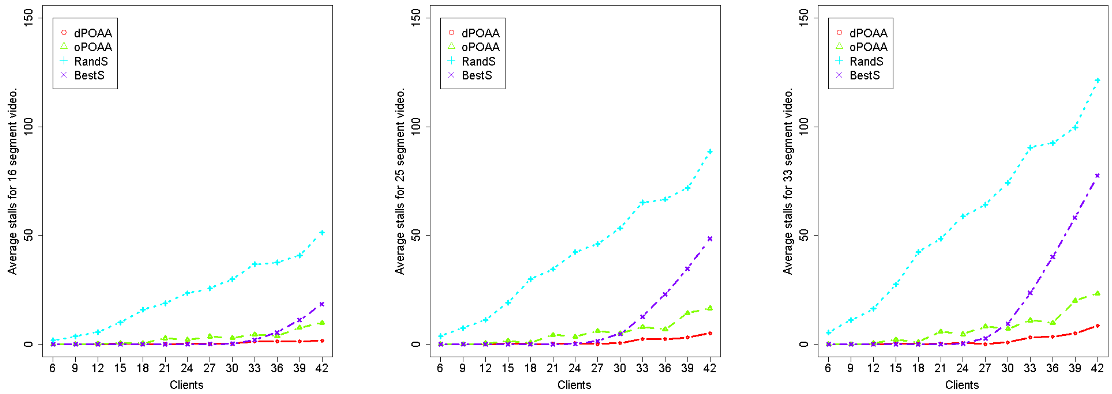

Figure 5-12: TC1 POAA_6s Average Stalls (seconds)

No. of Segments	No. of Clients	dPOAA (seconds)	oPOAA (seconds)	RandS (seconds)	BestS (seconds)	dPOAA vs RandS (%)	dPOAA vs BestS (%)	dPOAA vs oPOAA
16	6	0.00	0.00	1.80	0.00	N/A	0.00	0.00
	24	0.25	2.02	23.44	0.01	98.95	-2400	87.83
	42	1.58	9.79	51.33	18.46	96.92	91.42	83.83
25	6	0.00	0.00	3.71	0.00	N/A	0.00	0.00
	24	0.38	3.36	42.34	0.12	99.11	-207.46	88.76
	42	5.00	16.52	88.68	48.51	94.36	89.69	69.73
33	6	0.00	0.00	5.36	0.00	N/A	0.00	0.00
	24	0.42	4.57	58.87	0.21	99.28	-97.67	90.74
	42	8.52	23.25	121.33	77.51	92.97	89.00	63.34

Table 5-13: TC1 POAA_6s Average Stalls

Figure 5-12 indicates that the deployment of POAA algorithms reduces stalls with increasing numbers of clients requesting video (i.e. more than 27 clients). Here dPOAA significantly outperforms oPOAA. By contrast, in certain cases (i.e. 15 – 27 clients) both oPOAA and dPOAA introduce some stalls while BestS maintains stall-free playout. However, these stalls are short (less than 0.5 seconds per video clip) for dPOAA. The evaluation results are summarised in Table 5-13 which presents average stall durations for the cases with 6, 24 and 42 clients only. The negative values indicate that dPOAA introduced stalls (compared to BestS scenario).

Figure 5-13 indicates that the deployment of POAA algorithms maintains high levels of MOS (close to 4) for up to 33 clients and maintains acceptable levels (above 3) even with increasing numbers of requesting clients. Consistently, dPOAA outperforms oPOAA. The evaluation results are summarised in Table 5-14 which presents average MOS levels for the cases with 6, 24 and 42 clients only.

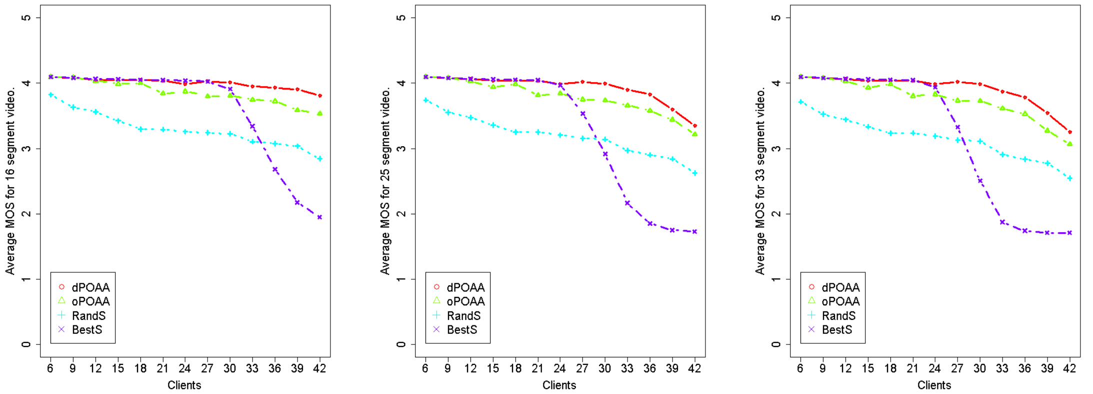

Figure 5-13: TC1 POAA_6s Average MOS

No. of Segments	No. of Clients	dPOAA	oPOAA	RandS	BestS	dPOAA vs RandS (%)	dPOAA vs BestS (%)	dPOAA vs oPOAA
16	6	4.10	4.10	3.82	4.10	7.14	0.01	0.00
	24	3.99	3.87	3.26	4.04	22.45	-1.21	2.97
	42	3.81	3.53	2.85	1.95	33.98	95.82	7.85
25	6	4.10	4.10	3.74	4.10	9.46	0.01	0.00
	24	3.99	3.84	3.20	3.97	24.50	0.48	3.80
	42	3.35	3.22	2.63	1.73	27.54	94.18	4.12
33	6	4.10	4.10	3.72	4.10	10.27	0.01	0.00
	24	3.99	3.83	3.19	3.94	25.17	1.32	4.19
	42	3.25	3.07	2.54	1.71	27.93	90.57	6.12

Table 5-14: TC1 POAA_6s Average MOS

In conclusion, the evaluation results suggest that when the clients request the same video (in 6 second segments) at random intervals over an initial 90 second window, the deployment of the dPOAA algorithm reduces join times while maintaining average bitrate levels. The estimated MOS levels are maintained at acceptable levels (above 3.25, see Table 2-1 (page 23) for MOS levels). The most significant improvements are observed when more than 27 clients are requesting video (i.e. the connecting link is congested). For fewer than 27 clients, the BestS scenario performs just as well.

5.2.5.2 TC2 POAA_10s - Results and Analysis

These tests were performed in order to determine if dPOAA performance depends on video segment duration (see Section 2.4.9 for segment duration discussion). Clients request the video under the settings given in Table 5-15. As before, the scenarios are compared in terms of the evaluation metrics given in Section 5.2.4. The results are presented in Figure 5-14 to Figure 5-19. Tabular data is omitted for brevity.

Property	Segment Duration	Server-Gateway		Random Request Time	Video Duration	Segments Requested
		Bandwidth	Delay			
Value	10 s	10 Mbps 6 Mbps 2 Mbps	15ms 50ms 85ms	1 – 90 s	100 and 200s	10 and 20

Table 5-15: TC2 POAA_10s Test Setting

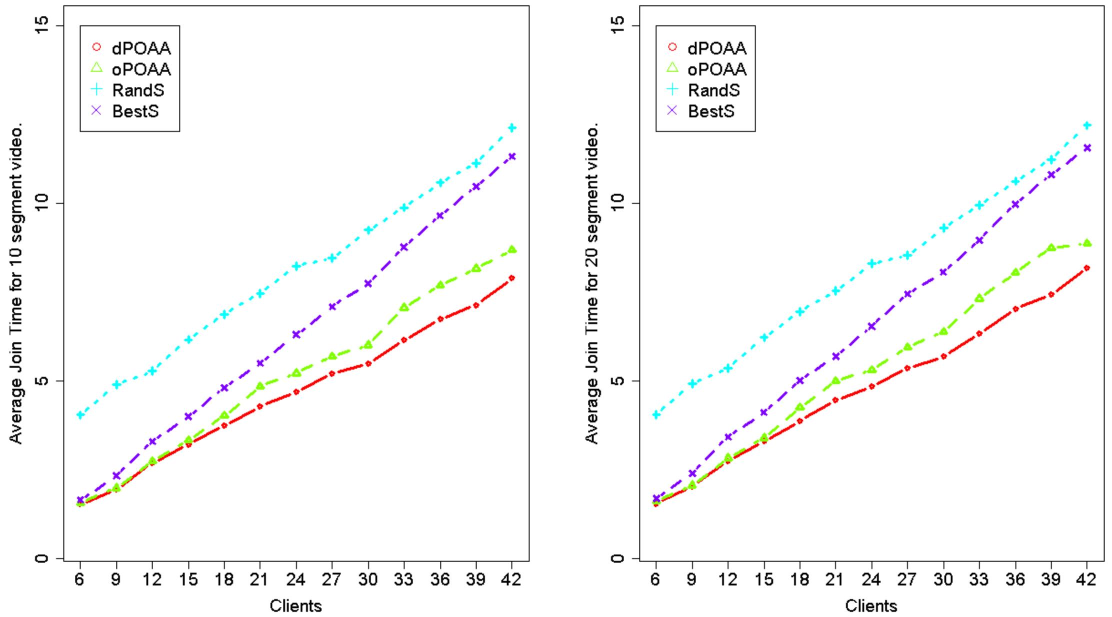

Figure 5-14: TC2 POAA_10s Average Join Time (seconds)

The results indicate that the join time has decreased for all scenarios. The best performing scenario is the dPOAA scenario as indicated in Figure 5-14. For example, in the case of 24 clients requesting ~100 seconds of content, the join time (seconds) has decreased from the TC1 POAA_6s average as follows: dPOAA from 6.27s in TC1 to 4.7s in TC2, oPOAA from 6.99s in TC1 to 5.22s in TC2, RandS from 10.27s in TC1 to 8.24s in TC2 and BestS from 7.45s in TC1 to 6.3s in TC2. This is expected as in TC2 the segments used are longer (10 seconds) than in TC1 (6 seconds). As the lowest bitrate is requested first the player receives a smaller amount of data for the first 10 s of video playout.

The utilisation of a POAA algorithm improves the average bitrate compared to the BestS setting as indicated in Figure 5-15. However, in line with TC1 POAA_6s results (see discussion on page 111), there is no improvement compared to the RandS scenario. It should be noted that when longer (10s) segments are used, fewer numbers of segments are requested (e.g. 10 or 20) per video clip. The bitrate of the requested video may only change (increase or decrease) at segment boundaries, consequently there are fewer opportunities to increase the bitrate, so the average bitrate is lower than in TC1.

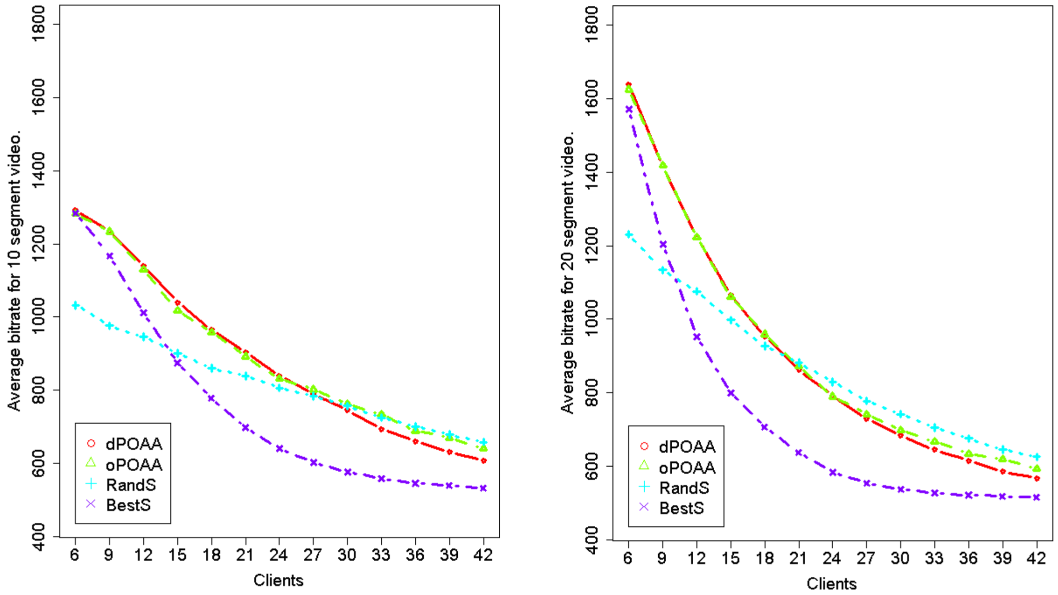

Figure 5-15: TC2 POAA_10s Average Bitrate (kbps)

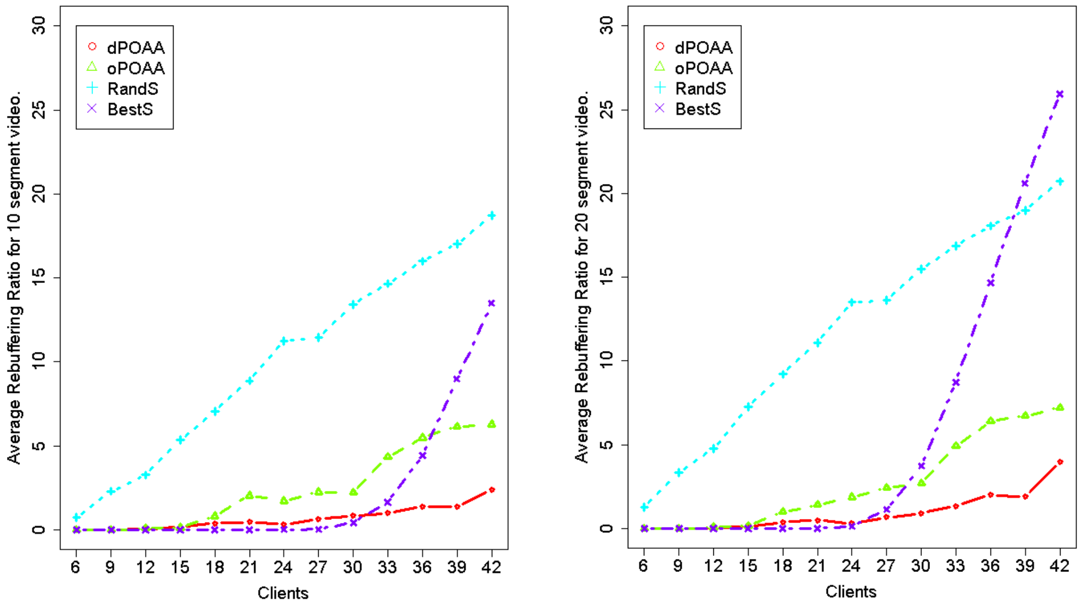

Figure 5-16: TC2 POAA_10s Average Rebuffering Ratio

POAA deployment significantly reduces both the rebuffering ratio (Figure 5-16) and rate (Figure 5-17) for an increased number of clients (more than 27). Here (similar to TC1 findings) dPOAA outperforms oPOAA.

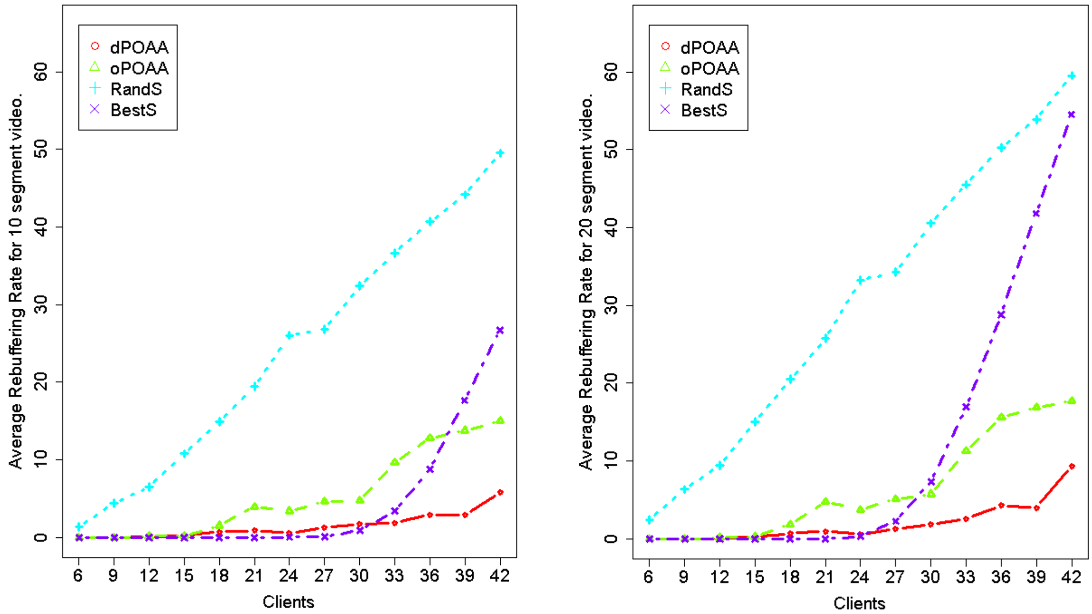

Figure 5-17: TC2 POAA_10s Average Rebuffering Rate (per minute)

Figure 5-18 indicates that the deployment of POAA algorithms significantly reduces stalls in the case of more than 27 requesting clients. As depicted, dPOAA significantly outperforms oPOAA.

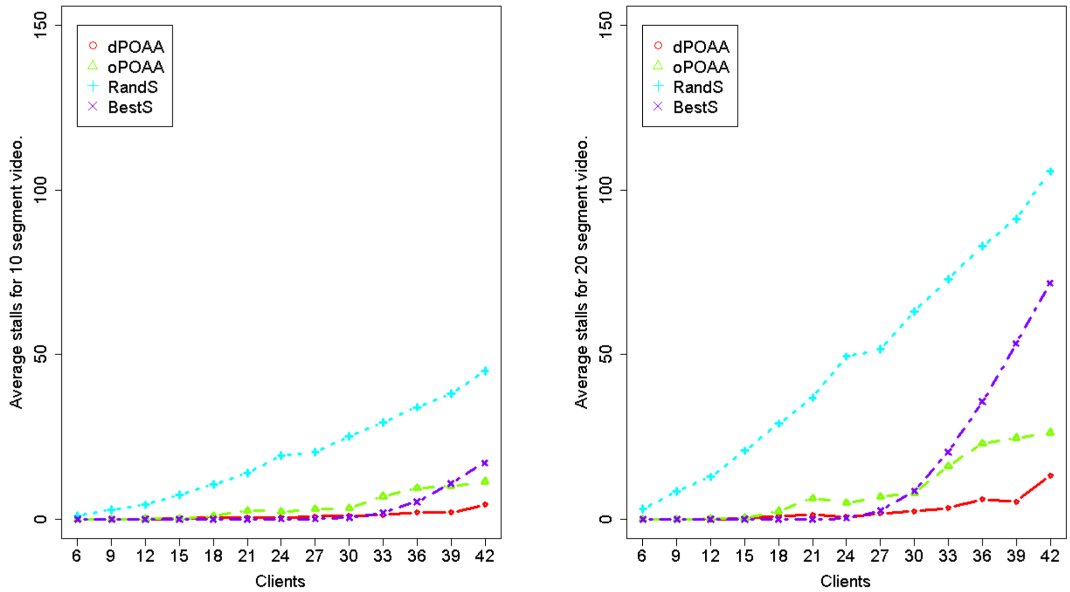

Figure 5-18: TC2 POAA_10s Average Stalls (seconds)

Figure 5-19 demonstrates that the deployment of POAA algorithms maintains acceptable levels (above 3) of MOS even with increasing numbers of requesting clients. Consistently, dPOAA outperforms oPOAA.

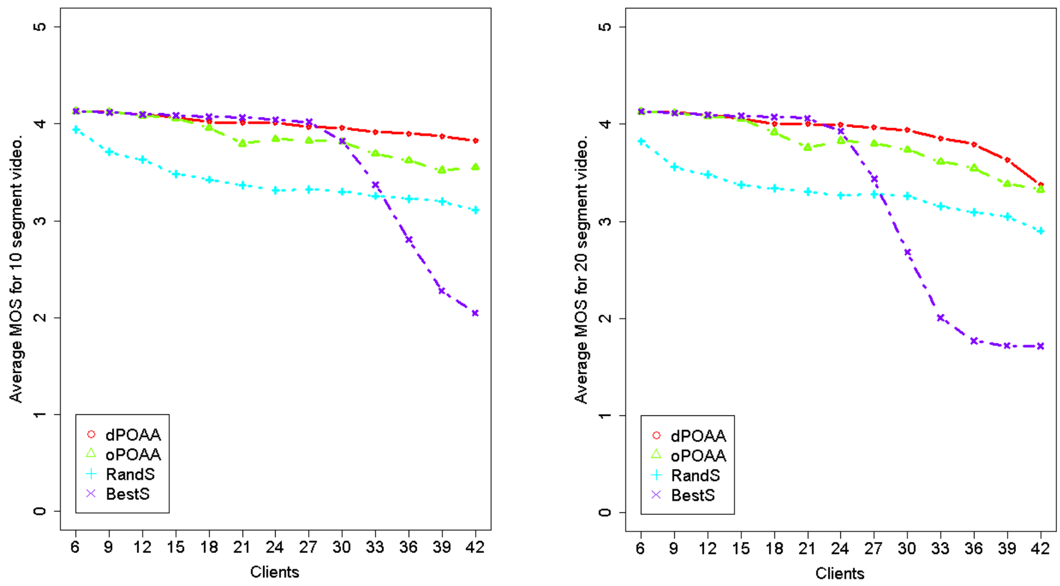

Figure 5-19: TC2 POAA_10s Average MOS

Overall results of TC2 (for 10 second segments), are in line with those for TC1 (6 second segments), suggesting that the dPOAA scenario clearly outperforms all others in cases when more than 27 clients are requesting video.

5.2.6 dPOAA Evaluation Summary

Obtaining all segments from the same server, albeit the best provisioned one, results in a prolonged initial delay proportional to the increasing number of concurrent client requests. However, users are more concerned with the higher frequency and duration of playback interruptions which is also reflected in the high rebuffering ratio where in some cases almost one third of the total playout time is made up of stalls required to replenish the buffer when all clients request from a single server. The introduction of a server selection algorithm reduces join times. However, random selection results in unpredictable video quality and therefore a better informed selection algorithm is required. Both oPOAA and dPOAA base their decisions on historic server-gateway link performance and outperform other approaches when connecting links are congested with requests (more than 27 clients requesting).

Overall, these results indicate that even a relatively simple, but informed remote server selection algorithm improves overall viewing experience for a large number of requesting clients.

5.3 DAV Evaluation

The proposed DAV algorithm is evaluated in a simulated setting. The Network Simulator version 3.14 NS-3 [25] is used for modelling and evaluation. The NS-3 simulator is described in Section 5.2.1. The simulator test bed configuration and the deployment of developed models are presented in Section 5.3.1, whereas the test scenarios are described in Section 5.3.2. Other simulation setting aspects are addressed in Section 5.2.2. The evaluation results are compared in terms of join time, buffering ratio, rate of buffering events, average bitrate and estimated MOS as described in Section 5.3.3.

5.3.1 Test Bed

The goal of the tests is to demonstrate that by using DAV improved video quality for delivered video content is achieved in comparison with (a) a classic DASH approach – fetching all content from the remote servers and (b) alternative DASH-based algorithms. The simulations consider a wired campus (local) network and remote server connections, as illustrated in Figure 5-20.

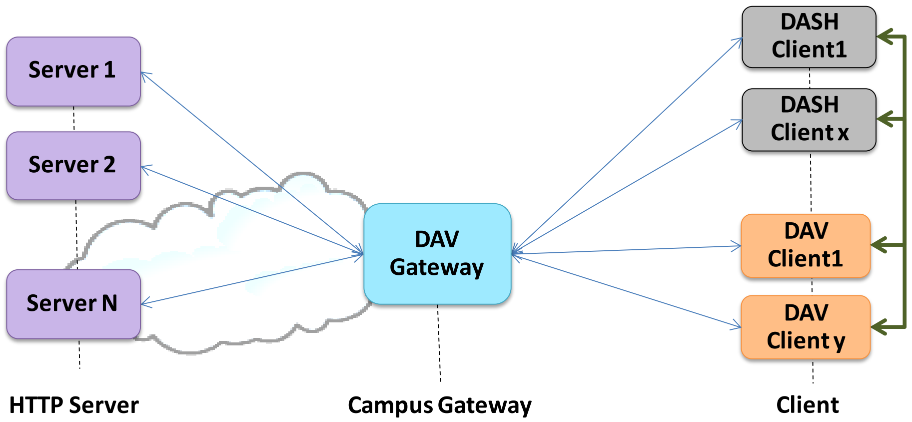

Figure 5-20: DAV Simulation Setting

A setting where during a laboratory/tutorial, a group of students is asked to watch a video clip is simulated. The PL system selects the video based on student learning profile, and a group of X students watches the same video clip using their laptops or university provided PCs (dL level used is f(*Portable–Wired*) = 20, as indicated in Table 4-9, ε = 2 seconds, β = 30 seconds). Segment bitrates range from 500 to 2000 kbps as indicated in Table 5-3. The links between remote servers and the DAV Gateway vary in data rate and delay as indicated for each test case. Simulations were performed with a varying number of clients each requesting a varying number of video segments. Each segment is either 6 or 10 seconds long. The simulated client's buffer is monitored to determine any stalls (intervals when the buffer fill level falls below the level of 0.4 seconds of video data). The initial playout buffer level is 10 seconds. The maximum buffer capacity is 35 seconds. The client players use the bandwidth estimation formula given in equation 2.4.8.2. Simulation settings such as client, video and request timing are addressed in Section 5.2.2.

5.3.2 Test Scenarios

The readings are collected for three scenarios. **Scenario 1 - *BestS*** involved all clients requesting all content from one remote server and represents a typical DASH setting where all segments are requested from the remote servers (results indicated in red - **o**). **Scenario 2 - *sDAV*** utilises local nodes as content providers (results indicated in green - **∆**). In this scenario, clients are provided with a new static MPD at the time of the request for the video. It should be noted, that sDAV approach, similar to pDASH [117], modifies MPDs to utilise content previously downloaded by other viewers (but only content available at the time the first request is made). However, there are differences, for example sDAV (unlike pDASH) utilises segments available within the LAN. Furthermore, pDASH selects peer hosts randomly, while sDAV uses a selection algorithm. sDAV is expected to outperform pDASH, as sDAV deploys a selection algorithm to identify the best performing node hosting the required segment version within the LAN. **Scenario 3** – ***dDAV*** utilises local nodes as content providers (results indicated in purple - +). In this scenario, clients are provided with a new dynamic MPD at the time of the request for the video, and the MPD is updated at the time of each segment request. In both Scenarios 2 and 3, half of the clients are DAV Clients and act as local segment providers and n - 2 out of n clients are provided with DAV generated MPDs and thus utilise local content if the requested content is available. Clients come online (and submit their initial request) either (a) sequentially or (b) at randomly generated intervals. In the sequential setting, the video (the initial request) is requested at constant t, $t \in \{2, 4, 6\}$ seconds intervals. Under the random setting, request times are generated using the uniform random number generator provided in NS-3 and range from 0.1 to 90 seconds. Evaluation settings use a single remote server so there is no need for remote server selection. The focus of the evaluation is solely on the merits of no local content vs local content with static MPDs vs. local content with dynamic MPDs. The number of requested segments is varied (e.g. 16, 25 and 33 for 6 second segments) and the result charts are generated for each number of requested segments.

5.3.3 Test Evaluation Metrics

The simulations mimic DASH-formatted video delivery (see Section 5.2.4), so the algorithms evaluated are compared according to the following metrics:

- ***Join Time*** – the pre-buffering time, calculated as the time that lapses from the initiation of the connection until the client buffer reaches playout level;
- ***Rebuffering Ratio*** - the relative time spent on rebuffering, calculated as the duration (total time) of buffer starvation over the total length of playout including pauses for rebuffering;
- ***Rebuffering Rate*** – the relative frequency of induced interruptions, calculated as the number of buffering events over the video playout time;
- ***Average Bitrate*** **-** the average segment bitrate presented in [51].

Reading	Start Value	End Value	Units
Join Time	0	15	seconds
Rebuffering Ratio	0	30	N/A (percentage)
Rebuffering Rate	0	60	Number of events per minute
Average Bitrate	450	1800	Kbps
Content from Server	0	30	Mb
Content from DAV Clients	0	30	Mb

Table 5-16: Y Axis Values for Result Graphs

Furthermore, the algorithms are compared based on the amount of data requested from the remote servers and campus network (content originating from DAV Clients). The results are presented with line graphs. The X axis values represent the number of clients each requesting the video clip (values range from 6 to 42). The Y axis values are specific for each metric reading and are indicated in Table 5-16.

5.3.4 Test Cases

These test cases evaluate the benefits of deploying DAV in a setting where one remote server provides content for the campus network. A bottleneck link of varying bandwidth and delay between the remote server and the DAV Gateway is imposed. Simulations were performed with *n* clients (*n* ranges from 6 to 42 in steps of 3 or 6), each requesting video of varying duration (96/100, 150, 198/200 seconds). The videos consist of a whole number of segments. The duration of segments varies and is specified in the test setting. Portable wired devices are used as clients. These tests were performed to determine if DAV performance and benefits depend on the segment duration (6 second vs. 10 second segments), the requested video duration, the initial request timing (sequential vs. random) and the number of nodes requesting video. Furthermore, the impact of the quality (bandwidth and delay) of the link connecting the remote server to the LAN is investigated and the tests were performed in various bandwidth/delay (15Mbps/15ms, 10Mbps/15ms and 15Mbps/65ms) settings as presented in Table 5-17.

Test Case Impact investigation	Segment Duration	Server-DAV Gateway		Request intervals	Segments Requested
		Bandwidth	Delay		
TC1 Base Case	6 s	15Mbps	15ms	1 – 90 s	16, 25 and 33
TC2 Timing	6 s	15Mbps	15ms	2, 4, 6 s	16, 25 and 33
TC3 Delay	6 s	15Mbps	65ms	1 – 90 s	16 and 33
TC4 Bandwidth	6 s	10Mpbs	15ms	1 – 90 s	16 and 33
TC5 Segment Length	10 s	15Mbps	15ms	1 – 90 s	10, 15 and 20

Table 5-17: DAV Evaluation Test Cases

5.3.4.1 TC1 Base Case - Results and Analysis

The tests were performed with random initial requests for video. The task is to evaluate DAV in a setting where there are no stalls for rebuffering, however, due to the nature of DASH adaptation and the link constraints the average bitrate of the requested segments is frequently reduced to the minimum bitrate of 500 kbps. The clients request the video under the settings given in Table 5-18. The results are compared in terms of the evaluation metrics described in Section 5.3.3. The results are presented in Figure 5-21 to Figure 5-24 (for 16 and 33 segments). Additionally, a representative subset of the evaluation results is provided in tabular form.

Property	Segment Duration	Server-Gateway		Initial Request Time	Video Duration	Segments Requested
		Bandwidth	Delay			
Value	6 s	15Mbps	15ms	1 – 90 s	96, 150 & 198s	16, 25 and 33

Table 5-18: TC1 Base Case Test Setting

The evaluation results indicate that the join time is significantly reduced (on average to 0.59 seconds) when DAV is deployed as shown in Figure 5-21 and Table 5-19. In this setting, there is no statistically significant difference between Scenario 2 and 3 (for α=0.05, P=0.98 for 16 segments, P=0.88 for 25 segments and P=0.88 for 33 segments), as the differences between the two approaches are applied from the second segment of the requested video.

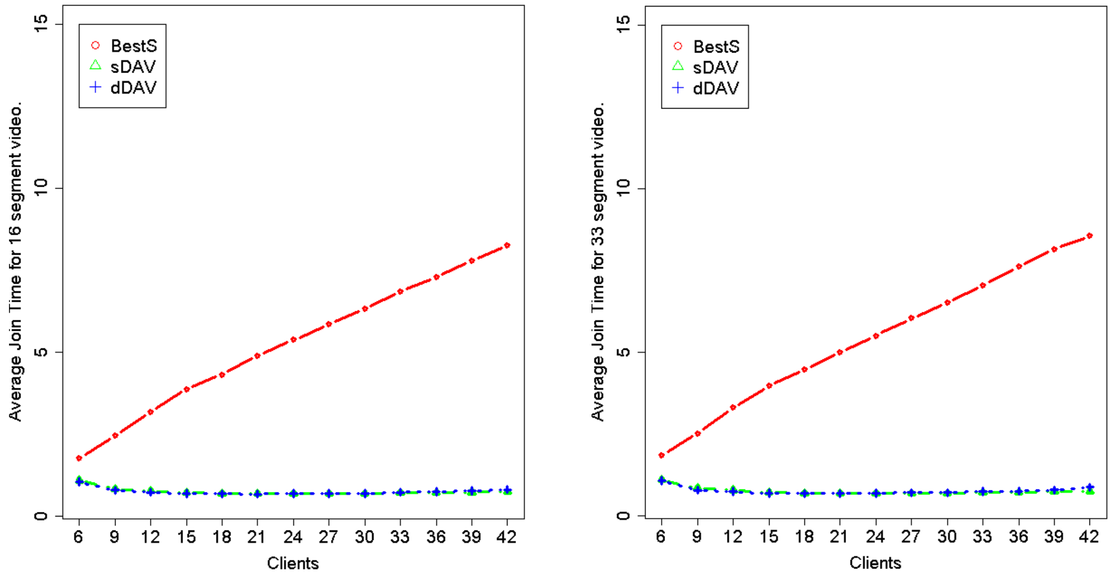

Figure 5-21: TC1 Base Case Average Join Time (seconds)

Number of Segments	Number of Clients	Scenario1 Best (seconds)	Scenario2 sDAV (seconds)	Scenario3 dDAV (seconds)	sDAV Reduction (%)	dDAV Reduction (%)	dDAV vs sDAV
16	6	1.78	1.09	1.06	38.77	40.67	3.10
	24	5.37	0.69	0.69	87.15	87.08	-0.57
	42	8.25	0.75	0.82	90.88	90.08	-8.77
25	6	1.85	1.11	1.08	40.19	41.74	2.60
	24	5.51	0.70	0.70	87.35	87.30	-0.35
	42	8.55	0.75	0.87	91.19	89.77	-16.09
33	6	1.85	1.11	1.08	40.19	41.74	2.60
	24	5.51	0.70	0.70	87.35	87.30	-0.35
	42	8.55	0.75	0.87	91.19	89.77	-16.09

Table 5-19: TC1 Base Case Average Join Time

Utilisation of local nodes as content providers results in a significantly higher average bitrate per client. It can be observed that Scenario 3 (dDAV) continues to outperform Scenario 2 (sDAV) as indicated in Figure 5-22 and Table 5-20. For instance, when 24 clients request 16 segment video, a 73.13% increase in average bitrate is recorded when using sDAV and 84.86% when dDAV is deployed which represents a significant improvement in video quality. However, in the case where a high number of clients initiate connections within a short period of time (e.g. more than 35 requests over 90 seconds) the average bitrate per segment is reduced with the increasing number of requests. The requested segment bitrate decreases due to the competing traffic on the Server–LAN link, and consequently, DAV Clients offer content of lower bitrate.

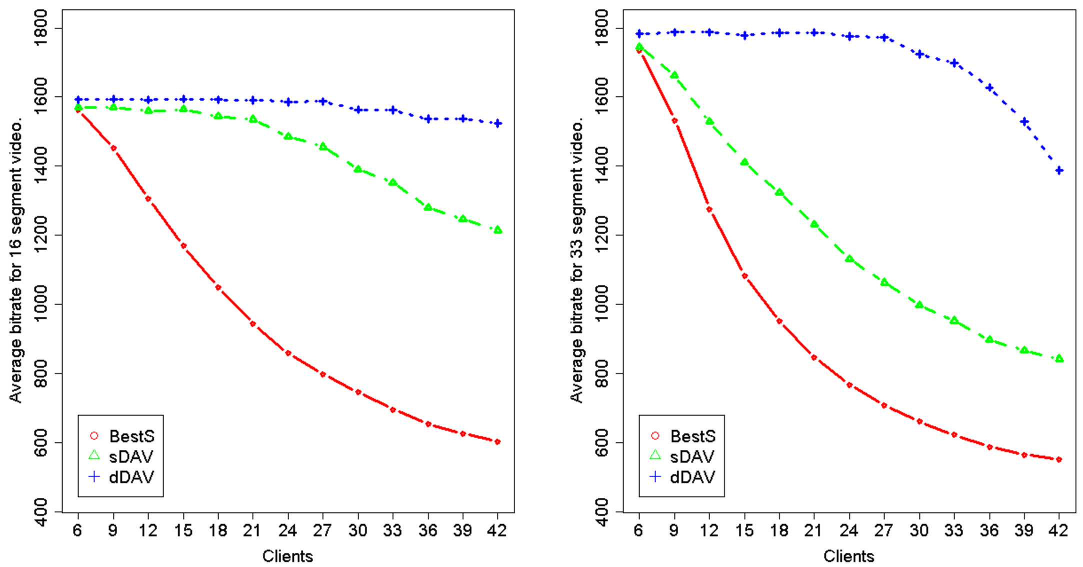

Figure 5-22: TC1 Base Case Average Bitrate (kbps)

Number of Segments	Number of Clients	Scenario1 Best (kbps)	Scenario2 sDAV (kbps)	Scenario3 dDAV (kbps)	sDAV Increase (%)	dDAV Increase (%)	dDAV vs sDAV Increase
16	6	1562.08	1569.30	1592.21	0.46	1.93	1.46
	24	858.10	1485.67	1586.32	73.13	84.86	6.78
	42	601.42	1214.66	1523.96	101.96	153.39	25.46
25	6	1665.42	1666.08	1689.25	0.04	1.43	1.39
	24	788.42	1273.18	1686.04	61.49	113.85	32.43
	42	562.92	956.14	1496.14	69.85	165.78	56.48
33	6	1735.00	1745.69	1783.44	0.62	2.79	2.16
	24	766.57	1131.05	1775.83	47.55	131.66	57.01
	42	550.73	841.65	1387.65	52.82	151.97	64.87

Table 5-20: TC1 Base Case Average Bitrate

In this setting, the link bandwidth and the client side adaptation reduce stalls and consequently rebuffering rate and ratio. While players stop for rebuffering the degree of interruption is low, and it can be assumed that playout is without noticeable stops. Slightly longer delays are present in Scenario 1 (BestS) when 42 clients request videos longer than 50 seconds. A summary of results for 42 clients is provided in Table 5-21. When short video clips (e.g. 16 segments) are requested, the BestS scenario results in a shorter rebuffering ratio compared to DAV scenarios.

However, the stalls introduced by DAV are short.

Metrics	Number of Segments	Number of Clients	Scenario1 BestS	Scenario2 sDAV	Scenario3 dDAV	sDAV Reduction (%)	dDAV Reduction (%)	dDAV vs sDAV
Rebuffering Ratio	16	42	0.0878	0.3750	0.2780	-327.12	-216.61	25.87
	25	42	1.3624	0.8072	0.4994	40.75	63.34	38.13
	33	42	1.9871	0.7637	0.5387	61.57	72.89	29.46
Rebuffering Rate	16	42	0.1821	0.0011	0.0010	99.39	99.46	12.18
	25	42	2.6167	0.0088	0.0009	99.67	99.97	90.17
	33	42	3.6792	0.0291	0.0007	99.21	99.98	97.53

Table 5-21: TC1 Base Case Rebuffering Ratio and Rate

The most significant improvement is in the average bitrate of the video segments downloaded at the client side. This improvement is a result of the increased amount of video data received by the client, most of which originates from within the LAN (DAV Clients) as indicated in Figure 5-23 and Table 5-22. As the number of requests increases, the quality of segments downloaded by DAV Clients decreases. This is reflected in the reduced quantity of video content found locally, especially when there are more than 35 requests issued.

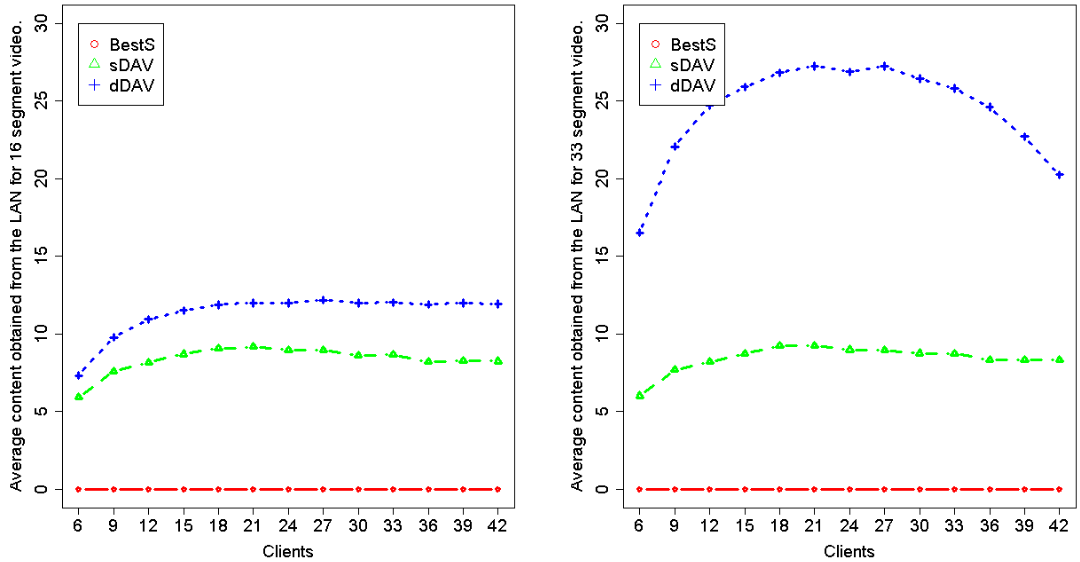

Figure 5-23: TC1 Base Case Average Content (Mb) from DAV Clients

Number of Segments	Number of Clients	Scenario2 sDAV (Mb)	Scenario3 dDAV (Mb)	dDAV vs sDAV Increase (%)
16	6	5.95	7.33	23.22
	24	8.97	11.99	33.63
	42	8.26	11.94	44.56
25	6	6.01	12.38	105.86
	24	9.00	20.26	125.19
	42	8.32	17.47	110.09
33	6	6.01	16.52	174.71
	24	9.00	26.87	198.69
	42	8.32	20.25	143.46

Table 5-22: TC1 Base Case Average Content from DAV Clients

DAV deployment (Scenarios 2 and 3) reduces load on the Remote Server - LAN Gateway link. The reduction in the link utilisation is highest when the number of clients range from 12 to 33 as indicated in Figure 5-24 and Table 5-23.

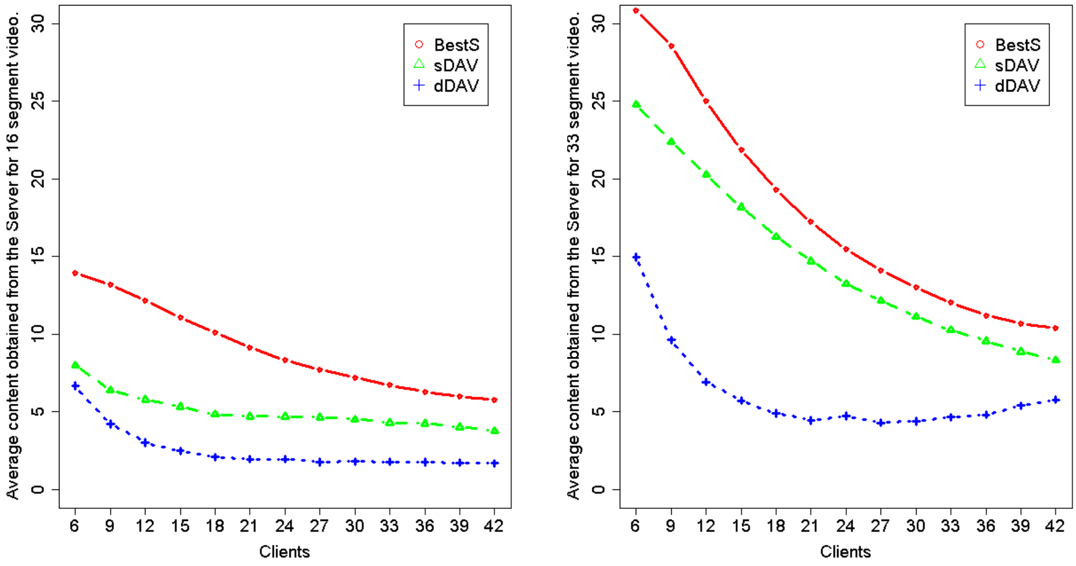

Figure 5-24: TC1 Base Case Average Content (Mb) from Remote Server

Number of Segments	Number of Clients	Scenario1 Best (Mb)	Scenario2 sDAV (Mb)	Scenario3 dDAV (Mb)	sDAV Reduction (%)	dDAV Reduction (%)	dDAV vs sDAV
16	6	13.93	8.01	6.67	42.47	52.13	16.79
	24	8.33	4.71	1.96	43.44	76.41	58.28
	42	5.76	3.76	1.69	34.78	70.66	55.02
25	6	23.01	17.10	11.25	25.67	51.10	34.21
	24	12.23	10.07	3.46	17.66	71.71	65.64
	42	8.27	6.43	3.84	22.21	53.60	40.36
33	6	30.83	24.78	14.97	19.63	51.45	39.59
	24	15.45	13.23	4.72	14.35	69.42	64.30
	42	10.37	8.35	5.79	19.45	44.13	30.65

Table 5-23: TC1 Base Case Content from Remote Server

In conclusion, the evaluation results suggest that when clients come online at random intervals, the deployment of local content aware solutions significantly reduces join time and improves the bitrate of the played video. Furthermore, the utilisation of the server-campus network link is reduced. It should be noted that the best results are achieved when 9 to 33 requests are randomly issued within the first 90 seconds.

5.3.4.2 TC2 Request Timing Impact - Results and Analysis

The same scenarios are tested with clients requesting the video clip with sequential initial request timing, in order to determine if DAV performance depends on the request timing. The setting parameters are given in Table 5-24. The segment duration is 6 seconds. The simulation results are compared in terms of the evaluation metrics given in Section 5.3.3. The results are presented in Figure 5-25 to Figure 5-34. Selected summary numeric values depicted in graphs are also provided in tabular form. Full results are omitted for conciseness.

Property	Segment Duration	Server-Gateway		Request intervals	Video Duration	Segments Requested
		Bandwidth	Delay			
Value	6 s	15Mbps	15ms	2, 4 & 6 s	96, 150 & 198s	16, 25 and 33

Table 5-24: TC2 Timing Test Setting

Figure 5-25 indicates average join times for 25 segment video, other charts are omitted for brevity. In sDAV and dDAV, where local content is made available to clients, the overall performance is enhanced as the join time decreases to an average of just 0.68 seconds per client. More importantly, the join time does not increase as subsequent clients come online and request video, as the first segment of the requested video is found locally. There is no significant statistical difference (α=0.05, P=0.788) between sDAV and dDAV in terms of join time. Significant reductions in join time (ranging from 54% to 93%) are observed regardless of the duration of video and the pacing of the sequential requests which is in line with TC1 (random request intervals). The evaluation results are summarised in Table 5-25 which contains average join times for the cases with 6, 24 and 42 clients only.

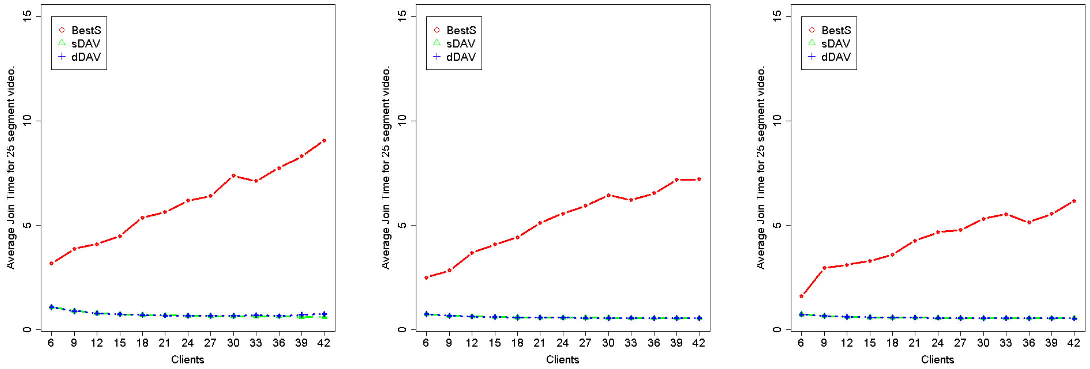

Figure 5-25: TC2 Timing Average Join Time (seconds) 150s (2, 4 and 6 s Request Intervals)

Request Interval (seconds)	Number of Segments	Number of Clients	BestS (seconds)	sDAV (seconds)	dDAV (seconds)	sDAV Reduction (%)	dDAV Reduction (%)
2	16	6	3.18	1.08	1.08	65.97	66.02
		24	6.18	0.66	0.67	89.27	89.18
		42	8.71	0.62	0.69	92.87	92.08
	25	6	3.18	1.08	1.08	65.97	66.02
		24	6.18	0.66	0.67	89.27	89.18
		42	9.05	0.62	0.74	93.16	91.82
	33	6	3.18	1.08	1.08	65.97	66.02
		24	6.18	0.66	0.67	89.27	89.18
		42	9.05	0.62	0.74	93.16	91.82

4	16	6	2.50	0.75	0.75	70.13	70.07
		24	4.88	0.58	0.57	88.08	88.30
		42	6.15	0.55	0.55	91.05	91.10
	25	6	2.50	0.75	0.75	70.13	70.07
		24	5.56	0.58	0.57	89.48	89.70
		42	7.20	0.56	0.55	92.25	92.37
	33	6	2.50	0.75	0.75	70.13	70.07
		24	5.56	0.58	0.57	89.48	89.70
		42	8.12	0.56	0.61	93.11	92.47
6	16	6	1.60	0.74	0.73	54.06	54.17
		24	4.06	0.56	0.56	86.15	86.10
		42	4.74	0.54	0.54	88.68	88.63
	25	6	1.60	0.74	0.73	54.06	54.17
		24	4.66	0.57	0.57	87.87	87.86
		42	6.17	0.55	0.54	91.14	91.21
	33	6	1.60	0.74	0.73	54.06	54.17
		24	4.89	0.57	0.57	88.41	88.40
		42	6.45	0.55	0.54	91.47	91.56

Table 5-25: TC2 Timing Average Join Time

As with TC1 (Base Case), the three scenarios are compared in terms of average segment bitrate (Figure 5-26 - Figure 5-28) and the simulation results suggest that the best results are achieved when the initial requests are paced at more than 2 seconds apart.

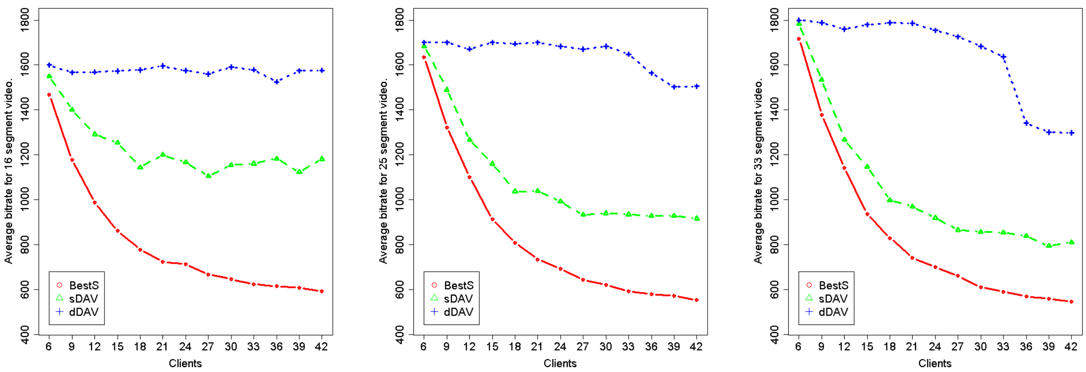

Figure 5-26: TC2 Timing Average Bitrate (kbps) 2 second Request Intervals

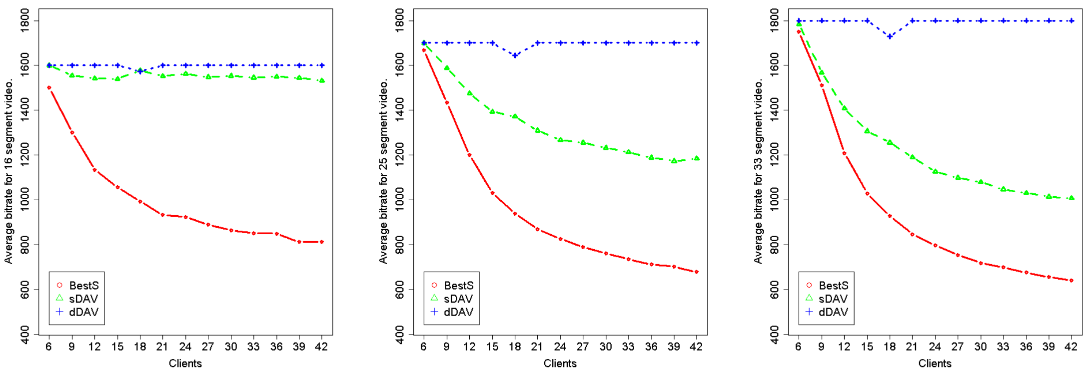

Figure 5-27: TC2 Timing Average Bitrate (kbps) 4 second Request Intervals

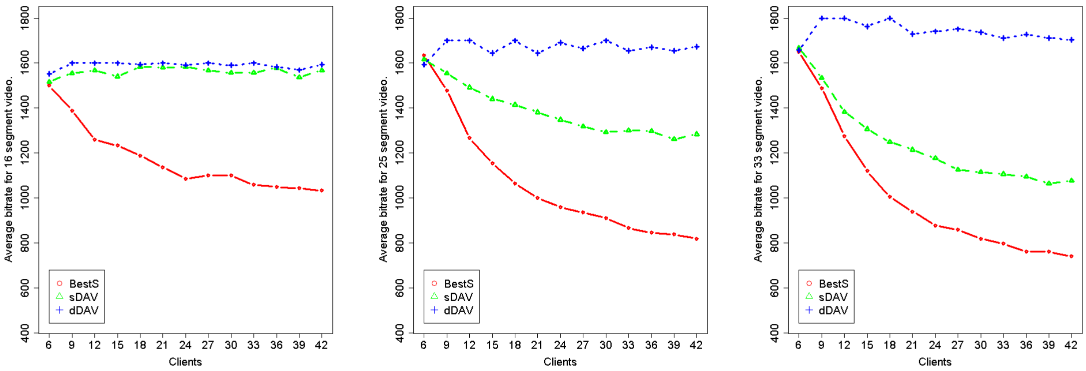

Figure 5-28: TC2 Timing Average Bitrate (kbps) 6 second Request Intervals

The longer intervals produce consistently significant improvements (as indicated in Figure 5-27 and Figure 5-28) which are more pronounced as the number of concurrent requests increases. The change in the average bitrate range from a decrease of 3% to an increase of 181% as indicated in Table 5-26 which contains average bitrates for the cases with 6, 24 and 42 clients.

Request Interval (seconds)	Number of Segments	Number of Clients	BestS (kbps)	sDAV (kbps)	dDAV (kbps)	sDAV Increase (%)	dDAV Increase (%)	dDAV vs sDAV Increase
2	16	6	1466.67	1550.00	1600.00	5.68	9.09	3.23
		24	712.08	1166.67	1575.00	63.84	121.18	35.00
		42	592.14	1180.71	1576.19	99.40	166.18	33.49
	25	6	1633.33	1683.33	1700.00	3.06	4.08	0.99
		24	692.08	992.08	1683.33	43.35	143.23	69.68
		42	552.14	916.19	1505.00	65.93	172.57	64.27
	33	6	1716.67	1783.33	1800.00	3.88	4.85	0.93
		24	698.75	919.17	1754.17	31.54	151.04	90.84
		42	545.95	810.00	1297.43	48.36	137.65	60.18
4	16	6	1500	1600	1600	6.67	6.67	0.00
		24	922.5	1562.5	1600	69.38	73.44	2.40
		42	813.57	1530.95	1600	88.18	96.66	4.51
	25	6	1666.67	1700	1700	2.00	2.00	0.00
		24	825.83	1266.67	1700	53.38	105.85	34.21
		42	677.86	1184.76	1700	74.78	150.79	43.49
	33	6	1750	1783.33	1800	1.90	2.86	0.93
		24	797.5	1125	1800	41.07	125.71	60.00
		42	640	1007.38	1800	57.40	181.25	78.68
6	16	6	1500.00	1516.67	1551.67	1.11	3.44	2.31
		24	1085.42	1583.33	1591.67	45.87	46.64	0.53
		42	1032.38	1566.67	1592.86	51.75	54.29	1.67
	25	6	1633.33	1616.67	1593.33	-1.02	-2.45	-1.44
		24	958.75	1347.50	1691.67	40.55	76.45	25.54
		42	818.10	1283.33	1672.38	56.87	104.42	30.32
	33	6	1650.00	1666.67	1658.33	1.01	0.51	-0.50
		24	876.25	1176.25	1741.25	34.24	98.72	48.03
		42	739.29	1076.43	1703.57	45.60	130.43	58.26

Table 5-26: TC2 Timing Average Bitrate

The three scenarios are compared in terms of rebuffering ratio and rate and the simulation results indicate that very short stalls for rebuffering occur in the case of requests paced at 2 seconds (indicated in Table 5-27). The link to the remote server is sufficiently provisioned so that there are no rebuffering events for 4 and 6 second paced initial requests.

Metric	Request Interval (seconds)	Number of Segments	Number of Clients	BestS	sDAV	dDAV
Rebuffering Ratio	2	16	42	0	0	0
		25	42	0.7	0	0
		33	42	1.4	0	0
Rebuffering Rate	2	16	42	0.12	0	0
		25	42	2.70	0	0
		33	42	3.54	0.12	0

Table 5-27: TC2 Timing Average Rebuffering Ratio and Rate

The amount of content downloaded from the remote server reduces as the gap between subsequent requests increases from 2 seconds, as indicated in Figure 5-29 to Figure 5-31. It can be concluded that DAV deployment significantly reduces the utilisation of the Server-DAV Gateway link and consequently the associated link utilisation costs are reduced as well. Furthermore, Scenario 3 significantly outperforms both Scenario 1 and 2 as shown in the last column of Table 5-28. This table provides values for the average volume (Mb) of content downloaded per client from the remote server for 6, 24 and 42 clients.

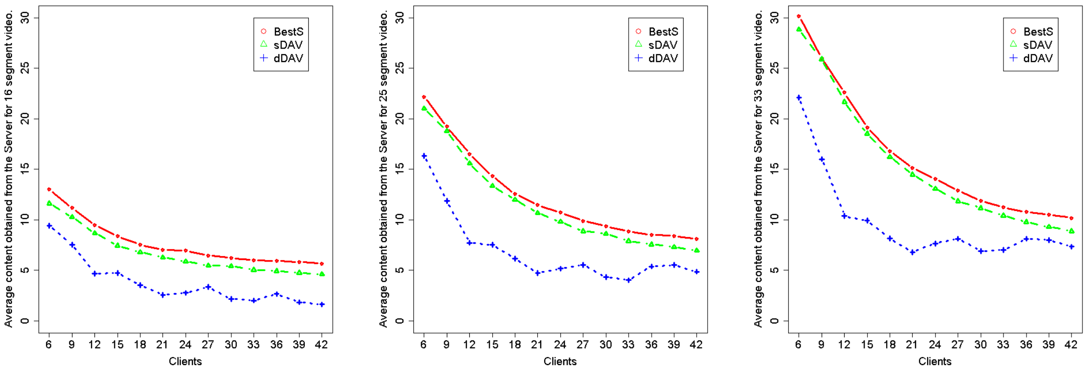

Figure 5-29: TC2 Timing Average Remote Content (Mb) 2 second Request Intervals

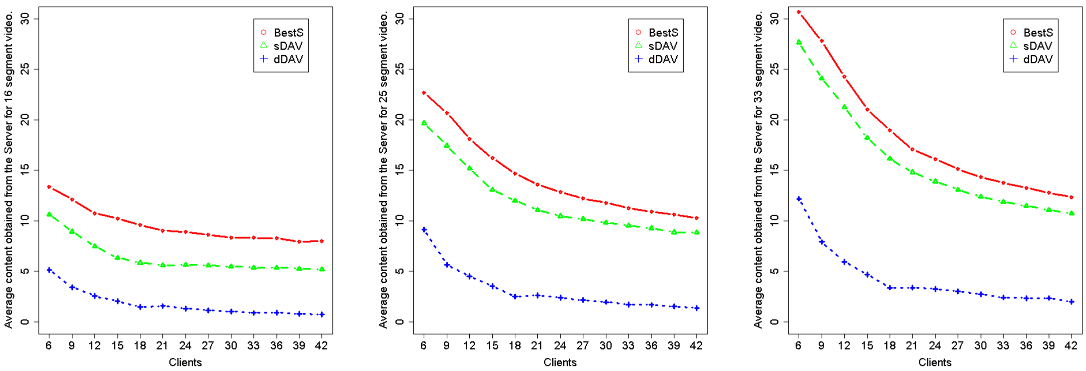

Figure 5-30: TC2 Timing Average Remote Content (Mb) 4 second Request Intervals

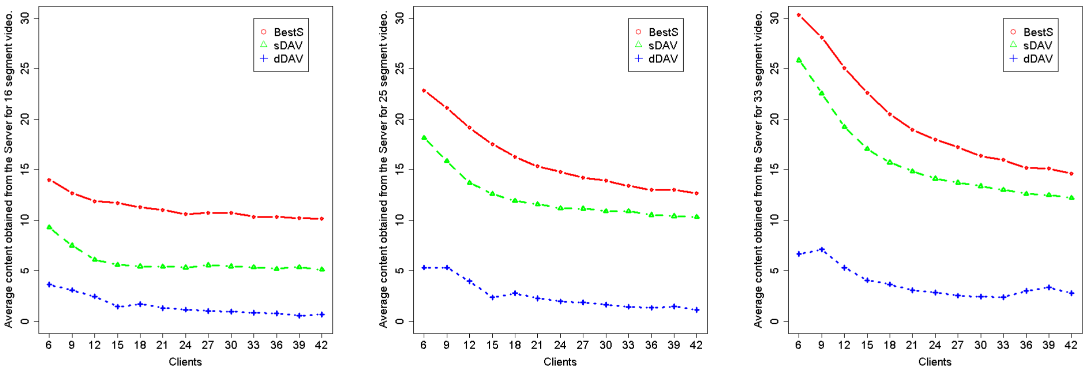

Figure 5-31: TC2 Timing Average Remote Content (Mb) 6 second Request Intervals

Request Interval (seconds)	Number of Segments	Number of Clients	BestS (Mb)	sDAV (Mb)	dDAV (Mb)	sDAV Reduction (%)	dDAV Reduction (%)	dDAV vs sDAV
2	16	6	13.00	11.62	9.43	10.64	27.44	18.79
		24	6.96	5.88	2.78	15.57	60.08	52.72
		42	5.67	4.63	1.63	18.35	71.19	64.71
	25	6	22.17	21.00	16.33	5.26	26.32	22.22
		24	10.73	9.80	5.18	8.62	51.72	47.16
		42	8.09	6.95	4.88	14.07	39.68	29.80
	33	6	30.17	28.83	22.10	4.42	26.74	23.35
		24	14.04	13.08	7.66	6.82	45.46	41.46
		42	10.19	8.88	7.34	12.87	27.98	17.35
4	16	6	13.33	10.63	5.15	20.25	61.37	51.57
		24	8.92	5.66	1.32	36.52	85.19	76.67
		42	7.97	5.19	0.74	34.93	90.77	85.82
	25	6	22.67	19.67	9.15	13.24	59.63	53.47
		24	12.83	10.47	2.40	18.38	81.33	77.12
		42	10.28	8.86	1.38	13.78	86.61	84.47
	33	6	30.67	27.67	12.18	9.78	60.27	55.96
		24	16.08	13.91	3.26	13.52	79.72	76.54
		42	12.36	10.72	1.98	13.27	83.95	81.49
6	16	6	14.00	9.30	3.67	33.57	73.81	60.57
		24	10.62	5.34	1.17	49.68	89.01	78.16
		42	10.14	5.12	0.68	49.49	93.29	86.72
	25	6	22.83	18.17	5.33	20.44	76.64	70.64
		24	14.79	11.18	1.98	24.45	86.61	82.28
		42	12.67	10.34	1.13	18.40	91.05	89.03
	33	6	30.33	25.83	6.67	14.84	78.02	74.19
		24	18.00	14.14	2.85	21.46	84.14	79.81
		42	14.62	12.23	2.79	16.34	80.92	77.20

Table 5-28: TC2 Timing Average Content from Remote Server

The average volume of content per client downloaded from the campus network (DAV Clients) is indicated in Figure 5-32 - Figure 5-34. In line with the previous results, the quantity of "reused" content increases with increasing intervals between initial client requests. Summary values for 6, 24 and 42 clients are provided in Table 5-29.

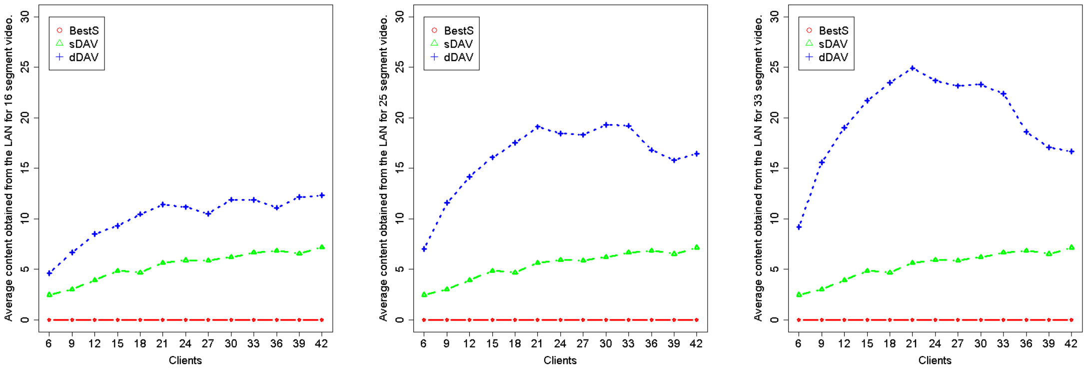

Figure 5-32: TC2 Average Content (Mb) from DAV Clients for 2 second Request Intervals

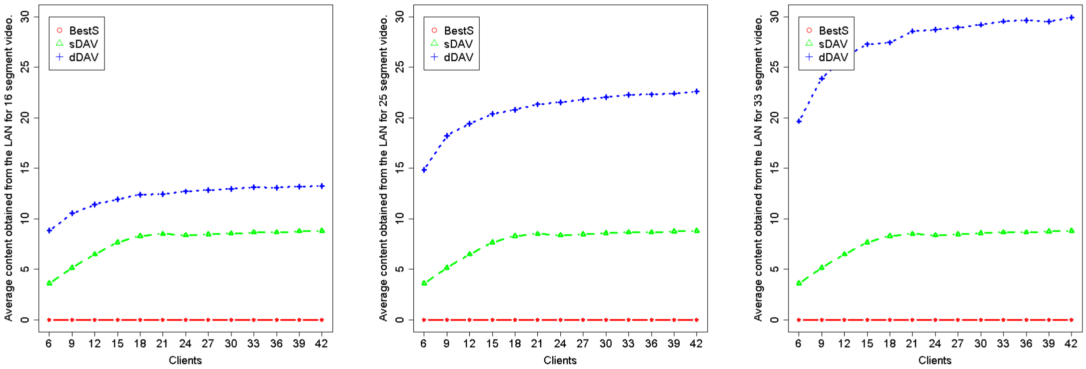

Figure 5-33: TC2 Average Content (Mb) from DAV Clients for 4 second Request Intervals

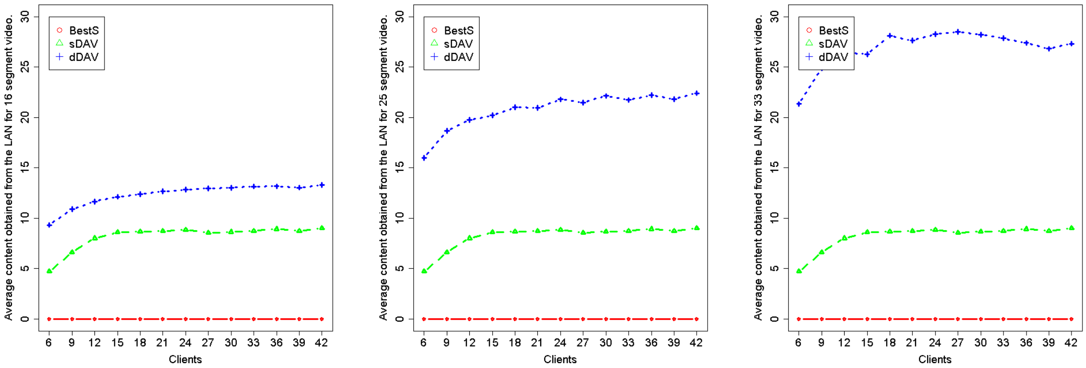

Figure 5-34: TC2 Average Content (Mb) from DAV Clients for 6 second Request Intervals

Request Interval (seconds)	Number of Segments	Number of Clients	sDAV (Mb)	dDAV (Mb)	dDAV vs sDAV Increase
2	16	6	2.48	4.63	86.58
		24	5.92	11.17	88.80
		42	7.20	12.32	71.21
	25	6	2.48	7.03	183.22
		24	5.94	18.46	210.66
		42	7.16	16.43	129.53
	33	6	2.48	9.20	270.47
		24	5.94	23.67	298.32
		42	7.16	16.69	133.09

4	16	6	3.62	8.83	144.24
		24	8.40	12.71	51.36
		42	8.82	13.26	50.42
	25	6	3.62	14.83	310.14
		24	8.40	21.54	156.58
		42	8.82	22.62	156.55
	33	6	3.62	19.67	443.78
		24	8.40	28.71	241.94
		42	8.82	29.90	239.18
6	16	6	4.73	9.33	97.18
		24	8.85	12.83	45.01
		42	9.00	13.31	47.88
	25	6	4.73	16.00	238.03
		24	8.85	21.79	146.23
		42	9.00	22.40	148.89
	33	6	4.73	21.33	350.70
		24	8.85	28.25	219.21
		42	9.00	27.31	203.47

Table 5-29: TC2 Timing Content from DAV Clients

In summary, the presented tests were conducted in a setting where the initial requests are sequentially paced and the link between the hosting server and the campus network is provisioned to eliminate stops for buffering (stalls). The results indicate that DAV deployment significantly increases the quality of video playout while reducing the strain on the Server-campus network link.

5.3.4.3 TC3 Link Delay Impact - Results and Analysis

Similar tests were performed in order to determine how DAV performance is influenced by the delay on the Server-DAV Gateway link. The clients request at random intervals the same video under the settings given in Table 5-30. The number of clients varies from 6 to 42 in increments of 6. As before the scenarios are compared in terms of the evaluation metrics given in Section 5.3.3. The results are presented in graphs in Figure 5-35 - Figure 5-40. Tabular data is omitted for brevity.

Property	**Segment Duration**	**Server-Gateway**		**Random Request Time**	**Video Duration**	**Segments Requested**
		Bandwidth	**Delay**			
Value	6 s	15Mbps	65ms	1 – 90 s	96 and 198s	16 and 33

Table 5-30: TC3 Delay Test Setting

TCP's sensitivity to delay is reflected in the increased join times in the BestS scenario (when all segments are requested from the remote server). For example, in the case of 24 clients requesting 33 segments of content, the join time has increased from the TC1 average of 5.51 seconds (TC1, Figure 5-21) to 9.15 seconds (Figure 5-35). The join times also increased in sDAV (from 0.70 to 1.06 seconds) and in dDAV (from 0.70 to 1.04 seconds).

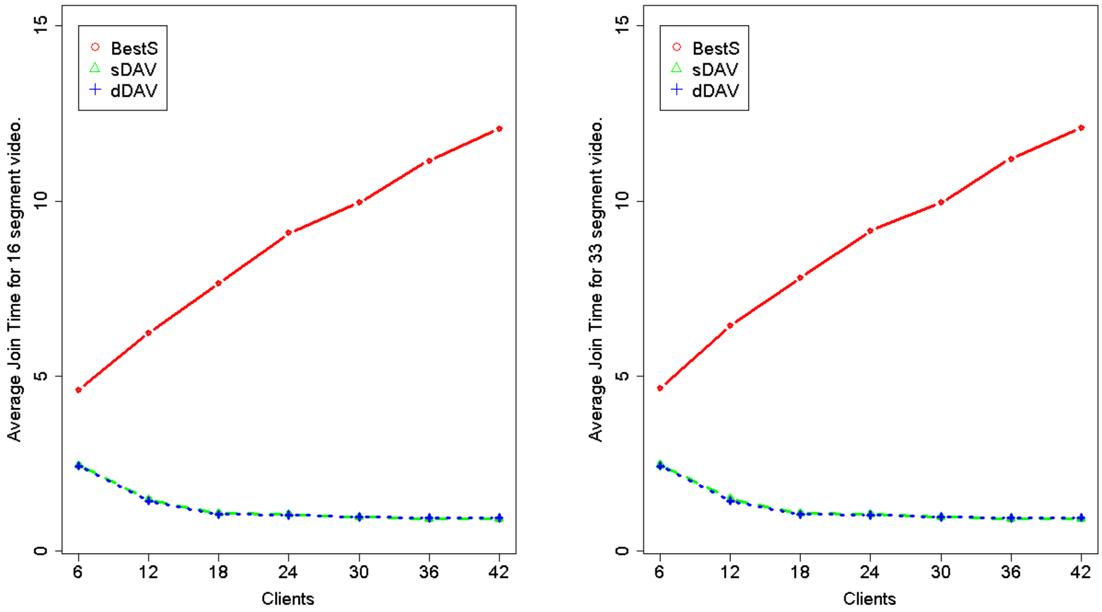

Figure 5-35: TC3 Delay Average Join Time (seconds)

The improvement in the average bitrate per client requesting is less than in the Base Case (TC1) setting due to the Server-DAV Gateway link constraints, however DAV deployment nonetheless produces improved results (Figure 5-36). For example, when 24 clients issue requests for a 198s video, the average bitrate in sDAV (891.56 kbps) has increased by 35.7% when compared to BestS (657.04 kbps), while the bitrate has doubled in dDAV (1322.74 kbps). Compared to the base case (TC1), the average bitrate has been reduced by 21% in sDAV (from 1131.05 kbps) and by 25.5% in dDAV (from 1775.83 kbps).

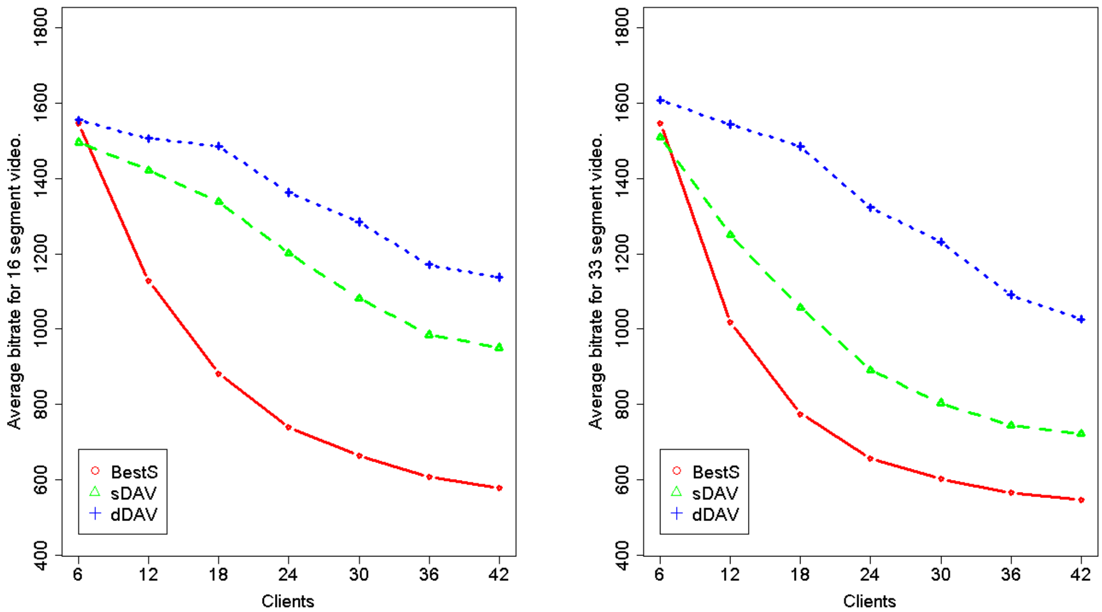

Figure 5-36: TC3 Delay Average Bitrate (kbps)

A significant increase in rebuffering ratio and rate are observed in this setting as indicated in Figure 5-37 and Figure 5-38 respectively. sDAV and dDAV introduce a limited degree of rebuffering for 6 clients requesting video (due to the limited number of DAV Clients providing content locally). For example, when 24 clients request a 198s video, the average rebuffering ratios per client are as follows: 2.0695 in BestS scenario, 0.6309 for sDAV and 0.5135 for dDAV. Similarly, the average rebuffering rates per client are as follows: 4.044 for BestS, 0.1217 in sDAV setting and 0.0022 in dDAV setting. The depicted simulation results indicate that dDAV brings the most significant reduction in rebuffering time as the number of clients increases.

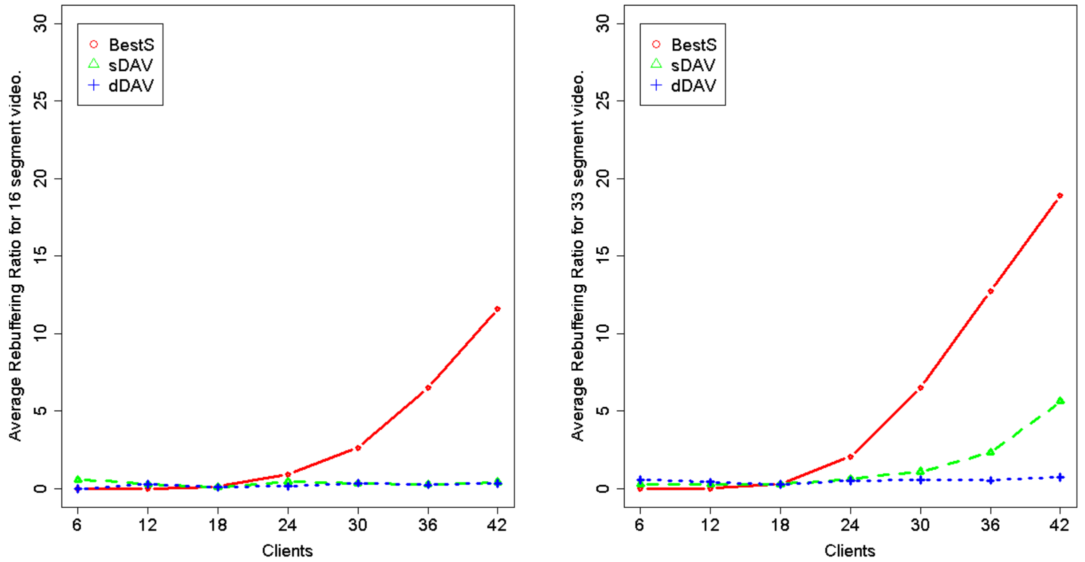

Figure 5-37: TC3 Delay Average Rebuffering Ratio

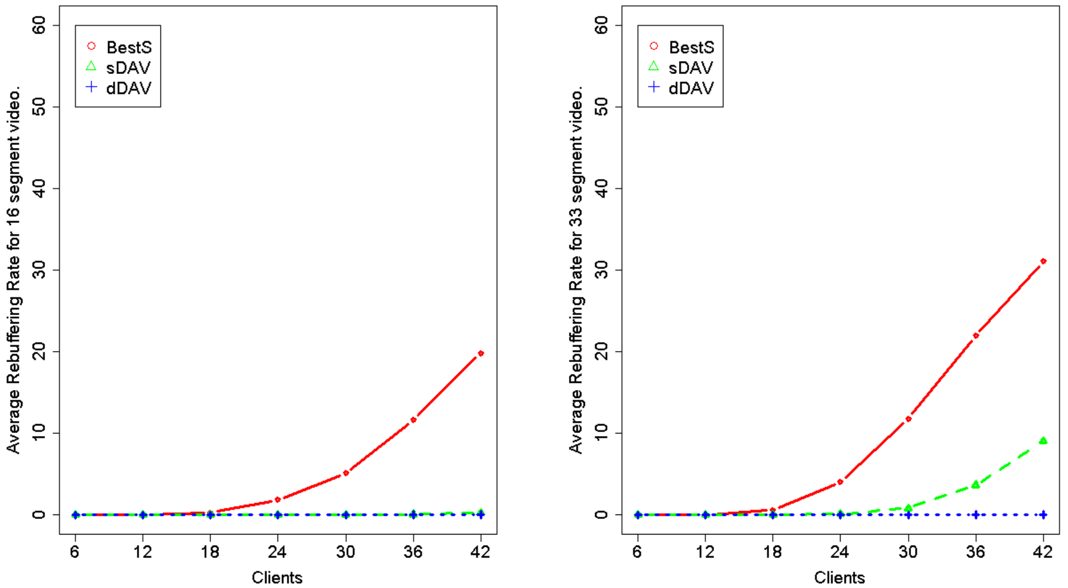

Figure 5-38: TC3 Delay Average Rebuffering Rate

Consistent with conclusions regarding the average bitrate, the quantity of content per client downloaded from the campus network (served by DAV Clients) is reduced compared to the base case (TC1), however DAV deployment still produces improvements as indicated in Figure

5-39. For example, when 24 clients request a 198s video, the average content downloaded per client in sDAV (7.13 Mb) and dDAV (18.45Mb) settings decreases when compared to the TC1 values (9.00 and 26.87Mb). The best results are achieved for higher numbers of clients.

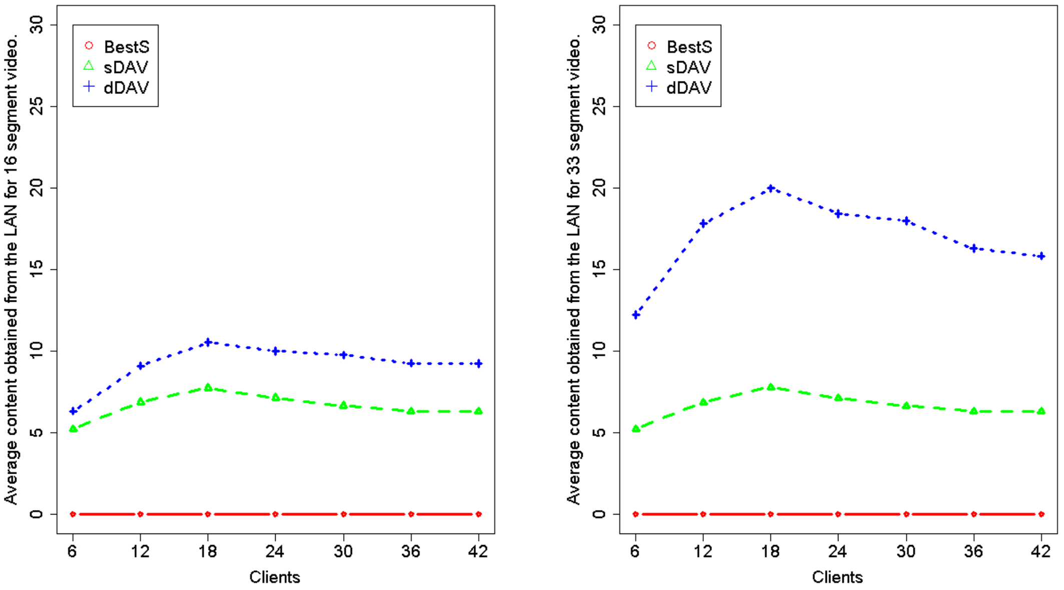

Figure 5-39: TC3 Delay Average Content (Mb) from DAV Clients

The volume of video data retrieved from the remote server (Figure 5-40) is lower for the BestS (from 15.45 to 12.82Mb) and sDAV (from 13.23 to 10.66Mb) scenarios, while there is an increase in the dDAV case (from 4.72 to 7Mb) on average for 24 clients requesting 33 segments of video content. Still dDAV requests the least amount of remotely stored content, when compared to the two other scenarios.

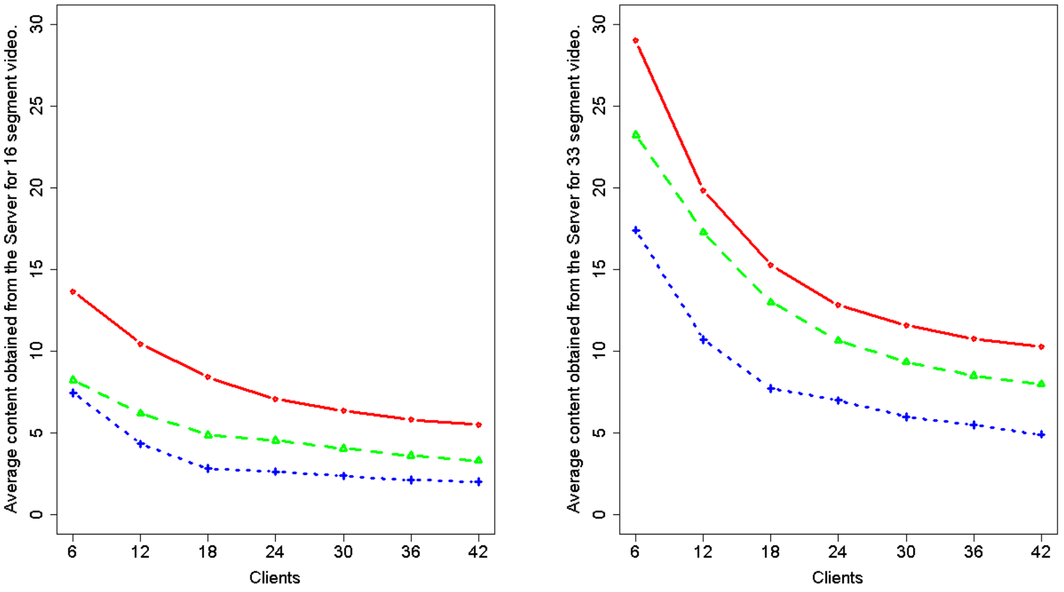

Figure 5-40: TC3 Delay Average Content (Mb) from Remote Server

In conclusion, a link with a longer delay affects the average bitrate as the quality (bitrate) of the segments downloaded from the server is lower. This also reflects on the volume of the content available locally (the segments do not arrive in time to be used by other nodes), which in turn reduces the average amount of content obtained from the campus network. Still, DAV deployment significantly improves overall video delivery. In this case, in line with previous findings, dDAV (Scenario 3) outperforms other scenarios.

5.3.4.4 TC4 Link Bandwidth Impact - Results and Analysis

These tests were performed with random initial request timing, in order to determine how DAV performance is influenced by the constraint bandwidth of the Server-DAV Gateway link. The number of clients requesting videos varies from 6 to 42 in increments of 6. The video is requested under the settings given in Table 5-31. The scenarios are compared in terms of the evaluation metrics given in Section 5.3.3. The results are presented in graphs in Figure 5-41 to Figure 5-46. Tabular data is not provided for brevity.

Property	Segment Duration	Server-Gateway Bandwidth	Server-Gateway Delay	Random Request Time	Video Duration	Segments Requested
Value	6 s	10Mbps	15ms	1 – 90 s	96 and 198s	16 and 33

Table 5-31: TC4 Bandwidth Test Setting

Increased join times can be observed in BestS scenario. For example, in the case of 24 clients requesting video of 33 segments, the average join time has increased from 5.51 seconds (TC1 Figure 5-21, Table 5-19) to 7.74 seconds (Figure 5-41). The join times also increased in other scenarios, from 0.70 seconds in TC1 to 0.79 seconds (sDAV) and 0.73 seconds (dDAV). There is no significant difference between two DAV scenarios.

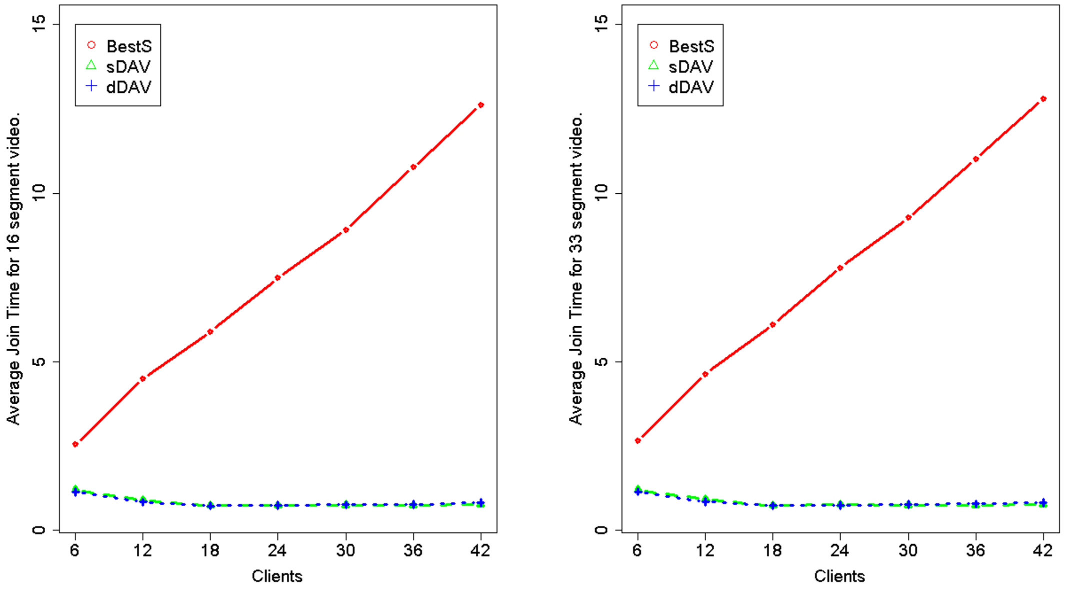

Figure 5-41: TC4 Bandwidth Average Join Time (seconds)

The average bitrate per client requesting is reduced with the decrease in the link bandwidth compared to the base case (TC1) setting, however DAV deployment still produces improved results (Figure 5-42). For example, when 24 clients request a ~200s video, the average bitrate in sDAV (867.39 kbps) has increased 49% when compared to BestS (581.1 kbps), while the bitrate almost tripled in dDAV (1502.84 kbps). Compared to the TC1, the average bitrate has been reduced by 23.3% in sDAV (from 1131.05 kbps) and by 15.4% in dDAV (from 1775.83 kbps).

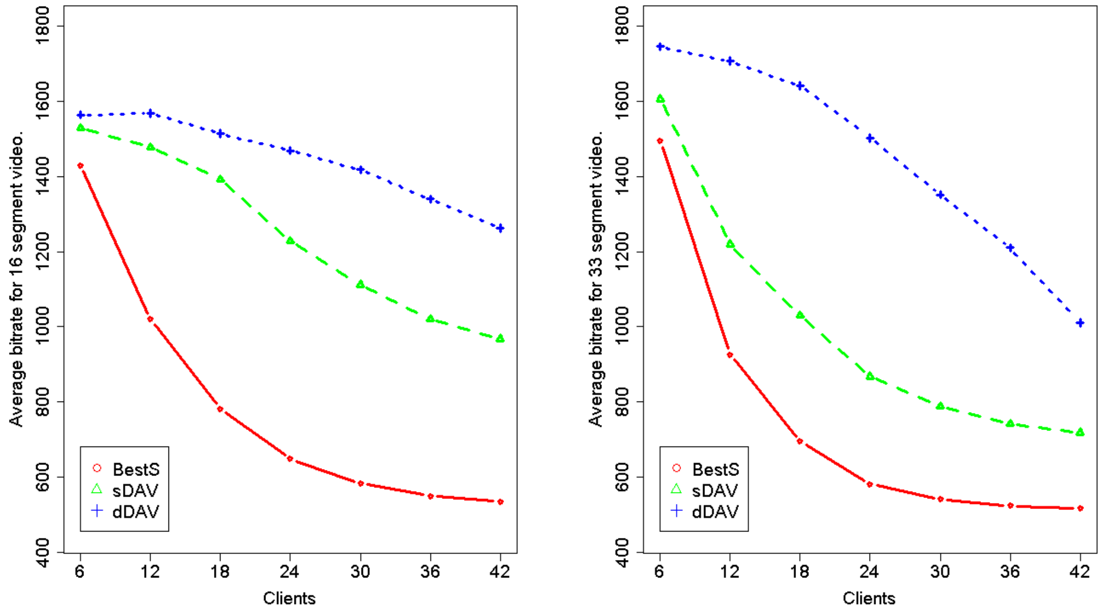

Figure 5-42: TC4 Bandwidth Average Bitrate (kbps)

A significant increase in rebuffering ratio and rate are observed in this setting as indicated in Figure 5-43 and Figure 5-44 respectively. For example, when 24 clients request a ~200s video, the average rebuffering rates per client are as follows: 0.193 in BestS, 0.0005 in sDAV and 0.0005 in dDAV. The depicted simulation results indicate that dDAV brings the most significant reduction in rebuffering with increasing number (25 or more) of clients requesting video.

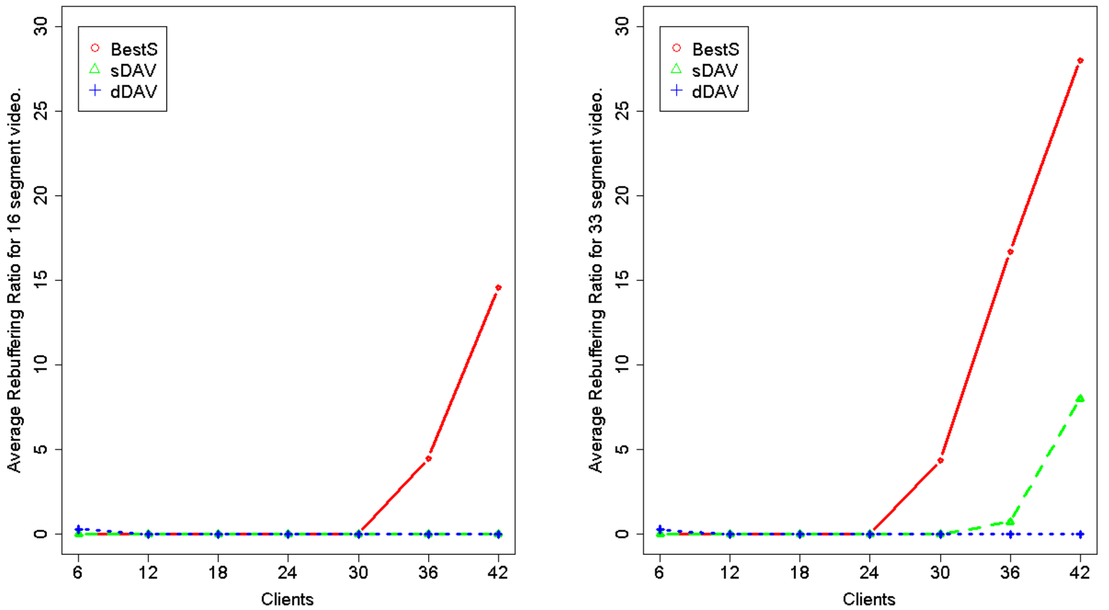

Figure 5-43: TC4 Bandwidth Average Rebuffering Ratio

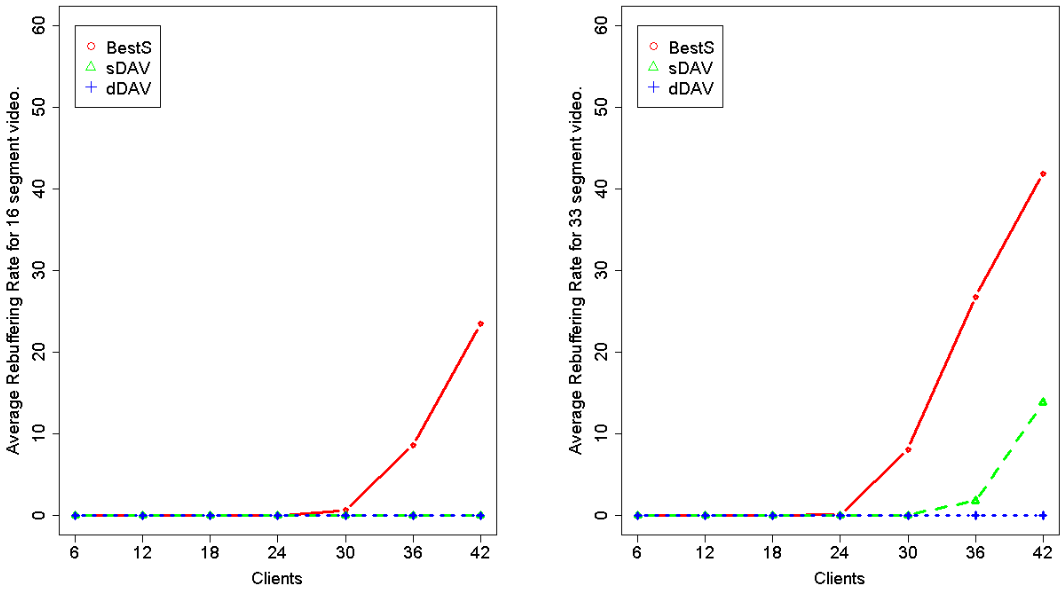

Figure 5-44: TC4 Bandwidth Average Rebuffering Rate

As expected, the volume of content per client downloaded from the campus network is reduced compared to the TC1 setting, however DAV deployment still produces improved results as indicated in Figure 5-45. For example, when 24 clients request a ~200s video, the average content downloaded per client in sDAV (7.66Mb) and dDAV (21.7Mb) decreases 15% and 19% respectively when compared to the TC1 values (9.0 and 26.87Mb).

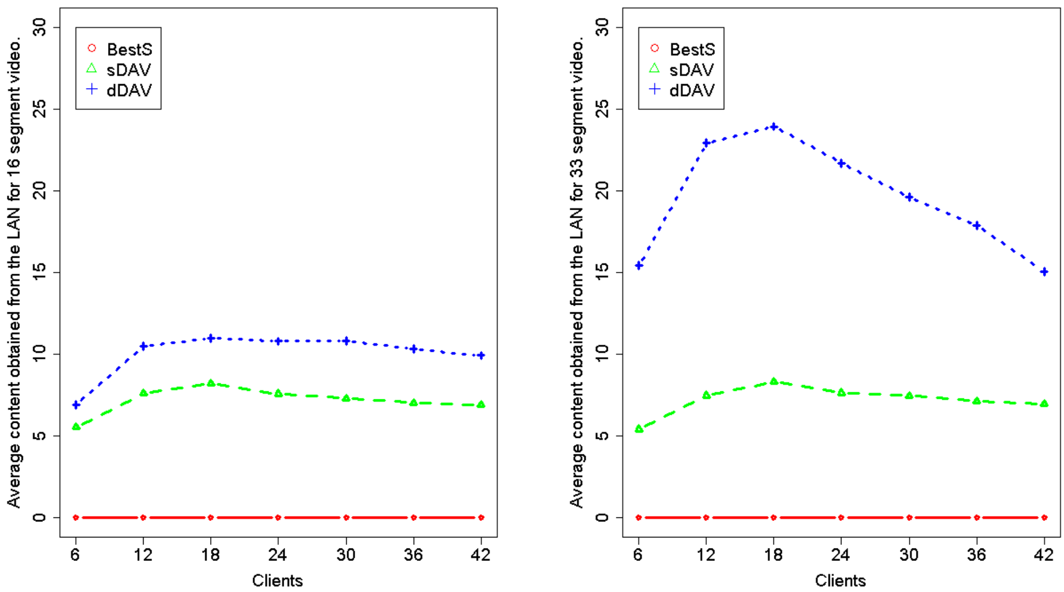

Figure 5-45: TC4 Bandwidth Average Content (Mb) from DAV Clients

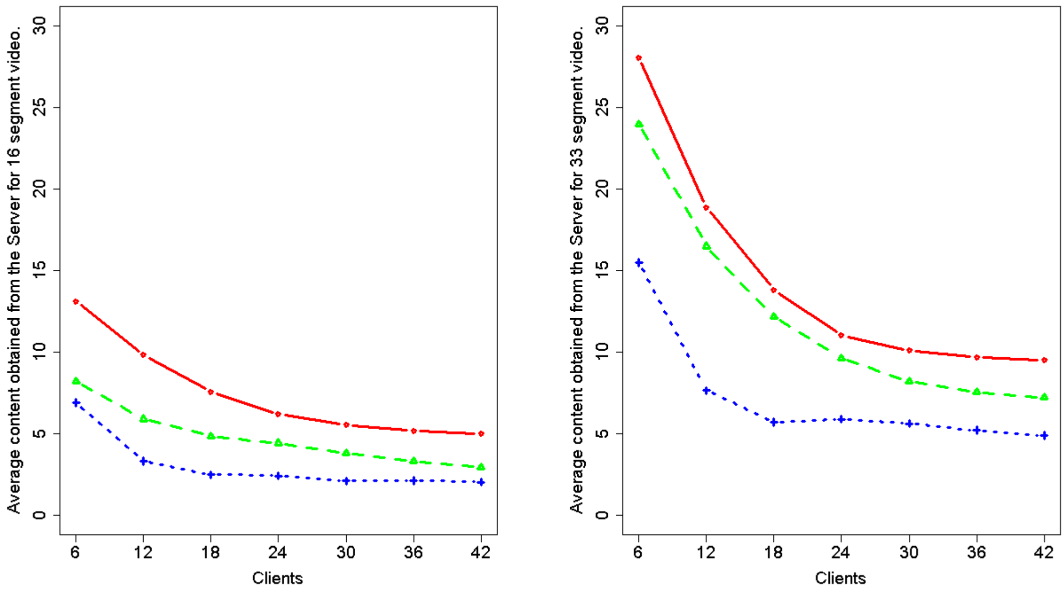

Figure 5-46: TC4 Bandwidth Average Content (Mb) from Remote Server

The quantity of video data retrieved from the remote Server (Figure 5-46) is lower for the BestS (from 15.45 to 11.05 Mb) and sDAV (from 13.23 to 9.65 Mb) scenarios, while there is an increase in dDAV (from 4.72 to 5.9 Mb) when compared to TC1 (24 clients requesting ~200s video) due to slower downloads by DAV clients. Still dDAV requests least amount of content, compared to other scenarios.

In conclusion, a link of lower bandwidth affects the average bitrate as the quality (bitrate) of the segments downloaded from the remote server is lower. This also reflects on the quality of the content available locally, however the results are improved compared to TC3 setting for longer videos. The evaluation results indicate that DAV deployment outperforms the typical DASH approach even in bandwidth constrained setting.

5.3.4.5 TC5 Segment Duration Impact - Results and Analysis

Similar tests were performed with random initial requests, in order to determine how DAV performance is influenced by the segment duration (10 seconds vs 6 seconds). Clients request the same video clip under the settings given in Table 5-32. As before, scenarios are compared in terms of the evaluation metrics given in Section 5.3.3. The results are presented in graphs in Figure 5-47 to Figure 5-50. Tabular data is omitted for brevity.

Property	Segment Duration	Server-Gateway Bandwidth	Server-Gateway Delay	Random Request Time	Video Duration	Segments Requested
Value	10 s	15Mbps	15ms	1 – 90 s	100 and 200s	10 and 20

Table 5-32: TC5 Segment Duration Test Setting

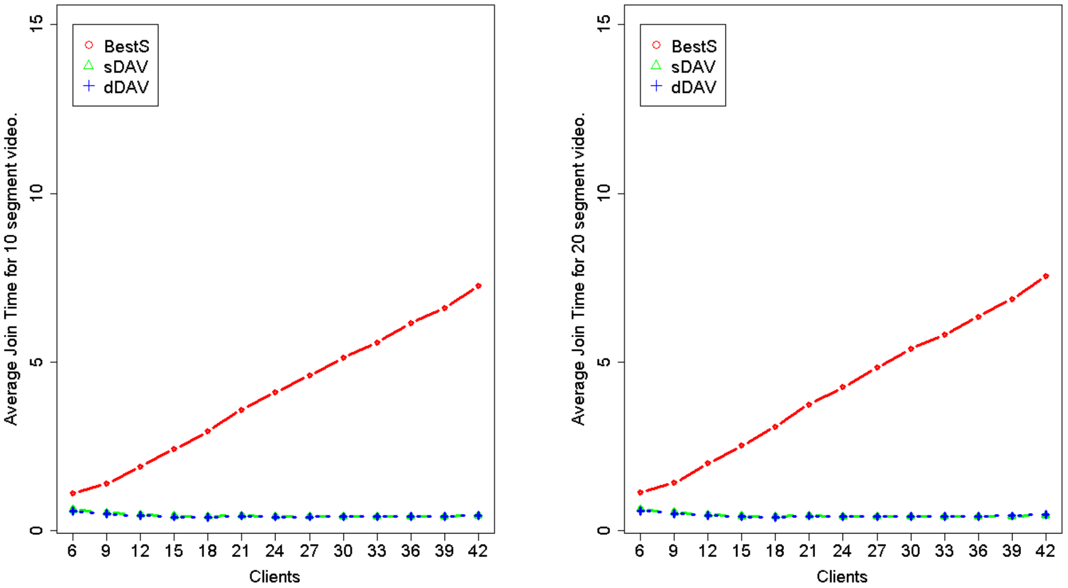

Figure 5-47: TC5 Segment Duration Average Join Time (seconds)

Join times are depicted in Figure 5-47. A small reduction in the average join time is present due to the difference in the amount of data required to reach the buffer playout level. This results from the use of longer segments. The first segment is requested at the lowest bitrate, and since the segment length is 4 seconds longer in TC5 (compared to other cases), a longer section of requested video will be downloaded at the lowest bitrate. TC1 results indicate 5.51s (BestS), 0.70s (for both sDAV and dDAV) as join times for 24 clients requesting ~200s video. Here the join times are: 4.25s (BestS) and 0.42s (DAV settings) for a video of 20 segments. There is no significant difference between sDAV and dDAV scenarios.

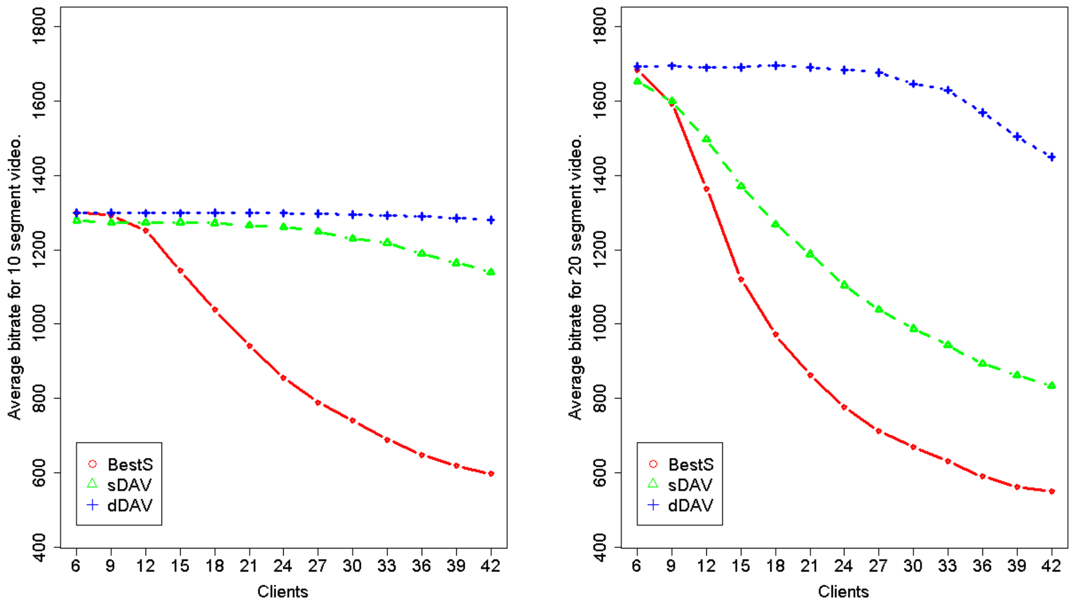

Figure 5-48: TC5 Segment Duration Average Bitrate (kbps)

The improvement in the average bitrate per client is not as significant for shorter videos in this setting, as the segments are 4 seconds longer, the video of the same duration contains fewer segments (e.g. 10 instead of 16). The DASH player initially requests the first segment at the lowest bitrate quality, and subsequently, the bitrate of the requested segments is gradually increased (in steps of one bitrate level if the link bandwidth permits) to avoid unnecessary fluctuations and smoothly change from one bitrate to the next. Still, DAV deployment produces significantly better results as indicated in Figure 5-48.

No significant rebuffering ratio or rate is observed in this setting.

The average bitrate is maintained with DAV deployment, as the content is found locally on DAV Clients as indicated in Figure 5-49. The depicted simulation results indicate that in this setting dDAV continues to deliver the most significant improvements.

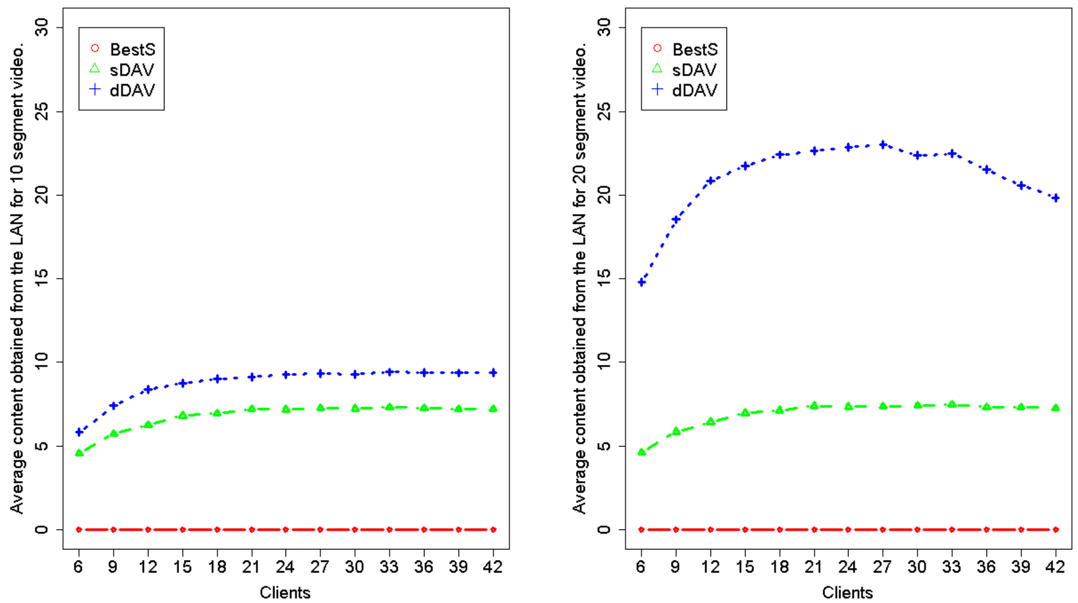

Figure 5-49: TC5 Segment Duration Average Content (Mb) from DAV Clients

Finally, in this setting, in line with 6 second long segments, the video traffic on the link Server-DAV Gateway is significantly reduced as indicated in Figure 5-50.

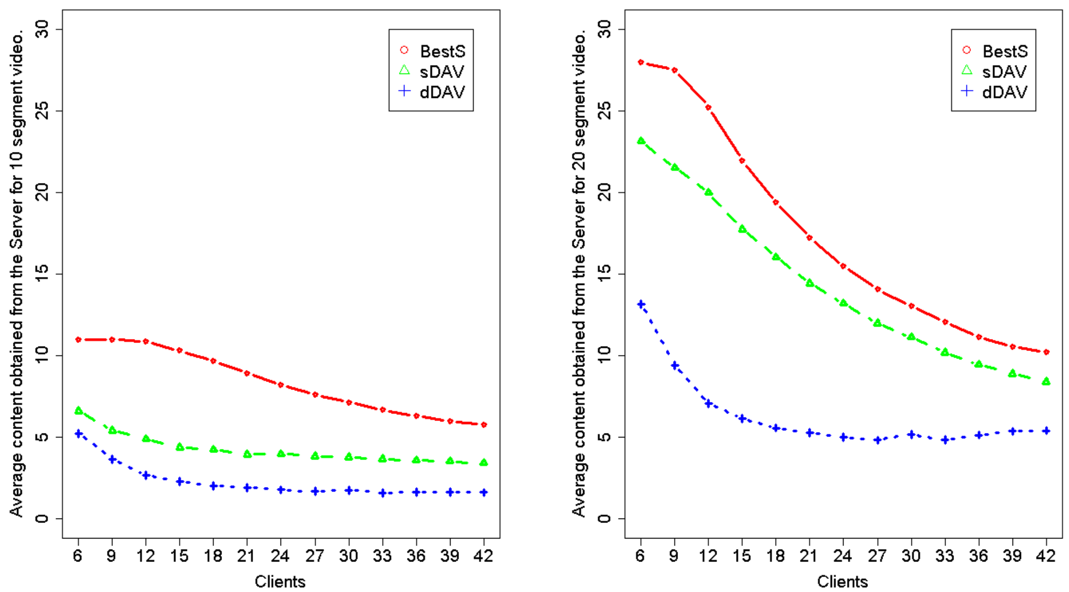

Figure 5-50: TC5 Segment Duration Average Content (Mb) from Remote Server

5.3.5 DAV Evaluation Summary

This section presents evaluation results for DASH-based Performance Oriented Adaptive Video Distribution Solution (DAV) [19], one of the solutions developed in this research. The evaluation setting is described in terms of video, client and network setting. Three scenarios are evaluated: BestS (Scenario 1) – content downloaded from remote server, sDAV (Scenario 2) – local content utilisation using static MPDs (to a degree similar to peer-assisted DASH system (pDASH) [117]) and dDAV (Scenario 3) – local content utilisation using dynamic MPDs. Both static and dynamic MPDs are produced by DAV Gateway. A number of test cases are presented in order to investigate the impact of the size and number of video segments, the timing and the number of requests, the server-DAV Gateway link characteristics (throughput/delay). The evaluation results are presented and discussed. It can be observed, regardless of test case investigated, that the overall performance of the system is enhanced in the sDAV and dDAV scenarios - when local content is made available to clients. In all cases the average join time is decreased significantly while the rate of buffering events and buffering ratio are reduced. In most cases the average bitrates are significantly increased. It should be noted that with DAV deployment the utilisation of the Server-DAV Gateway link is reduced, as a large portion of the video content is found locally and is not requested/delivered from the remote server. In all settings Scenario 3 (dynamic DAV MPDs) outperforms Scenario 2 (static DAV MPDs).

5.4 Summary

This chapter presents evaluation results for the solutions developed in this research. The evaluation setting is described, test cases introduced and results presented and discussed.

For conciseness, results for a number of test cases are provided in graphical form only.

6 Conclusions

This thesis presents novel solutions addressing several issues relating to video delivery by adaptive Personalised Learning Systems. A discussion of insights arising from the literature review presented in Chapters 2 and 3 and comparisons to related work are summarised in Section 6.1. The chapter continues with a summary of contributions and an overview of simulation results in Section 6.2. Deployment overheads and solution limitations are discussed in Sections 6.3 and 6.4, respectively. Existing systems that could be enhanced with the proposed solutions are identified and presented in Section 6.5. Suggestions for future work are provided in Section 6.6. Section 6.7 contains concluding remarks.

6.1 Literature Review Insights

This dissertation presented, in Chapter 2, the technological setting for this thesis: a literature review of online video delivery with an emphasis on video streaming over HTTP and MPEG-DASH. Section 6.1.1 presents a comparison of the proposed DPEA solution with other solutions in this area. Additionally, Chapter 3 presented a literature review in the area of Web-based learning systems with a focus on adaptive Personalised Learning (PL) systems. Such systems, including Adaptive Hypermedia systems were investigated to identify issues relating to learning content adaptation to the delivery context. Insights and a brief comparison of DPEA with the related solutions in the area are presented in Section 6.1.2.

6.1.1 MPEG-DASH Setting

Video streaming approaches have shifted from UDP-based to TCP-based in recent years. Most existing HTTP/TCP-based solutions are proprietary (e.g. Adobe HDS [99], Apple HLS [100], Microsoft Smooth Streaming [101]). MPEG-DASH [12] is an international standard for describing multi-rate encoded multimedia for adaptive HTTP streaming. Client players dynamically choose the quality (bitrate) for segments of a DASH media presentation to request the best match to estimated current network dynamics and/or to available device resources. DASH-based content is growing increasingly prevalent, where the quantity of free and commercially available videos is expanding rapidly. The DPEA architecture proposed in this research enhances DASH video distribution in a campus setting by utilising best performing local and remote hosts.

dPOAA component of DPEA evaluates remote servers based on their historic performance in terms of the measured throughput and RTT of the link to the server. This rating is used for remote server selection when the requested video resides on multiple servers.

Video content is typically delivered by CDNs which host videos at a number of servers. Distributed DASH datasets such as [124] provide identical DASH content on multiple sites. The standard supports provision of alternate base URLs through the BaseURL element at any level when identical segments are accessible at multiple locations. Section 5.6.5 (Alternative base URLs) of the standard states: “In the absence of other criteria, the DASH Client may use the first BaseURL element as ‘base URI’. The DASH Client may use base URLs provided in the BaseURL element as ‘base URI’ and may implement any suitable algorithm to determine which URLs it uses for requests.” [13, p. 66]. Accordingly, the first challenge after “retrieving an MPD with multiple BaseURLs is determining with which BaseURL to start a DASH session. As the BaseURL does not have any metrics associated (some text omitted) it is up to client implementation to decide the location of the first segments to be downloaded.” [124, p. 134]. The same source stipulates that determining the best BaseURL may influence the initial delay. While the standard supports specification of multiple hosting servers, it does not propose a selection algorithm. To the best of our knowledge, there are no other DASH-based solutions that provide intelligent remote host selection based on statistical estimators.

Server selection strategies are typically deployed by content providers (e.g. within a CDN) to reduce cost and to improve the end-user experience through load balancing. While they utilise proprietary algorithms, studies reveal the algorithms applied by content providers are geographically (locality) aware (e.g. YouTube [269]) and mainly static in nature (e.g. Netflix [107]). Proposed solutions, such as the Control Plane framework [136] allocate CDNs based on global knowledge of delivery network (CDN performance, client activity, etc.).

It can be argued that clients are ideally positioned to observe local network performance and consequently to react promptly to network dynamics, so dPOAA chooses servers based on their historical performance observed from a client’s perspective without any input from the hosting server. A client-based approach to dynamic CDN selection was explored in [149], where multiple dynamic probes were used to identify the best performing CDN at session startup. In contrast, dPOAA selects servers based on historic readings without incurring additional probing traffic.

DAV components of DPEA utilise locally available content through modification of the MPD file provided to the video requesters. Here, DAV is compared to solutions that propose use of content residing on peers and to systems that centrally utilise client provided information. The peer-assisted DASH system (pDASH) [117] was described in Section 2.4.12. pDASH, similar to DAV, modifies MPD files. In the pDASH setting modified MPDs provide clients with an option to download parts of segments (chunks) from Web nodes (peers) which have the segments cached. However, unlike pDASH, DAV considers peer hosts inside a campus network where uplink characteristics need not be considered and consequently segments need not be “chunked”. Additionally, the utilisation of local content requires minimal firewall modifications (a local system administrator simply opens port 80 on client machines). Furthermore, while pDASH randomly selects peer hosts, DAV selects the best performing hosts for inclusion in the modified MPD, based on host rating. Apart from simplifying the decision-making process at the client, limiting the number of alternative hosts listed per segment also reduces the size of the

MPD file. Furthermore, the pDASH player requires an algorithm for concurrent download of peer-chunks and segments from servers, while DAV's modified MPDs can be used with standard DASH players. pDASH focuses on reducing bandwidth utilisation, and client side evaluation results were not presented in [117]. Apart from modifying the original MPD at request time, the DAV Gateway provides dynamic MPD generation at each segment request. The latter approach outperforms the MPD modifications proposed in pDASH.

QDASH [122] utilises a hardware proxy hosting QDASH-abw [122] which accurately measures available link capacity to achieve gradual quality changes. While a QDASH-enabled video player maintains a "light-weight flow" [122] with the proxy to receive current measurements, the proxy does not provide further guidance in terms of hosting server selection. Furthermore, QDASH does not take the locality of the segments into account.

Similar to our solution, clients in NOVA [126] contact the network controller (centralised unit) to indicate segment download completion. However, NOVA clients do not provide information about locally stored content, so such content cannot be used by other clients.

Control Plane framework [136] also receives client side information, where active clients periodically (every few seconds) report quality statistics (e.g., buffering, join time, average bitrate) to the Framework's Measurement Engine. However, the downloaded content is not utilised by other active clients.

6.1.2 Personalised Learning Systems

Online distributed systems, despite continuous hardware and network capacity improvements, remain vulnerable to delays, especially in settings where a high number of Web users access real time media. Open and distributed PL systems suffer from the same problem. Chapter 3 of this thesis presented a review of adaptive PL systems. One of the first families of well-defined and formally evaluated personalised online systems in the educational setting was Adaptive Educational Hypermedia (AEH) systems. These systems were investigated in Chapter 3 with a focus on their structure and adaptation approaches. AEH systems adapt learning material (in terms of content selection and presentation) to learner characteristics and learning context. Therefore, approaches to context-aware adaptation were outlined and PL systems supporting network and user device adaptation were explored.

Early AEH systems were not modularised and offered limited opportunity for improvement since, in most cases, modification and/or extension required full access to the system source. Network-awareness could be implemented by changing the system's Presentation Model (PM) and Adaptive Engine (AE), so that the system considers network factors and user device. Third generation, service oriented systems, such as APeLS [197] addressed this issue, providing extensibility via new modules, such as the performance-aware solutions developed in this work.

While AEH systems are online systems, potentially using distributed content, very few consider network/device characteristics when adaptation is performed and would benefit from the solutions developed in this research. Solutions that perform adaptation based on user device and the underlying network conditions are limited to static content (e.g. QoE-aware AHA system (QoEAHA) [63]) or focus on device characteristics only (e.g. Mobile Mathematics Tutoring (MoMT) [240]). However, neither of these solutions considers transmission of video content. Solutions that deal with video [228] make adaptation decisions at the provider's side. Our solutions focus on the learner side, which is an approach that scales better.

6.2 Contributions and Evaluation Results

We are witnessing an explosion in free educational video availability (Coursera [2], edX [2], MITx [270], Udacity [29], etc.) and a parallel increase in demand for education (e.g. a 50% increase in programming courses in Australia and New Zealand from 2010 to 2013 [271]). Educational video content can be produced rapidly and at a low cost. Today's students demand access to course material via their mobile devices [4] and have a strong preference for video content including lecture recordings as indicated in Figure 6-1. Students expect high quality video streaming regardless of their device and network delivery characteristics. Thus there is a clear need to adjust video content selection to both network conditions and device characteristics in order to improve viewing experience as educational video becomes an integrated part of the learning process. Progress has been made with the deployment of adaptive bitrate streaming (e.g. MPEG-DASH) that reduces the number of playout interruptions due to buffer underruns, which is an important factor in determining the overall viewing experience.

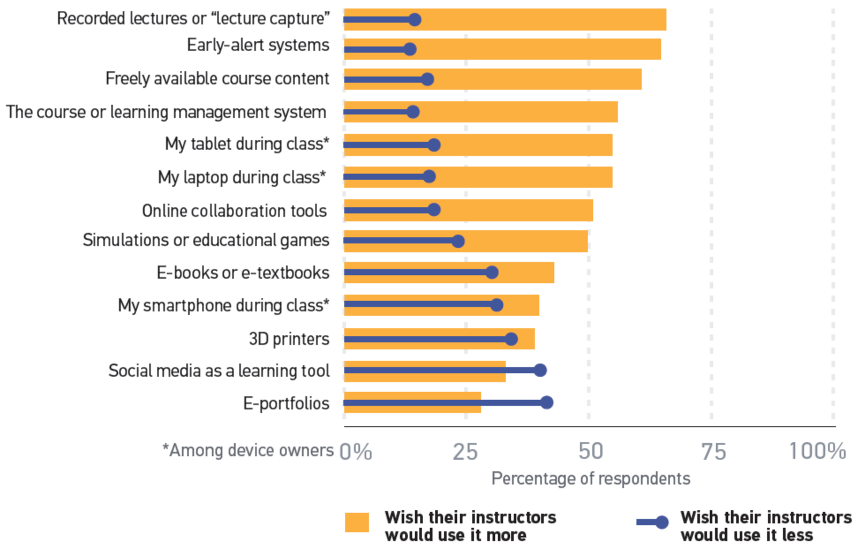

Figure 6-1: Student Responses Regarding the Use of Technology [28]

For further improvement we propose DPEA which deploys two novel solutions, dPOAA and DAV. dPOAA performs server selection while DAV utilises locally available content to provide high quality video streaming to multiple learners requesting identical video content residing on multiple remote servers. These solutions and associated evaluation results are reviewed in the following sections.

6.2.1 POAA

Delays (manifested as pauses prior and during video playout) are identified as a particular annoyance for online content consumers, a phenomenon discussed in Section 2.3.2. Performance Oriented Adaptation Agent (POAA) solutions were proposed and developed in order to minimise initial delays in learning content download by determining the best performing server when multiple remote servers host requested content. POAA solutions are located at the campus gateway and the host selection process is based on the observed quality of network connection links between the servers (e.g. learning content repositories) and the campus network. These solutions address research question 1.2.1 of the research problem presented in Section 1.2: "How can better video quality be obtained when video content resides on multiple remote servers?".

Open POAA (oPOAA) [14]–[16] is a solution that selects the hosting server from which to download learning objects residing on a number of remote servers in order to minimise initial delays. The solution extends the Learning Object (LO) selection process in Open Corpus adaptive PL systems by considering the links to servers hosting LOs. A literature review of Open Corpus adaptive PL systems including Adaptive Hypermedia systems was compiled to identify issues and related works in the area. The associated research contribution is the design and evaluation of an oPOAA algorithm based on a utility function. The proposed algorithm deals with all types of educational content delivered over UDP. oPOAA calculates the estimated delivery time for each server hosting the relevant LO. The LO is then requested from the server with the shortest estimated delivery time. This algorithm was evaluated in a simulated setting (NS-2 [272]) and results demonstrating improvements in download speed were presented in Section 5.1.

DASH-based POAA (dPOAA) [17], [18], focuses on MPEG-DASH [12], [13], [98] video content only. dPOAA is an efficient solution as the learning content is available in different qualities (bitrates) on multiple servers removing the need for transcoding. dPOAA rates remote servers based on the observed throughput and RTT of the connection link where the rating is calculated using a utility function. Video content is then requested from the remote server with the highest estimated performance. The dPOAA algorithm was evaluated in a simulated setting (NS-3 [25]). To our knowledge, there is currently no rival DASH-based statistical estimator server selection solution so during evaluation dPOAA was compared with random server selection, always the same (best) server selection and a TCP variant of the oPOAA algorithm. The results presented in Section 5.2 demonstrate that the deployment of dPOAA enhances user experience as it reduces both rebuffering rate and ratio as well as join times, while maintaining acceptable MOS levels. dPOAA requires no modification of the HTTP servers hosting video content and could be easily applied as a plug-in for MPEG-DASH players or as a server rating solution for DAV (as was done in this thesis). Client-side solutions are criticised for being unaware of significant temporal and spatial variability in provider network performance [136].

dPOAA, when deployed at DAV Gateway (campus proxy), utilises information about past performance of the hosting servers based on the interactions of all campus users, i.e. more complete and up-to-date information about hosting servers.

6.2.2 DAV

The proliferation of educational video content and increasing student numbers (using a variety of devices for learning) has heightened demand for high quality streaming services. The DASH-based Performance Oriented Adaptive Video Distribution Solution (DAV) [18], [19] is deployed in a setting where members of a large class (all students with the same or similar learning profile) concurrently watch an educational video. In this context, the MPEG-DASH standard is harnessed in an innovative way to utilise locally available content leading to a better quality viewing experience. DAV considers viewer preferences (i.e. learner profiles as provided by the PL system), viewing device capabilities and utilises content available locally by recruiting groups of active (i.e. watching) learners within the campus network to share their downloaded video segments with other users in the campus network. The DAV solution addresses research question 1.2.2: "How can video streaming be improved using video content available within the campus network? How can new standards for Internet video delivery be best utilised in this context?".

The solution consists of a DAV Gateway (deployed at the campus gateway) and DAV Client (deployed at selected nodes in the campus network). The solution was evaluated in a simulated setting (NS-3 [25]). Additional contributions of the research described in this thesis include the development of a number of NS-3 application modules described in Section 5.2.1. These modules are deployed to request and deliver video segments as well as to model and track video playback.

The results presented in Section 5.3 demonstrate that DAV deployment enhances the performance of a personalised distributed video delivery system, which in turn improves viewing experience. The playout is improved with notably reduced join times and increased bitrates while rebuffering rate and ratio are at minimal levels. The solution requires no modification of the HTTP servers hosting video content. Furthermore, both DAV Client-enabled devices and also with no installed DAV Client benefit from the proposed approach.

6.3 DPEA Deployment Overheads

This section presents a discussion of DPEA (dPOAA and DAV Gateway and Client) overheads.

6.3.1 dPOAA

The overheads introduced by dPOAA deployment are limited to computational requirements (storage and processing requirements) as no additional network traffic is introduced by dPOAA. The performance data for the link connecting the remote server with the campus network is collected during client-server interaction. The rating algorithm is of low computational complexity. The servers hosting content are identified by the BaseURL of which an MPD typically contains a limited number (e.g. 2 or 3). The server rating is calculated based on the X (e.g. 5) most recent performance readings for each server identified in the MPD. Thus, producing ratings for an MPD requires a constant amount of time. Overall performance is proportional to the number of video requests. dPOAA maintains limited historic information (e.g. X last readings) on the performance of remote servers and storage requirements are proportional to the number of remote servers.

6.3.2 DAV

The overheads introduced by DAV deployment can be grouped in two categories: network traffic overhead (updated MPDs and DAV Client updates), computational performance (DAV Gateway and Client processing and storage requirements).

Network Traffic Overhead. The network traffic is increased by the delivery of additional updated MPD files. The overhead depends on the type of MPD used. In the case of static DAV, similarly to typical MPEG-DASH video delivery, the MPD file is delivered once at the time of the video request. Implementation of DAV results in an increase in the size of the MPD file. In the DAV Gateway-modified file, BaseURL elements are specified at representation level for each segment of the requested video. This in turn increases the number of entries in the MPD file. A sample MPD file is provided in Figure 4-17 (page 87) where an entry is required for each host storing the segment and additional entries are required for each segment in each representation of the requested video. The volume of MPD data is further increased with dynamic MPDs (see Figure 4-18 on page 88). In this case, an updated MPD is requested/delivered for each period (containing a single segment) of the requested video. In this case, the quantity of MPD data transmitted is not significantly increased, compared to the static MPD approach, however, updated MPDs are sent period number times which incurs overheads in terms of TCP connection establishment and data transfer. Transmission of modified MPD files is confined to the campus network but should delivery improvements be required, MPD files can be compressed. Overall, improvements in terms of delivery and playout significantly outweigh the costs incurred by increased modified MPD sizes used in the DAV setting. DAV Clients submit updates on locally available segments. Limited additional traffic is introduced with these updates, however, their low frequency and the limited message sizes means campus network performance is not significantly affected.

DAV Gateway, Client Performance and Computational Complexity of introduced algorithms. The *DAV Gateway* is aware of the content available locally and while the deployed MPD building algorithm is simple, the algorithm's complexity is proportional to the number of local hosts storing the requested content. DAV groups users by (a) enrolled course and learning preferences (based on the information provided by the associated PL system) and (b) viewing device type. This reduces the number of considered nodes during MPD building. Furthermore, only nodes storing content belonging to the same video are considered. In terms of storage, the DAV Gateway stores information about the content, but not the content itself, so storage requirements are not an issue. The *DAV Client* performs a number of tasks, it accepts requests and provides video segments to other nodes in the network, it informs the associated DAV Gateway about the locally stored content and it acts as a DASH player. While providing content for other nodes has an impact on the Client, the rate of response to requests for local content is proportional to the device capabilities and the MPD building process ensures that the number of concurrent requests for content is limited so that it does not adversely affect Client performance in terms of playout quality. DAV Clients send updates regarding locally stored content, however, the algorithm and resources required for such updates do not significantly impinge on overall device performance.

6.4 DPEA Limitations

The proposed solutions are domain and criteria-specific. In terms of domain, they are deployed in conjunction with a personalised system (e.g. PL system) which provides a user modelling facility for grouping users with similar video requirements. Furthermore, server selection decisions are based on the values of a limited set of estimated QoS attributes (e.g. RTT and throughput). While it is an advantage that the estimations are derived without a direct input/involvement from the user, other criteria could be considered. Furthermore, the proposed solutions do not consider trends and seasonal patterns (e.g. time-of-day, day-of-week, week-of-semester, etc.) in the collected link performance data.

6.4.1 dPOAA

The server selection decisions are made at run-time, so the efficiency of the applied selection mechanism is crucial. Therefore, dPOAA applies a utility function to historical server link performance to select the remote server with the maximum utility. The dPOAA approach is highly efficient in terms of computation time as the time complexity is O(N), where N is the number hosting servers. Complex selection problems are typically NP-hard problems where an optimal solution may not be found in time to meet real-time requirements. For example, typical solutions for service selection are of exponential time complexity (e.g. [142]) but could be deployed if the number of candidates is limited. Since the number of remote servers hosting identical video content is limited, the prediction algorithm could be made more sophisticated.

6.4.2 DAV

One of the key requirements for this solution is timely (run-time) reaction to changes in the delivery network environment. Therefore the generation of new MPDs (involving selection of the hosting servers) should not negatively affect video playout and consequently DAV applies a simple multiplicative utility function considering content freshness and device load for a limited number of nodes. However, additional parameters could be considered in this process, such as remaining battery life for portable devices.

6.5 Systems/Settings that Benefit from DPEA Solutions

With the evolution and standardisation of eLearning systems many researchers in the area see modular, distributed, open corpus, semantically well-described, pedagogically-sound systems as the future. These systems, apart from performing typical adaptation tasks, must deal with open corpus content domain to identify and integrate relevant and suitable learning content.

This section identifies distributed/open systems that would benefit from the solutions developed in this research. POAA solutions select best connected servers and could be deployed with the indicated systems to enhance their adaptation/selection process. For all listed systems, the DAV solution could be deployed when DASH video is used in a campus setting. The following sections present application possibilities for the proposed solutions.

6.5.1 Open Corpus Context and Digital Content Repositories

Open Corpus Content is content that is freely available for use by any educational institution or system. Such content is available in public repositories. These, so called, Digital Educational Repositories (DER)s or digital Learning Object Repositories (dLOR) foster courseware reusability through hosting reusable learning content. They host pools of varied learning objects, ranging from simple, mostly static, learning objects, to highly interactive and adaptive learning content, including teaching texts and graphics, interactive educational software, animations, simulations, video/audio recordings, podcasts, 3D artefacts, various types of assessment, etc. The access, retrieval and storage of LOs is simplified, where, for example, LOs are automatically added to DERs when published to the local virtual learning environment.

Integration of existing DERs is of strategic interest to the European Union which has funded a number of DER development projects (e.g. Alliance of Remote Instructional Authoring and Distribution Networks for Europe - ARIADNE[18]) and cross-integration of national DERs (e.g. the National Digital Learning Resources (originally called National Digital Learning Repository) – NDLR [273] in Ireland) over the past decade. Today, a number of open source initiatives, such as Open Science Resources - OSR[19]and ARIADNE projects remain ongoing. Worldwide, a number of DERs exist such as, Multimedia Educational Resources for Learning and Online Teaching and Gateway to Educational Materials - MERLOT[20] [274], and Education Network Australia Online - EdNA [275].

DERs are large collections of LOs, storing similar or identical learning content and thus oversupply of information may occur, disorienting the learner. In this context open AEH systems are of significant benefit as they provide support for the selection of the best LOs for a particular learner, based on the learner's interests, goals, background knowledge, learning style, etc. A number of factors determine the technical performance of such distributed systems. For

[18] http://www.ariadne-eu.org [Accessed: 2-Jan-2016]
[19] http://www.osrportal.eu/en/repository [Accessed: 2-Jan-2016]
[20] https://www.merlot.org [Accessed: 2-Jan-2016]

example, every personalisation task requires considerable information exchange between various components of a distributed adaptive system (i.e., a portal, a personalisation service, a user model server, etc.). Furthermore, the learning content is stored on a remote DER and the delivery time will depend on the quality of the link between the DER and the portal. Therefore, the performance of these systems could be improved with the deployment of the POAA solutions.

6.5.2 Learning Portals

Knowledge Tree Portal [198] is a distributed learning management system providing centralised access (single sign-on functionality) to different kinds of learning content. Instructors can use this portal to structure the learning content as a sequence of nested folders to match the needs of their courses. The portal implements several adaptive navigation techniques to help students choose the most suitable learning activity, stored at geographically distributed *Activity Servers*. Such servers host both static and interactive/adaptive content. *Value adding service* is course-neutral and extends "raw" content/services with added functionality, such as sequencing, annotation, visualisation and integration. The CUMULATE server [185], [186] is used as the *Student Model Server*. The system developers stress the role of system performance stating that "frequent inter-server communication should not be allowed to slow down the student interface" [198, p. 6]. Performance of these systems could be improved with POAA deployment.

6.5.3 Educational Institutions with Limited Internet Connectivity

Education is key to escaping poverty in third world and post-crisis regions. There is a wealth of valuable educational video resources available online (e.g. [2], [3]). Such resources, if accessible, provide the third world with a vital opportunity to develop. However two problems hamper accessibility: (a) the cripplingly high cost of Internet connectivity and (b) the poor quality (low bandwidth) network connection from the third world to the rest of the world. For example, the University of Kinshasa is the largest university in the Democratic Republic of the Congo (nearly 30,000 students, faculty, and research staff) however "its link to the outside world is no better than that of a typical household in the United States or Europe" [276, p. 55]. On-demand access to video content via traditional multimedia players in such a context is therefore impractical. However, the local campus data network is fast enough to support on-campus e-mail, virtual library access, and online coursework. This setting is precisely the context for which DAV provides best results, as the content present in the internal network can be exploited thus alleviating the requirement for an expensive high bandwidth connection to the outside world. Improvements in the network experience (e.g. faster browsing) is the most requested change among Internet users across Africa [1].

6.6 Future Work

This section indicates avenues that can be explored in future extensions to the proposed solutions.

Generalised Adaptation Framework. The solutions proposed in this research deploy utility-based adaptive algorithms and could be further generalised into an adaptation framework that could be deployed in conjunction with a generic personalised system. For example, a solution that enriches standard application logic (base-level system such as a DASH player) with a control loop (functionality added with a DAV Client unit) that monitors the context of execution (video segment download), that determines the changes to be enforced, and enacts them is an adaptive system that could be mapped to an architectural pattern such as MAPE-K [153].

Prediction Element. The proposed solutions can be extended to consider seasonal and diurnal patterns in the collected data. This could be achieved by collecting repeated measurements under each condition and deriving models which both smooth inputs and predict the trend and periodicity in historical data (e.g. [277]) to further improve the network performance estimates at some future point in time.

Consideration of other attributes. Furthermore, the set of adaptation criteria could be extended with additional criteria encompassing attributes that are directly provided by content providers (e.g. price), or based on user feedback (e.g. server reputation, usability, threshold levels, etc.). For example NOVA [126] allows users to set rebuffering thresholds. For example, the proposed remote server selection solutions (oPOAA and dPOAA) could be extended to use the knowledge accumulated about remote servers by a wider community of users. However, the well-known problems related to "following the crowd" need to be addressed (e.g. when a user recommends an inappropriate resource of poor quality other users may tend to follow this bad example).

Use of Mobile Devices and Other Types of Networks. Mobile video streaming solutions utilise mobile devices and cellular networks for video streaming. Our solution deploys centralised tracking (DAV Gateway) and high-performance nodes hosting DAV Clients to boost the viewing experience for all users in the campus network. This is achieved by utilising locally availably content and thus augmenting video distribution capacity in a campus setting. Our solutions do not utilise cellular networks or low-performance nodes (e.g. smartphones) as content providers (DAV Clients) but could be extended in this regard.

An example of a DASH-based system for video sharing deployed in a mobile P2P network is MyMedia 1.0 [278]. MyMedia 1.0 is an Android mobile application which improves quality of DASH-annotated (video on demand and live sessions) content in wireless networks with an unstructured and semantic P2P overlay. This system deploys a high-precision semantic P2P search to perform DASH streaming from mobile to mobile devices in unstructured wireless P2P networks.

Content (Segment) Prefetching and Pushing. DAV does not prefetch nor push content to campus network nodes, but instead uses content already present at the node. The DAV Client is installed on well-resourced nodes, and the bitrate of downloaded segments matches screen requirements of well-resourced peers. In this setting, handheld devices, can access segments stored on DAV Clients, but the bitrate of such segments is currently too high for the handheld devices. In this context, DAV could be extended to request and push segments of lower bitrate to active DAV Clients and to provide bitrates appropriate for handheld devices.

Request Prioritisation. The DAV Gateway processes requests for video on a first-come first-served basis regardless of the type of node requesting the content. DAV Gateway functionality could be modified to prioritise requests from DAV Clients. In this case, when multiple requests for the same video are detected, DAV Clients would be sent the modified MPDs first and will begin downloading content earlier, which in turn increases the bitrate of downloaded segments. Thus, higher quality segments would be available to other nodes in the LAN.

6.7 Concluding Remarks

University campus students are demanding more educational video [28]. Large quantities of educational video are offered free of charge. HTTP servers provide multiple versions of a video (i.e. segments of various bitrates). MPEG-DASH provides a practical solution for addressing the surge in availability of Internet connections and the ubiquitous utilisation of smartphones [279]. University campus networks provide free (to students), fast and reliable communication networks and local well-resourced devices that can host media segments. The proposed DPEA solution considers network and viewing device characteristics to exploit both remote and local content in order to achieve high video quality levels that will enhance the learning process.

The proposed solutions could be applied to any situation where a group of users on the same network will watch the same collection of videos (not necessarily educational). Set in an educational context the solutions bring most benefit for the following reasons:

- A typical university campus network is constantly utilised by students sharing similar interests/requirements and having similar/identical needs for educational video. Personalisation is achieved using User Models provided by the associated PL system.
- There are a large number of worldwide settings where university campus network infrastructure is adequate, but where Internet connectivity is poor. Our solutions increase the quality of delivered video, even where Internet connection is constrained.
- The current trend is towards the use of educational video which will place increasing demand on the campus network and Internet connection.

However, our solutions are not exclusively tied to education. They can be deployed to settings where large groups of users are interacting with a personalised system (e.g. personalised video retrieval system) in a corporate network (e.g. training or promotional video).

7 Bibliography

[1] J. Meffert, E. Chappuis, E. Hazan, E. Duncan, and N. Wagner, "iConsumers: Life online," McKinsey & Company, 2013.

[2] "edX." [Online]. Available: https://www.edx.org/. [Accessed: 13-Oct-2013].

[3] "Coursera," *Coursera*. [Online]. Available: https://www.coursera.org/. [Accessed: 13-Oct-2013].

[4] E. Dahlstrom, "ECAR Study of Undergraduate Students and Information Technology," *EDUCAUSE Center for Applied Research*, 2012. [Online]. Available: http://net.educause.edu/ir/library/pdf/ERS1208/ERS1208.pdf. [Accessed: 07-Aug-2013].

[5] S. Smith, G. Salaway, and J. Caruso, "The ECAR Study of Undergraduate Students and Information Technology," *EDUCAUSE Center for Analysis and Research*, 2009. [Online]. Available: http://net.educause.edu/ir/library/pdf/ers0906/rs/ers0906w.pdf. [Accessed: 07-Aug-2013].

[6] B. Oliver and B. Whelan, "Designing an e-portfolio for assurance of learning focusing on adoptability and learning analytics," *Australasian Journal of Educational Technology - AJET*, vol. 27, no. 6, pp. 1026–1041, 2011.

[7] M. Sharples, P. McAndrew, M. Weller, R. Ferguson, FitzGerald, E., Hirst, T., Mor, and Y., Gaved, M. and Whitelock, D., "Innovating Pedagogy 2012," Open University Innovation Report 1 Milton Keynes: The Open University ., 1, 2012.

[8] B. R. von Konsky and B. Oliver, "The iPortfolio: Measuring uptake and effective use of an institutional electronic portfolio in higher education," *Australasian Journal of Educational Technology*, vol. 28, no. 1, pp. 67–90, 2012.

[9] D. Mackenzie, "Who has the fastest internet?," *IEEE Spectrum*, vol. 47, no. 8, pp. 60–60, 2010.

[10] C. H. Muntean, "Quality of experience aware adaptive hypermedia system," doctoral, Dublin City University. School of Electronic Engineering, 2005.

[11] S. Moebs, "QoS-based experience-aware adaptation in multimedia e-learning - A learner, is a learner, is a user, is a customer," doctoral, Dublin City University. School of Electronic Engineering, 2011.

[12] "ISO/IEC 23009-1:2012 - Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats." [Online]. Available: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=57623. [Accessed: 23-Aug-2012].

[13] ISO, "ISO/IEC 23009-1:2014 Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats," May 2014.

[14] L. Rovcanin, C. H. Muntean, and G.-M. Muntean, "A Performance Oriented Adaptation Agent for Open Adaptive Hypermedia Systems," presented at the IT&T, Cork, Ireland, 2005.

[15] L. Rovcanin, C. Muntean, and G.-M. Muntean, "Performance Enhancement for Open Corpus Adaptive Hypermedia Systems," in *Adaptive Hypermedia and Adaptive Web-Based Systems*, vol. 4018, V. Wade, H. Ashman, and B. Smyth, Eds. Springer Berlin / Heidelberg, 2006, pp. 462–466.

[16] L. Rovcanin, C. H. Muntean, and G.-M. Muntean, "Performance Aware Adaptation in Open Corpus E-Learning Systems," presented at the International Workshop on Technologies for Mobile and Wireless-based Adaptive e-Learning Environments, 2008, p. 27.

[17] L. Rovcanin and G.-M. Muntean, "A DASH-aware Performance Oriented Adaptation Agent," presented at the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, Beijing, China, 2014, pp. 1–6.

[18] L. Rovcanin and G.-M. Muntean, "MPEG-DASH: A Solution for Improving Video Delivery Quality in Heterogeneous Network Environments," in *Convergence of Broadband, Broadcast, and Cellular Network Technology*, G.-M. Muntean and R. Trestian, Eds. IGI Global, 2014.

[19] L. Rovcanin and G.-M. Muntean, "A DASH-based Performance-oriented Adaptive Video Distribution Solution," presented at the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, London, UK, 2013, pp. 1–6.

[20] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1," 1999. [Online]. Available: http://www.w3.org/Protocols/rfc2616/rfc2616.txt. [Accessed: 08-Dec-2012].

[21] J. Postel, "Transmission Control Protocol." [Online]. Available: http://gamay.tools.ietf.org/html/rfc793. [Accessed: 24-May-2012].

[22] S. Basso, A. Servetti, E. Masala, and J. C. De Martin, "Measuring DASH Streaming Performance from the End Users Perspective Using Neubot," in *Proceedings of the 5th ACM Multimedia Systems Conference*, New York, NY, USA, 2014, pp. 1–6.

[23] D. E. Comer, D. Gries, M. C. Mulder, A. Tucker, A. J. Turner, and P. R. Young, "Computing As a Discipline," *Commun. ACM*, vol. 32, no. 1, pp. 9–23, Jan. 1989.

[24] "The Network Simulator - ns-2." [Online]. Available: http://www.isi.edu/nsnam/ns/. [Accessed: 29-Nov-2012].
[25] "Network Simulator ns-3." [Online]. Available: http://www.nsnam.org/. [Accessed: 15-Dec-2014].
[26] J. Scheirer, R. Fernandez, J. Klein, and R. W. Picard, "Frustrating the user on purpose: a step toward building an affective computer," *Interacting with Computers*, vol. 14, no. 2, pp. 93–118, Feb. 2002.
[27] A. Trifonova, "Towards Hoarding Content in M-Learning Context," Ph.D., University of Trento, Trento, 2006.
[28] E. Dahlstrom and J. Bichsel, "ECAR Study of Undergraduate Students and Information Technology," CO:ECAR, Louisville, Oct. 2014.
[29] "Advance Your Education With Free College Courses Online - Udacity." [Online]. Available: https://www.udacity.com/. [Accessed: 13-Oct-2013].
[30] R. M. Steinman, Z. Pizlo, and F. J. Pizlo, "Phi is not beta, and why Wertheimer's discovery launched the Gestalt revolution," *Vision Research*, vol. 40, no. 17, pp. 2257–2264, Aug. 2000.
[31] H. R. Wu and K. R. Rao, *Digital Video Image Quality and Perceptual Coding*. CRC Press, 2005.
[32] "The JPEG committee home page." [Online]. Available: http://www.jpeg.org/. [Accessed: 14-Oct-2013].
[33] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Transactions on Consumer Electronics*, vol. 16, no. 4, pp. 1103–1127, 2000.
[34] "MPEG Working Group." [Online]. Available: http://mpeg.chiariglione.org/. [Accessed: 14-Oct-2013].
[35] "ITU Telecommunication Standardization Sector," *ITU*. [Online]. Available: http://www.itu.int/en/ITU-T/Pages/default.aspx. [Accessed: 17-Oct-2013].
[36] ITU-T Recommendations, "H.264 : Advanced video coding for generic audiovisual services," Jan-2012. [Online]. Available: http://www.itu.int/rec/T-REC-H.264. [Accessed: 18-Mar-2013].
[37] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[38] H. Schulzrinne, "Real Time Streaming Protocol (RTSP)." [Online]. Available: http://tools.ietf.org/html/rfc2326. [Accessed: 31-Oct-2012].
[39] V. Jacobson, R. Frederick, S. Casner, and H. Schulzrinne, "RFC 3550: RTP: A Transport Protocol for Real-Time Applications," *IETF*, 2003.
[40] M. Westerlund and I. Johansson, "Support for Reduced-Size Real-Time Transport Control Protocol (RTCP): Opportunities and Consequences." [Online]. Available: http://svn.tools.ietf.org/html/rfc5506. [Accessed: 24-May-2012].
[41] J. Postel, "User Datagram Protocol." [Online]. Available: http://merlot.tools.ietf.org/html/rfc768. [Accessed: 24-May-2012].
[42] K. Egevang and P. Francis, "RFC 1631: The IP Network Address Translator (NAT)," 1994.
[43] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, "Multimedia streaming via TCP: An analytic performance study," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 4, no. 2, pp. 16:1–16:22, May 2008.
[44] M. Allman, V. Paxson, and E. Blanton, "RFC 5681: TCP Congestion Control," *IETF*, 2009.
[45] E. Kohler, M. Handley, and S. Floyd, "RFC 4340: Datagram Congestion Control Protocol (DCCP)," *IETF*, Mar. 2006.
[46] R. Stewart, Ed., "RFC 4960: Stream Control Transmission Protocol," 2007. [Online]. Available: http://www.hjp.at/doc/rfc/rfc4960.html. [Accessed: 13-Mar-2013].
[47] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "RFC 6824: TCP Extensions for Multipath Operation with Multiple Addresses," *IETF*, Jan. 2013.
[48] J. Postel, "Internet Protocol." [Online]. Available: http://tools.ietf.org/html/rfc791. [Accessed: 24-May-2012].
[49] S. E. Deering and R. Hinden, "Internet protocol, version 6 (IPv6) specification," 1998.
[50] Y. Xiao, "Performance analysis of priority schemes for IEEE 802.11 and IEEE 802.11e wireless LANs," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1506 – 1515, Jul. 2005.
[51] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," in *Proceedings of the ACM SIGCOMM 2011 conference*, New York, NY, USA, 2011, pp. 362–373.
[52] N. Bhatti, A. Bouch, and A. Kuchinsky, "Integrating user-perceived quality into Web server design," *Computer Networks*, vol. 33, no. 1–6, pp. 1–16, Jun. 2000.
[53] D. F. Galletta, R. M. Henry, S. McCoy, and P. Polak, "When the Wait Isn't So Bad: The Interacting Effects of Website Delay, Familiarity, and Breadth," *Information Systems Research*, vol. 17, no. 1, pp. 20–37, Mar. 2006.

[54] S. R. Gulliver and G. Ghinea, "Defining user perception of distributed multimedia quality," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 2, no. 4, pp. 241–257, Nov. 2006.

[55] F. F.-H. Nah, "A study on tolerable waiting time: how long are Web users willing to wait?," *Behaviour & Information Technology*, vol. 23, no. 3, pp. 153–163, 2004.

[56] "Keynote Systems – The Mobile and Internet Performance Authority." [Online]. Available: http://www.keynote.com/. [Accessed: 03-Dec-2012].

[57] I. T. S. ITU-T RECOMMENDATION, "Vocabulary for performance and quality of service Amendment2: New definitions for inclusion in Recommendation ITU-T P.10/G.100," Jul. 2008.

[58] Jumisko-Pyykkö, V. K. Malamal Vadakital, and M. M. Hannuksela, "Acceptance Threshold: A Bidimensional Research Method for User-Oriented Quality Evaluation Studies," *International Journal of Digital Multimedia Broadcasting*, vol. 2008, p. e712380, Aug. 2008.

[59] P. Le Callet, S. Moller, and A. Perkis, Eds., "Qualinet White Paper on Definitions of Quality of Experience," *European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003)*, 2012.

[60] R. Schatz, T. Hoßfeld, L. Janowski, and S. Egger, "From Packets to People: Quality of Experience as a New Measurement Challenge," in *Data Traffic Monitoring and Analysis*, E. Biersack, C. Callegari, and M. Matijasevic, Eds. Springer Berlin Heidelberg, 2013, pp. 219–263.

[61] C. H. Muntean and J. McManis, "End-User Quality of Experience Oriented Adaptive E-learning System," *Journal Of Digital Information*, vol. 7, no. 1, 2006.

[62] G. Ghinea and J. P. Thomas, "Quality of Perception: User Quality of Service in Multimedia Presentations," *Trans. Multi.*, vol. 7, no. 4, pp. 786–789, Aug. 2005.

[63] C. H. Muntean, "Improving learner quality of experience by content adaptation based on network conditions," *Computers in Human Behavior*, vol. 24, no. 4, pp. 1452–1472, Jul. 2008.

[64] E. Comer Douglas, *Computer Networks and Internets*, 2nd ed. Prentice-Hall, NJ, 1999.

[65] A. Sears, J. A. Jacko, and M. S. Borella, "Internet delay effects: how users perceive quality, organization, and ease of use of information," in *CHI '97 extended abstracts on Human factors in computing systems: looking to the future*, New York, NY, USA, 1997, pp. 353–354.

[66] "ITU-T G.1010 : End-user multimedia QoS categories," 2001.

[67] S. Egger, P. Reichl, T. Hosfeld, and R. Schatz, "'Time is bandwidth'? Narrowing the gap between subjective time perception and Quality of Experience," in *2012 IEEE International Conference on Communications (ICC)*, 2012, pp. 1325–1330.

[68] D. F. Galletta, R. Henry, S. McCoy, and P. Polak, "Web Site Delays: How Tolerant are Users?," *Journal of the Association for Information Systems*, vol. 5, no. 1, Jan. 2004.

[69] A. Sears and J. A. Jacko, "Understanding the relation between network quality of service and the usability of distributed multimedia documents," *Hum.-Comput. Interact.*, vol. 15, no. 1, pp. 43–68, Mar. 2000.

[70] P. Reichl, S. Egger, R. Schatz, and A. D'Alconzo, "The Logarithmic Nature of QoE and the Role of the Weber-Fechner Law in QoE Assessment," in *2010 IEEE International Conference on Communications (ICC)*, 2010, pp. 1–5.

[71] X. Tan, J. Gustafsson, and G. Heikkilä, "Perceived Video Streaming Quality under Initial Buffering and Rebuffering Degradations," presented at the International Conference MESAQIN Measurement of Speech, Audio and Video Quality in Networks, Prague, Czech Republic, 2006.

[72] G. Ghinea and J. P. Thomas, "Crossing the Man-Machine Divide: A Mapping Based on Empirical Results," *The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, vol. 29, no. 1–2, pp. 139–147, Aug. 2001.

[73] G. Ghinea and S. Y. Chen, "The impact of cognitive styles on perceptual distributed multimedia quality," *British Journal of Educational Technology*, vol. 34, no. 4, pp. 393–406, Sep. 2003.

[74] E. S. Davis and D. A. Hantula, "The effects of download delay on performance and end-user satisfaction in an Internet tutorial," *Computers in Human Behavior*, vol. 17, no. 3, pp. 249–268, May 2001.

[75] C. H. Muntean and J. McManis, "The Value of QoE-Based Adaptation Approach in Educational Hypermedia: Empirical Evaluation," in *Adaptive Hypermedia and Adaptive Web-Based Systems*, V. P. Wade, H. Ashman, and B. Smyth, Eds. Springer Berlin Heidelberg, 2006, pp. 121–130.

[76] C. H. Muntean and J. McManis, "End-User Quality of Experience Layer for Adaptive Hypermedia Systems," presented at the Int. Conf. on Adaptive Hypermedia and Web-based Systems, Eindhoven, Netherlands, 2004.

[77] P. H. Tipton, M. Pulliam, S. H. Allen, and C. Sherwood, "Lessons learned: Pointers for successfully teaching via videoconferencing," *Teaching and Learning in Nursing*, vol. 6, no. 1, pp. 27–30, Jan. 2011.

[78] P. Selvidge, "How long is too long to wait for a website to load," *Usability news*, vol. 1, no. 2, 1999.

[79] P. R. Selvidge, B. S. Chaparro, and G. T. Bender, "The world wide wait: effects of delays on user performance," *International Journal of Industrial Ergonomics*, vol. 29, no. 1, pp. 15–20, Jan. 2002.

[80] J. Crosbie and G. Kelly, "Effects of imposed postfeedback delays in programmed instruction," *J Appl Behav Anal*, vol. 27, no. 3, pp. 483–491, 1994.

[81] G. Kelly and J. Crosbie, "Immediate and Delayed Effects of Imposed Postfeedback Delays in Computerized Programmed Instruction," *The Psychological Record*, vol. 47, no. 4, 1997.

[82] G. J. Gorn, A. Chattopadhyay, J. Sengupta, and S. Tripathi, "Waiting for the Web: How Screen Color Affects Time Perception," *Journal of Marketing Research*, vol. 41, no. 2, pp. 215–225, May 2004.

[83] P. Fajfar, G. Campitelli, and M. Labollita, "Effects of immediacy of feedback on estimations and performance," *Australian Journal of Psychology*, vol. 64, no. 3, pp. 169–177, 2012.

[84] A. W. Salmoni, R. A. Schmidt, and C. B. Walter, "Knowledge of results and motor learning: A review and critical reappraisal," *Psychological Bulletin*, vol. 95, no. 3, pp. 355–386, 1984.

[85] "ITU-T Rec Y.1541 : Network performance objectives for IP-based services," 2011.

[86] 3GPP, "Specification: 23.207 End-to-end Quality of Service (QoS) concept and architecture." [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/23207.htm. [Accessed: 28-Oct-2013].

[87] C. Alberti, D. Renzi, C. Timmerer, C. Mueller, S. Lederer, S. Battista, and M. Mattavelli, "Automated QoE Evaluation of Dynamic Adaptive Streaming Over HTTP," presented at the Fifth International Workshop on Quality of Multimedia Experience (QoMEX), Klagenfurt, Austria, 2013.

[88] J. Shaikh, M. Fiedler, and D. Collange, "Quality of Experience from user and network perspectives," *Ann. Telecommun.*, vol. 65, no. 1–2, pp. 47–57, Feb. 2010.

[89] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600 –612, Apr. 2004.

[90] S. Chikkerur, V. Sundaram, M. Reisslein, and L. J. Karam, "Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison," *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 165–182, 2011.

[91] ITU Recommendation P800, "Methods for Subjective Determination of Transmission Quality," International Telecommunication Union, 1996.

[92] ITU Recommendation BT.500-13, "Methodology for the subjective assessment of the quality of television pictures," 2012.

[93] ITU Recommendation P.910, "Subjective video quality assessment methods for multimedia applications," 2008.

[94] T. Hossfeld, M. Hirth, P. Korshunov, P. Hanhart, B. Gardlo, C. Keimel, and C. Timmerer, "Survey of Web-based Crowdsourcing Frameworks for Subjective Quality Assessment," presented at the 16th International Workshop on Multimedia Signal Processing, 2014.

[95] A. Begen, T. Akgul, and M. Baugher, "Watching Video over the Web: Part 2: Applications, Standardization, and Open Issues," *IEEE Internet Computing*, vol. 15, no. 3, pp. 59 –63, Jun. 2011.

[96] "Progressive download | Adobe Developer Connection." [Online]. Available: http://www.adobe.com/devnet/video/progressive.html. [Accessed: 23-Aug-2012].

[97] S. Xiang, L. Cai, and J. Pan, "Adaptive scalable video streaming in wireless networks," in *Proceedings of the 3rd Multimedia Systems Conference*, New York, NY, USA, 2012, pp. 167–172.

[98] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP–Design Principles and Standards," presented at the The second annual ACM conference on Multimedia Systems (MMSys11), New York, NY, USA, 2011, pp. 133–144.

[99] "Adobe HTTP Dynamic Streaming * Using Adobe HTTP Dynamic Streaming." [Online]. Available: http://help.adobe.com/en_US/HTTPStreaming/1.0/Using/index.html. [Accessed: 23-Aug-2012].

[100] R. Pantos, "HTTP live streaming," 2011. [Online]. Available: http://tools.ietf.org/html/pantos-http-live-streaming-07). [Accessed: 23-Aug-2012].

[101] "Microsoft Smooth Streaming." [Online]. Available: http://www.iis.net/download/smoothstreaming. [Accessed: 23-Aug-2012].

[102] D. Evans, "Look, no plugins: Live streaming to the browser using Media Source Extensions and MPEG-DASH - Blog - BBC R&D," *BBC R&D Blog*, 17-Mar-2014. .

[103] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE MultiMedia*, vol. 18, no. 4, pp. 62 –67, Apr. 2011.

[104] N. Ozbek and E. T. Tunali, "On optimal receiver buffer size in adaptive Internet video streaming," *J. Zhejiang Univ. - Sci. A*, vol. 7, no. 1, pp. 112–118, Jan. 2006.

[105] A. Zambelli, “IIS smooth streaming technical overview,” *Microsoft Corporation*, 2009.
[106] S. Akhshabi, A. C. Begen, and C. Dovrolis, “An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP,” in *Proceedings of the second annual ACM conference on Multimedia systems*, New York, NY, USA, 2011, pp. 157–168.
[107] “Netflix - Watch TV Shows Online, Watch Movies Online.” [Online]. Available: https://www.netflix.com. [Accessed: 14-Dec-2014].
[108] B. Krogfoss, A. Agrawal, and L. Sofman, “Analytical method for objective scoring of HTTP Adaptive Streaming (HAS),” in *2012 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2012, pp. 1–6.
[109] M.-L. Alberi Morel, S. Kerboeuf, B. Sayadi, Y. Leprovost, and F. Faucheux, “Performance Evaluation of Channel Change for DVB-SH Streaming Services,” presented at the 2010 IEEE International Conference on Communications (ICC), Cape Town, South Africa, 2010, pp. 1–6.
[110] D. Jarnikov and T. Özçelebi, “Client intelligence for adaptive streaming solutions,” *Signal Processing: Image Communication*, vol. 26, no. 7, pp. 378–389, Aug. 2011.
[111] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, “Adaptation algorithm for adaptive streaming over HTTP,” in *Packet Video Workshop (PV), 2012 19th International*, Munich, Germany, 2012, pp. 173–178.
[112] T. C. Thang, Q.-D. Ho, J. W. Kang, and A. T. Pham, “Adaptive streaming of audiovisual content using MPEG DASH,” *IEEE Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 78 –85, Feb. 2012.
[113] T. Lohmar, T. Einarsson, P. Frojdh, F. Gabin, and M. Kampmann, “Dynamic adaptive HTTP streaming of live content,” in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, Lucca, Italy, 2011, pp. 1–8.
[114] J. Case, M. Fedor, M. Schoffstall, and J. Davin, “RFC 1157: A Simple Network Management Protocol (SNMP),” *lETF*, May 1990.
[115] J. Sommers, P. Barford, N. Duffield, and A. Ron, “Accurate and efficient SLA compliance monitoring,” in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, New York, NY, USA, 2007, pp. 109–120.
[116] X. Luo, E. W. W. Chan, and R. K. C. Chang, “Design and implementation of TCP data probes for reliable and metric-rich network path monitoring,” in *Proceedings of the 2009 conference on USENIX Annual technical conference*, Berkeley, CA, USA, 2009, pp. 4–4.
[117] S. Lederer, C. Muller, and C. Timmerer, “Towards peer-assisted dynamic adaptive streaming over HTTP,” in *Packet Video Workshop (PV), 2012 19th International*, 2012, pp. 161 –166.
[118] B. Rainer, S. Lederer, C. Muller, and C. Timmerer, “A seamless Web integration of adaptive HTTP streaming,” presented at the Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European, Bucharest, Romania, 2012, pp. 1519 –1523.
[119] C. Liu, I. Bouazizi, and M. Gabbouj, “Rate adaptation for adaptive HTTP streaming,” in *Proceedings of the second annual ACM conference on Multimedia systems*, New York, NY, USA, 2011, pp. 169–174.
[120] S. Gouache, G. Bichot, A. Bsila, and C. Howson, “Distributed & Adaptive HTTP Streaming,” presented at the 2011 IEEE International Conference on Multimedia and Expo (ICME), Barcelona, Spain, 2011, pp. 1–6.
[121] L. Rubio Romero, “A Dynamic Adaptive HTTP Streaming Video Service for Google Android,” 2011.
[122] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, “QDASH: a QoE-aware DASH system,” in *Proceedings of the 3rd Multimedia Systems Conference*, New York, NY, USA, 2012, pp. 11–22.
[123] K. Evensen, D. Kaspar, C. Griwodz, P. Halvorsen, A. F. Hansen, and P. Engelstad, “Using bandwidth aggregation to improve the performance of quality-adaptive streaming,” *Signal Processing: Image Communication*, vol. 27, no. 4, pp. 312–328, Apr. 2012.
[124] S. Lederer, C. Mueller, C. Timmerer, C. Concolato, J. Le Feuvre, and K. Fliegel, “Distributed DASH dataset,” in *Proceedings of the 4th ACM Multimedia Systems Conference*, New York, NY, USA, 2013, pp. 131–135.
[125] D. Suh, I. Jang, and S. Pack, “QoE-enhanced adaptation algorithm over DASH for multimedia streaming,” in *2014 International Conference on Information Networking (ICOIN)*, 2014, pp. 497–501.
[126] V. Joseph and G. de Veciana, “NOVA: QoE-driven optimization of DASH-based video delivery in networks,” in *2014 Proceedings IEEE INFOCOM*, 2014, pp. 82–90.
[127] 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.

[128] J. Yao, S. S. Kanhere, I. Hossain, and M. Hassan, “Empirical Evaluation of HTTP Adaptive Streaming under Vehicular Mobility,” in *NETWORKING 2011*, J. Domingo-Pascual, P. Manzoni, S. Palazzo, A. Pont, and C. Scoglio, Eds. Springer Berlin Heidelberg, 2011, pp. 92–105.

[129] J. Kovacevic, G. Miljkovic, K. Lazic, and M. Stankic, “Evaluation of adaptive streaming algorithms over HTTP,” presented at the IEEE International Conference on Consumer Electronics (ICCE), Berlin, 2013, pp. 1–4.

[130] G. Tian and Y. Liu, “Towards Agile and Smooth Video Adaptation in Dynamic HTTP Streaming,” in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, New York, NY, USA, 2012, pp. 109–120.

[131] M. Goldmann and G. Kreitz, “Measurements on the spotify peer-assisted music-on-demand streaming system,” in *2011 IEEE International Conference on Peer-to-Peer Computing (P2P)*, 2011, pp. 206 –211.

[132] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack, “Study of Subjective and Objective Quality Assessment of Video,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1427–1441, Jun. 2010.

[133] S.-B. Lee, G. Muntean, and A. F. Smeaton, “Performance-Aware Replication of Distributed Pre-Recorded IPTV Content,” *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 516–526, 2009.

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

[135] 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,” *IEEE Transactions on Broadcasting*, vol. 59, no. 1, pp. 47–61, Mar. 2013.

[136] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang, “A case for a coordinated internet video control plane,” in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, New York, NY, USA, 2012, pp. 359–370.

[137] T. Hossfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, “Initial delay vs. interruptions: Between the devil and the deep blue sea,” in *2012 Fourth International Workshop on Quality of Multimedia Experience (QoMEX)*, 2012, pp. 1–6.

[138] S. G. Dykes, C. L. Jeffery, and K. A. Robbins, “An empirical evaluation of client-side server selection algorithms,” in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2000, vol. 3, pp. 1361–1370.

[139] R. M. Hinden and S. E. Deering, “RFC 4291: IP Version 6 Addressing Architecture,” *IETF*, Feb. 2006.

[140] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao, “Dissecting Video Server Selection Strategies in the YouTube CDN,” in *2011 31st International Conference on Distributed Computing Systems (ICDCS)*, 2011, pp. 248–257.

[141] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford, “DONAR: decentralized server selection for cloud services,” in *Proceedings of the ACM SIGCOMM 2010 conference*, New York, NY, USA, 2010, pp. 231–242.

[142] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, “QoS-aware middleware for Web services composition,” *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, May 2004.

[143] M. Alrifai and T. Risse, “Combining Global Optimization with Local Selection for Efficient QoS-aware Service Composition,” in *Proceedings of the 18th International Conference on World Wide Web*, New York, NY, USA, 2009, pp. 881–890.

[144] T. Yu, Y. Zhang, and K.-J. Lin, “Efficient Algorithms for Web Services Selection with End-to-end QoS Constraints,” *ACM Trans. Web*, vol. 1, no. 1, May 2007.

[145] A. Metzger, O. Sammodi, and K. Pohl, “Accurate Proactive Adaptation of Service-Oriented Systems,” in *Assurances for Self-Adaptive Systems*, J. Cámara, R. de Lemos, C. Ghezzi, and A. Lopes, Eds. Springer Berlin Heidelberg, 2013, pp. 240–265.

[146] B. Cavallo, M. Di Penta, and G. Canfora, “An Empirical Comparison of Methods to Support QoS-aware Service Selection,” in *Proceedings of the 2Nd International Workshop on Principles of Engineering Service-Oriented Systems*, New York, NY, USA, 2010, pp. 64–70.

[147] V. K. Adhikari, S. Jain, and Z.-L. Zhang, “YouTube traffic dynamics and its interplay with a tier-1 ISP: an ISP perspective,” in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, New York, NY, USA, 2010, pp. 431–443.

[148] V. K. Adhikari, S. Jain, Y. Chen, and Z.-L. Zhang, “Vivisecting YouTube: An active measurement study,” in *2012 Proceedings IEEE INFOCOM*, 2012, pp. 2521–2525.

[149] V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, “Unreeling Netflix: Understanding and improving multi-cdn movie delivery,” in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 1620–1628.

[150] R. de Lemos, H. Giese, H. A. Müller, M. Shaw, J. Andersson, M. Litoiu, B. Schmerl, G. Tamura, N. M. Villegas, T. Vogel, D. Weyns, L. Baresi, B. Becker, N. Bencomo, Y. Brun, B. Cukic, R. Desmarais, S. Dustdar, G. Engels, K. Geihs, K. M. Göschka, A. Gorla, V. Grassi, P. Inverardi, G. Karsai, J. Kramer, A. Lopes, J. Magee, S. Malek, S. Mankovskii, R. Mirandola, J. Mylopoulos, O. Nierstrasz, M. Pezzè, C. Prehofer, W. Schäfer, R. Schlichting, D. B. Smith, J. P. Sousa, L. Tahvildari, K. Wong, and J. Wuttke, "Software Engineering for Self-Adaptive Systems: A Second Research Roadmap," in *Software Engineering for Self-Adaptive Systems II*, R. de Lemos, H. Giese, H. A. Müller, and M. Shaw, Eds. Springer Berlin Heidelberg, 2013, pp. 1–32.

[151] D. Weyns, S. Malek, and J. Andersson, "FORMS: A Formal Reference Model for Self-adaptation," in *Proceedings of the 7th International Conference on Autonomic Computing*, New York, NY, USA, 2010, pp. 205–214.

[152] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, Jan. 2003.

[153] R. Bruni, A. Corradini, F. Gadducci, A. L. Lafuente, and A. Vandin, "A Conceptual Framework for Adaptation," in *Fundamental Approaches to Software Engineering*, J. de Lara and A. Zisman, Eds. Springer Berlin Heidelberg, 2012, pp. 240–254.

[154] J. O. Kephart and R. Das, "Achieving Self-Management via Utility Functions," *IEEE Internet Computing*, vol. 11, no. 1, pp. 40–48, Jan. 2007.

[155] M. Butler, F. Giannetti, R. Gimson, and T. Wiley, "Device independence and the Web," *Internet Computing, IEEE*, vol. 6, no. 5, pp. 81 – 86, Oct. 2002.

[156] F. Manola, E. Miller, and B. McBride, "RDF primer," *W3C recommendation*, vol. 10, pp. 1–107, 2004.

[157] H. Ogata and Y. Yano, "Context-aware support for computer-supported ubiquitous learning," in *The 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education, 2004. Proceedings*, 2004, pp. 27 – 34.

[158] "Device Independence Principles." [Online]. Available: http://www.w3.org/TR/di-princ/. [Accessed: 01-Jun-2012].

[159] D. Corlett, M. Sharples, S. Bull, and T. Chan, "Evaluation of a mobile learning organiser for university students," *Journal of Computer Assisted Learning*, vol. 21, no. 3, pp. 162–170, 2005.

[160] C. Bradley, R. Haynes, and T. Boyle, "Adult multimedia learning with PDAs-the user experience," in *4th World Conference on mLearning*, Cape Town, South Africa, 2005.

[161] S. So, "A Study on the Acceptance of Mobile Phones for Teaching and Learning with a group of Pre-service teachers in Hong Kong," *Journal of Educational Technology Development and Exchange*, vol. 1, no. 1, pp. 81–92, 2008.

[162] "Composite Capabilities/Preferences Profile Home Page," *W3C*. [Online]. Available: http://www.w3.org/Mobile/CCPP/. [Accessed: 01-Jan-2015].

[163] O. M. Alliance, "User Agent Profile approved Version 2.0," *Open Mobile Alliance*, vol. 1, no. OMA-TS-UAProf-V2_0–20060206-A, 2006.

[164] T. Glover and J. Davies, "Integrating device independence and user profiles on the Web," *BT Technol J*, vol. 23, no. 3, pp. 239–248, Jul. 2005.

[165] "UPnP Forum." [Online]. Available: http://www.upnp.org/. [Accessed: 01-Jun-2012].

[166] L. Passani, "WURFL - Mobile Device Database by ScientiaMobile." [Online]. Available: http://wurfl.sourceforge.net/. [Accessed: 13-Dec-2012].

[167] S. Y. Tsai, "Device Profiling Analysis in Device-Aware Network," Monterey, California Naval Postgraduate School, 2004.

[168] R. H. P. Lima, M. M. El-Bishouty, and S. Graf, "A Framework for Automatic Identification and Visualization of Mobile Device Functionalities and Usage," in *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*, A. Holzinger and G. Pasi, Eds. Springer Berlin Heidelberg, 2013, pp. 148–159.

[169] M. F. Ibrahim, S. Yahya, and M. N. Taib, "Device Characteristics and Capabilities Discovery for Multimedia Content," *Journal of Computer Networks and Communications*, vol. 2012, pp. 1–15, 2012.

[170] L. Naismith, P. Lonsdale, G. N. Vavoula, and M. Sharples, "Mobile technologies and learning," *Futurelab Literature Review Series, Report No 11*, Dec. 2004.

[171] S. Jones, M. Jones, and S. Deo_andA2, "Using keyphrases as search result surrogates on small screen devices," *Personal Ubiquitous Comput.*, vol. 8, no. 1, pp. 55–68, Feb. 2004.

[172] T. Goh and D. Kinshuk, "Getting ready for mobile learning-adaptation perspective," *Journal of Educational Multimedia and Hypermedia*, vol. 15, no. 2, p. 175, 2006.

[173] C. H. Muntean, G.-M. Muntean, J. McManis, and A. I. Cristea, "Quality of Experience-LAOS: create once, use many, use anywhere," *International Journal of Learning Technology*, vol. 3, no. 3, pp. 209–229, Jan. 2007.

[174] L. D. Cicco and S. Mascolo, “An Experimental Investigation of the Akamai Adaptive Video Streaming,” in *HCI in Work and Learning, Life and Leisure*, G. Leitner, M. Hitz, and A. Holzinger, Eds. Springer Berlin Heidelberg, 2010, pp. 447–464.

[175] P. Stacey, “Open educational resources in a global context,” *First Monday*, vol. 12, no. 4, Apr. 2007.

[176] P. Brusilovsky, “Methods and techniques of adaptive hypermedia,” *User Modeling and User-Adapted Interaction*, vol. 6, no. 2, pp. 87–129, 1996.

[177] “IEEE Standard for Learning Object Metadata,” *IEEE Std 1484.12.1-2002*, pp. i–32, 2002.

[178] “SCORM 2004,” *Advanced Distributed Learning Initiative*.

[179] S.-S. Tseng, J.-M. Su, G.-J. Hwang, G.-H. Hwang, C.-C. Tsai, and C.-J. Tsai, “An object-oriented course framework for developing adaptive learning systems,” *Educational Technology & Society*, vol. 11, no. 2, pp. 171–191, 2008.

[180] D. Benyon and D. Murray, “Adaptive systems: from intelligent tutoring to autonomous agents,” *Knowledge-Based Systems*, vol. 6, no. 4, pp. 197–219, Dec. 1993.

[181] L. Van Velsen, T. Van Der Geest, R. Klaassen, and M. Steehouder, “User-centered evaluation of adaptive and adaptable systems: a literature review,” *The Knowledge Engineering Review*, vol. 23, no. 03, pp. 261–281, 2008.

[182] A. I. Cristea and A. De Mooij, “LAOS: Layered WWW AHS authoring model and their corresponding algebraic operators,” in *WWW03 (The Twelfth International World Wide Web Conference), Alternate Track on Education, Budapest, Hungary*, 2003.

[183] P. Brusilovsky, “Adaptive Hypermedia for Education and Training,” in *Adaptive Technologies for Training and Education*, P. J. Durlach and A. M. Lesgold, Eds. Cambridge University Press, 2012, p. 46.

[184] A. Moore, V. Macarthur, and O. Conlan, “Core Aspects of Affective Metacognitive User Models,” in *Advances in User Modeling*, vol. 7138, L. Ardissono and T. Kuflik, Eds. Springer Berlin / Heidelberg, 2012, pp. 47–59.

[185] P. Brusilovsky, S. Sosnovsky, and O. Shcherbinina, “User modeling in a distributed e-learning architecture,” *User Modeling 2005*, pp. 151–151, 2005.

[186] M. Yudelson, P. Brusilovsky, and V. Zadorozhny, “A User Modeling Server for Contemporary Adaptive Hypermedia: An Evaluation of the Push Approach to Evidence Propagation,” in *Proceedings of the 11th international conference on User Modeling*, Berlin, Heidelberg, 2007, pp. 27–36.

[187] P. Brusilovsky, S. Sosnovsky, M. V. Yudelson, D. H. Lee, V. Zadorozhny, and X. Zhou, “Learning SQL Programming with Interactive Tools: From Integration to Personalization,” *Trans. Comput. Educ.*, vol. 9, no. 4, pp. 19:1–19:15, Jan. 2010.

[188] A. Moore, T. J. Brailsford, and C. D. Stewart, “Personally tailored teaching in WHURLE using conditional transclusion,” in *Proceedings of the 12th ACM conference on Hypertext and Hypermedia*, New York, NY, USA, 2001, pp. 163–164.

[189] A. Cristea, C. Stewart, T. Brailsford, and P. Cristea, “Adaptive Hypermedia System Interoperability: a ‘real world’ evaluation,” *Journal of Digital Information*, vol. 8, no. 3, Oct. 2007.

[190] B. A. Soloman and R. M. Felder, “Index of Learning Styles Questionnaire.” [Online]. Available: http://www.engr.ncsu.edu/learningstyles/ilsweb.html. [Accessed: 15-Nov-2012].

[191] P. Brusilovsky and M. T. Maybury, “From adaptive hypermedia to the adaptive web,” *Commun. ACM*, vol. 45, no. 5, pp. 30–33, May 2002.

[192] P. De Bra, D. Smits, and N. Stash, “The Design of AHA!,” in *Proceedings of the Seventeenth Conference on Hypertext and Hypermedia*, New York, NY, USA, 2006, pp. 133–134.

[193] P. Brusilovsky, “Adaptive hypermedia,” *User modeling and user-adapted interaction*, vol. 11, no. 1, pp. 87–110, 2001.

[194] E. Knutov, P. De Bra, and M. Pechenizkiy, “AH 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques,” *New Review of Hypermedia and Multimedia*, vol. 15, no. 1, pp. 5–38, 2009.

[195] P. Brusilovsky, “Adaptive Navigation Support,” in *The Adaptive Web*, vol. 4321, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Springer Berlin / Heidelberg, 2007, pp. 263–290.

[196] D. Dagger, A. O’Connor, S. Lawless, E. Walsh, and V. P. Wade, “Service-Oriented E-Learning Platforms: From Monolithic Systems to Flexible Services,” *IEEE Internet Computing*, vol. 11, no. 3, pp. 28–35, May 2007.

[197] O. CONLAN and V. P. WADE, “Context-informed adaptive hypermedia,” 2004.

[198] P. Brusilovsky, “KnowledgeTree: A distributed architecture for adaptive e-learning,” in *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, 2004, pp. 104–113.

[199] I. O’Keeffe and V. Wade, “Personalised Web Experiences: Seamless Adaptivity across Web Service Composition and Web Content,” in *User Modeling, Adaptation, and Personalization*, vol.

5535, G.-J. Houben, G. McCalla, F. Pianesi, and M. Zancanaro, Eds. Springer Berlin / Heidelberg, 2009, pp. 480–485.

[200] Y. Morimoto, M. Ueno, I. Kikukawa, S. Yokoyama, and Y. Miyadera, “SALMS: SCORM-compliant Adaptive LMS,” *ELEARN 2007*, vol. 2007, no. 1, pp. 7287–7296.

[201] P. Brusilovsky and N. Henze, “Open Corpus Adaptive Educational Hypermedia,” in *The Adaptive Web*, vol. 4321, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Springer Berlin / Heidelberg, 2007, pp. 671–696.

[202] N. Henze and W. Nejdl, “Adaptation in open corpus hypermedia,” *International Journal of Artificial Intelligence in Education*, vol. 12, no. 4, pp. 325–350, 2001.

[203] C. Bailey, W. Hall, D. E. Millard, and M. J. Weal, “Towards Open Adaptive Hypermedia,” in *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, London, UK, UK, 2002, pp. 36–46.

[204] N. Henze, “Personal Readers: Personalized Learning Object Readers for the Semantic Web,” in *Proceedings of the 2005 conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, Amsterdam, The Netherlands, The Netherlands, 2005, pp. 274–281.

[205] C. Carmona, D. Bueno, E. Guzman, and R. Conejo, “SIGUE: Making Web Courses Adaptive,” in *Adaptive Hypermedia and Adaptive Web-Based Systems*, P. D. Bra, P. Brusilovsky, and R. Conejo, Eds. Springer Berlin Heidelberg, 2006, pp. 376–379.

[206] P. Brusilovsky, G. Chavan, and R. Farzan, “Social Adaptive Navigation Support for Open Corpus Electronic Textbooks,” in *Adaptive Hypermedia and Adaptive Web-Based Systems*, P. M. E. D. Bra and W. Nejdl, Eds. Springer Berlin Heidelberg, 2004, pp. 24–33.

[207] S. Lawless and V. Wade, “Dynamic content discovery, harvesting and delivery, from open corpus sources, for adaptive systems,” in *Adaptive Hypermedia and Adaptive Web-Based Systems*, 2006, pp. 445–451.

[208] A. Alvarez, S. Ruiz, M. Martín, I. F. de Castro, and M. Urretavizcaya, “MAGADI: a Blended-Learning Framework for Overall Learning.,” in *International Conference on Artificial Intelligence in Education (AIED’09)*, 2009, pp. 557–564.

[209] A. Álvarez, M. Martín, I. Fernández-Castro, and M. Urretavizcaya, “Blending traditional teaching methods with learning environments: Experience, cyclical evaluation process and impact with MAgAdI,” *Computers & Education*, vol. 68, pp. 129–140, Oct. 2013.

[210] W.-S. Lin and C.-H. Wang, “Antecedences to continued intentions of adopting e-learning system in blended learning instruction: A contingency framework based on models of information system success and task-technology fit,” *Computers & Education*, vol. 58, no. 1, pp. 88–99, Jan. 2012.

[211] G. Nuzzo-Jones, J. A. Walonoski, N. T. Heffernan, and T. Livak, “The eXtensible Tutor Architecture: A New Foundation for ITS.,” presented at the Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology, 2005, pp. 902–904.

[212] M. Trella, C. Carmona, and R. Conejo, “MEDEA: An open service-based learning platform for developing intelligent educational systems for the Web,” presented at the Workshop on Adaptive Systems for Web-based Education at 12th International Conference on Artificial Intelligence in Education (AIED’05), 2005, pp. 27–34.

[213] S. M. Alessi and S. R. Trollip, “Multimedia for learning,” 2001.

[214] J. Baggaley, “The sudden revival of educational video,” in *Educational Media (ICEM), 2013 IEEE 63rd Annual Conference International Council for*, 2013, pp. 1–6.

[215] “ALISON,” *Advance Learning Interactive Systems Online*. [Online]. Available: http://alison.com/. [Accessed: 11-Jun-2014].

[216] “MIT OpenCourseWare.” [Online]. Available: http://ocw.mit.edu/index.htm. [Accessed: 13-Oct-2013].

[217] P. J. Guo, J. Kim, and R. Rubin, “How Video Production Affects Student Engagement: An Empirical Study of MOOC Videos,” in *Proceedings of the First ACM Conference on Learning @ Scale Conference*, New York, NY, USA, 2014, pp. 41–50.

[218] A. K. Dey, “Understanding and using context,” *Personal and ubiquitous computing*, vol. 5, no. 1, pp. 4–7, 2001.

[219] A. Goker and H. I. Myrhaug, “User context and personalisation,” in *Workshop on Case Based Reasoning and Personalization*, Scotland, 2002.

[220] D. Mérida, R. Fabregat, X. Prat, D. Huerva, and J. Velez, “A Dynamic Content Generator for Adaptation in Hypermedia Systems,” in *Adaptive Hypermedia and Adaptive Web-Based Systems*, vol. 5149, W. Nejdl, J. Kay, P. Pu, and E. Herder, Eds. Springer Berlin / Heidelberg, 2008, pp. 320–323.

[221] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni, “A survey of context modelling and reasoning techniques,” *Pervasive and Mobile Computing*, vol. 6, no. 2, pp. 161–180, Apr. 2010.

[222] M. Baldauf, S. Dustdar, and F. Rosenberg, “A survey on context-aware systems,” *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, Jan. 2007.

[223] W.-Y. Ma, I. Bedner, G. Chang, A. Kuchinsky, and H. Zhang, “A framework for adaptive content delivery in heterogeneous network environments,” in *SPIE proceedings series*, 2000, pp. 86–100.

[224] G. Muntean, G. Ghinea, and T. N. Sheehan, “Region of Interest-Based Adaptive Multimedia Streaming Scheme,” *IEEE Transactions on Broadcasting*, vol. 54, no. 2, pp. 296–303, 2008.

[225] B. Ciubotaru, G. Ghinea, and G.-M. Muntean, “Subjective Assessment of Region of Interest-Aware Adaptive Multimedia Streaming Quality,” *IEEE Transactions on Broadcasting*, vol. 60, no. 1, pp. 50–60, Mar. 2014.

[226] C. H. Muntean and J. McManis, “A QoS-aware adaptive Web-based system,” in *2004 IEEE International Conference on Communications*, 2004, vol. 4, pp. 2204 – 2208 Vol.4.

[227] X. Fu and V. Karamcheti, “Performance implications of different adaptation mechanisms for network content delivery,” in *Performance, Computing, and Communications, 2004 IEEE International Conference on*, 2004, pp. 435 – 444.

[228] S. A. Moebs, “A learner, is a learner, is a user, is a customer: QoS-based experience-aware adaptation,” in *Proceedings of the 16th ACM international conference on Multimedia*, New York, NY, USA, 2008, pp. 1035–1038.

[229] D. Dagger, O. Conlan, and V. Wade, “An architecture for candidacy in adaptive eLearning systems to facilitate the reuse of learning Resources,” in *Proc. of World Conference on E-Learning, E-Learn*, 2003, pp. 49–56.

[230] O. Conlan, V. Wade, C. Bruen, and M. Gargan, “Multi-model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning,” in *Adaptive Hypermedia and Adaptive Web-Based Systems*, vol. 2347, P. De Bra, P. Brusilovsky, and R. Conejo, Eds. Springer Berlin / Heidelberg, 2006, pp. 100–111.

[231] A. Brady, O. Conlan, and V. Wade, “Dynamic composition and personalization of PDA-based eLearning–Personalized mLearning,” in *Proceedings of world conference on e-learning in corporate, government, healthcare, and higher education*, 2004, pp. 234–242.

[232] D. Dagger, V. Wade, and O. Conlan, “Towards ‘anytime, anywhere’ Learning: The Role and Realization of Dynamic Terminal Personalization in Adaptive eLearning,” CiteSeerX, 2003.

[233] A. O’Connor, V. Wade, and O. Conlan, “Context-informed adaptive hypermedia,” 2004. [Online]. Available: http://www.tara.tcd.ie/handle/2262/22605. [Accessed: 01-Jun-2012].

[234] D. Mérida, R. Fabregat, and J. L. Marzo, “SHAAD: Adaptable, adaptive and dynamic hypermedia system for content delivery,” in *Workshop on Adaptive Systems for Web Based Education. WASWE2002. Málaga España*, 2002.

[235] D. Huerva, J. Velez, S. Baldiris, R. Fabregat, and D. Merida, “Adaption of Courses and Learning Environment to the User Context in dotLRN,” in *Computational Intelligence for Modelling Control Automation, 2008 International Conference on*, 2008, pp. 1264 –1267.

[236] O. C. Santos, J. G. Boticario, E. Raffenne, and R. Pastor, “Why using dotLRN? UNED use cases,” presented at the FLOSS (Free/Libre/Open Source Systems) International Conference 2007, 2007, pp. 195–212.

[237] P. Lonsdale, C. Barber, M. Sharples, W. Byrne, T. Arvanitis, and P. Brundell, “Context awareness for MOBIlearn: creating an engaging learning experience in an art museum,” in *Mobile learning anytime everywhere: a book of papers from MLEARN 2004*, 2004, pp. 115–118.

[238] P. Lonsdale, C. Barber, M. Sharples, and T. Arvanitis, “A context awareness architecture for facilitating mobile learning,” in *Learning with mobile devices: research and development, a book of papers*, J. Attewell and C. Savill-Smith, Eds. London: Learning and Skills Development Agency, 2004, pp. 79–86.

[239] I. O’Keeffe, A. Brady, O. Conlan, and V. Wade, “Just-in-Time Generation of Pedagogically Sound, Context Sensitive Personalized Learning Experiences,” *International Journal on E-Learning*, vol. 5, no. 1, pp. 113–127, 2006.

[240] X. Zhao, F. Anma, T. Ninomiya, and T. Okamoto, “Personalized Adaptive Content System for Context-Aware Mobile Learning,” *Int. J. of Computer Science and Network Security*, vol. 8, no. 8, pp. 153–161, 2008.

[241] O. C. Santos, J. Granado, E. Raffenne, and J. G. Boticario, “Offering recommendations in OpenACS/dotLRN,” in *Int. Conf. on Community based environments, Valencia*, 2008.

[242] J. Scotton, C. Stewart, and A. I. Cristea, “ADE: The Adaptive Display Environment for Adaptive Hypermedia,” presented at the Conference on Hypertext and Hypermedia, Eindhoven, Netherlands, 2011.

[243] J. Scotton, “Supporting Delivery of Adaptive Hypermedia,” Ph.D., University of Warwick, 2013.

[244] J. G. K. Foss, A. I. Cristea, and M. Hendrix, “Continuous Use of Authoring for Adaptive Educational Hypermedia: A Long-term Case Study,” in *2010 IEEE 10th International Conference on Advanced Learning Technologies (ICALT)*, 2010, pp. 194 –196.

[245] A.-N. Moldovan, C. A. Moraru, and C. H. Muntean, “MediaMTool: Multimedia content management tool,” in *2011 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2011, pp. 1 –6.
[246] P. Dourish and M. Chalmers, “Running out of space: models of information navigation,” in *Short paper presented at HCI*, 1994, vol. 94, pp. 23–26.
[247] J. Dron, R. Mitchell, P. Siviter, and C. Boyne, “CoFIND—An experiment in N-dimensional collaborative filtering,” *Journal of Network and Computer Applications*, vol. 23, no. 2, pp. 131–142, Apr. 2000.
[248] J. Kurhila, M. Miettinen, P. Nokelainen, and H. Tirri, “Educo - A Collaborative Learning Environment Based on Social Navigation,” in *Adaptive Hypermedia and Adaptive Web-Based Systems*, P. D. Bra, P. Brusilovsky, and R. Conejo, Eds. Springer Berlin Heidelberg, 2002, pp. 242–252.
[249] S. Ziovas and M. Grigoriadou, “CRICOS: A Web-Based System for Creating Interconnected Communities,” in *International Symposium on Applications and the Internet, 2008. SAINT 2008*, 2008, pp. 313 –316.
[250] R. Farzan and P. Brusilovsky, “Social Navigation Support Through Annotation-Based Group Modeling,” in *User Modeling 2005*, L. Ardissono, P. Brna, and A. Mitrovic, Eds. Springer Berlin Heidelberg, 2005, pp. 463–472.
[251] J. Freyne, R. Farzan, P. Brusilovsky, B. Smyth, and M. Coyle, “Collecting community wisdom: integrating social search & social navigation,” in *Proceedings of the 12th international conference on Intelligent user interfaces*, 2007, pp. 52–61.
[252] P. Dolog, B. Simon, W. Nejdl, and T. Klobučar, “Personalizing access to learning networks,” *ACM Trans. Internet Technol.*, vol. 8, no. 2, pp. 3:1–3:21, Feb. 2008.
[253] Y. Atif, D. G. Sampson, and S. Harous, “Supporting the Selection of Digital Resources for Classroom-Based Teaching Activities,” in *2013 IEEE 13th International Conference on Advanced Learning Technologies (ICALT)*, 2013, pp. 359–363.
[254] R. Calinescu, K. Johnson, and Y. Rafiq, “Using Observation Ageing to Improve Markovian Model Learning in QoS Engineering,” in *Proceedings of the 2Nd ACM/SPEC International Conference on Performance Engineering*, New York, NY, USA, 2011, pp. 505–510.
[255] M. Meccawy, P. Blanchfield, H. Ashman, T. Brailsford, and A. Moore, “Whurle 2.0: Adaptive learning meets web 2.0,” *Times of Convergence. Technologies Across Learning Contexts*, vol. 5192, pp. 274–279, 2008.
[256] Y. Liu, Y. Guo, and C. Liang, “A survey on peer-to-peer video streaming systems,” *Peer-to-Peer Netw. Appl.*, vol. 1, no. 1, pp. 18–28, Mar. 2008.
[257] T. Berners-Lee, R. Fielding, and L. Masinter, “RFC 3986: Uniform Resource Identifier (URI): Generic Syntax,” *IETF*, Jan. 2005.
[258] D. Mills, U. Delaware, J. Martin, J. Burbank, and W. Kasch, “RFC 5905: Network Time Protocol Version 4: Protocol and Algorithms Specification,” Jun-2010. [Online]. Available: https://tools.ietf.org/html/rfc5905. [Accessed: 12-Jan-2015].
[259] T. Stockhammer and K. R. Fall, “Live timing for dynamic adaptive streaming over http (dash),” US20140195651 A1, 10-Jul-2014.
[260] C. Mueller, S. Lederer, and C. Timmerer, “A proxy effect analysis and fair adaptation algorithm for multiple competing Dynamic Adaptive Streaming over HTTP clients,” in *2012 IEEE Visual Communications and Image Processing (VCIP)*, 2012, pp. 1–6.
[261] I. Stoica, H. Zhang, S. R. McCanne, J. Zhan, and A. R. Ganjam, “Distributing information over a network,” US8775512 B1, 08-Jul-2014.
[262] N. M. Villegas and H. A. Müller, “Managing Dynamic Context to Optimize Smart Interactions and Services,” in *The Smart Internet*, M. Chignell, J. Cordy, J. Ng, and Y. Yesha, Eds. Springer Berlin Heidelberg, 2010, pp. 289–318.
[263] B. Wang, J. Kurose, P. Shenoy, and D. Towsley, “Multimedia streaming via TCP: An analytic performance study,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 4, no. 2, pp. 16:1–16:22, May 2008.
[264] T. Muto, W. Wu, S. Awiphan, K. Kanai, and J. Katto, “Evaluation of MPEG-DASH over CCNx over different TCPs,” presented at the Packet Video Workshop 2013, San Jose, CA USA, 2013.
[265] “Big Buck Bunny Movie.” [Online]. Available: https://peach.blender.org/. [Accessed: 22-Nov-2014].
[266] Y. Liu, J. Geurts, J.-C. Point, S. Lederer, B. Rainer, C. Muller, C. Timmerer, and H. Hellwagner, “Dynamic adaptive streaming over CCN: A caching and overhead analysis,” in *2013 IEEE International Conference on Communications (ICC)*, 2013, pp. 3629–3633.
[267] S. Lederer, C. Mueller, B. Rainer, C. Timmerer, and H. Hellwagner, “An experimental analysis of Dynamic Adaptive Streaming over HTTP in Content Centric Networks,” in *2013 IEEE International Conference on Multimedia and Expo (ICME)*, 2013, pp. 1–6.

[268] J. Le Feuvre, J.-M. Thiesse, M. Parmentier, M. Raulet, and C. Daguet, "Ultra High Definition HEVC DASH Data Set," in *Proceedings of the 5th ACM Multimedia Systems Conference*, New York, NY, USA, 2014, pp. 7–12.
[269] V. K. Adhikari, S. Jain, Y. Chen, and Z.-L. Zhang, "How Do You 'Tube,'" *SIGMETRICS Perform. Eval. Rev.*, vol. 39, no. 1, pp. 329–330, Jun. 2011.
[270] "Current MITx Courses," *MIT OpenCourseWare | Free Online Course Materials*. [Online]. Available: http://ocw.mit.edu/courses/mitx-related-courseware/. [Accessed: 29-Dec-2014].
[271] R. Mason and G. Cooper, "Introductory programming courses in Australia and New Zealand in 2013: trends and reasons," *Proceedings of the Sixteenth Australasian Computing Education Conference (ACE2014)*, pp. 139–147, Jan. 2014.
[272] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. B. Kopena, "Network simulations with the ns-3 simulator," *SIGCOMM demonstration*, 2008.
[273] "NDLR National Digital Learning Resources," 2009. [Online]. Available: http://www.ndlr.ie/. [Accessed: 30-Oct-2013].
[274] M. Burns and G. P. Schell, "Merlot : A Repository of e-Learning Objects for Higher Education," *e-Service Journal*, vol. 1, no. 2, pp. 53–64, 2002.
[275] edna E. N. Australia, "Education Network Australia," 2009. [Online]. Available: http://apps-new.edna.edu.au/edna_retired/edna/go.html. [Accessed: 30-Oct-2013].
[276] R. Les Cottrell, "Pinging Africa - A decadelong quest aims to pinpoint the internet bottlenecks holding Africa back," *IEEE Spectrum*, vol. 50, no. 2, pp. 54–59, Feb. 2013.
[277] P. S. Kalekar, "Time series Forecasting using Holt-Winters Exponential Smoothing," Kanwal Rekhi School of Information Technology, 2004.
[278] M. Klusch, P. Kapahnke, X. Cao, B. Rainer, C. Timmerer, and S. Mangold, "MyMedia: Mobile Semantic Peer-to-peer Video Search and Live Streaming," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, ICST, Brussels, Belgium, Belgium, 2014, pp. 277–286.
[279] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012–2017," *Cisco*, 06-Feb-2013. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html. [Accessed: 25-Mar-2013].
[280] ITU Recommendation J.247, "Objective perceptual multimedia video quality measurement in the presence of a full reference," 2008.
[281] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electronics Letters*, vol. 44, no. 13, p. 800, 2008.
[282] C. J. Van den Branden Lambrecht and O. Verscheure, "Perceptual quality measure using a spatiotemporal model of the human visual system," 1996, vol. 2668, pp. 450–461.
[283] A. P. Hekstra, J. G. Beerends, D. Ledermann, F. E. de Caluwe, S. Kohler, R. H. Koenen, S. Rihs, M. Ehrsam, and D. Schlauss, "PVQM – A perceptual video quality measure," *Signal Processing: Image Communication*, vol. 17, no. 10, pp. 781–798, Nov. 2002.
[284] ITU-T Recommendation P.861, "Objective quality measurement of telephone-band (300 - 3400 Hz) speech codecs," Aug. 1996.
[285] ISO, "ISO/IEC 14496-12 Information technology -- Coding of audio-visual objects -- Part 12: ISO base media file format," 2012.
[286] ITU/ISO, "ITU-T Rec. H.222.0 | ISO/IEC 13818-1, Information technology – Generic coding of moving pictures and associated audio information: Systems," 2007.
[287] "ITEC MPEG-DASH MPD Validator." [Online]. Available: http://www-itec.uni-klu.ac.at/dash/?page_id=605. [Accessed: 20-Oct-2013].
[288] "MP4Box | GPAC." [Online]. Available: http://gpac.wp.mines-telecom.fr/mp4box/. [Accessed: 18-Mar-2013].
[289] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over HTTP dataset," in *Proceedings of the 3rd Multimedia Systems Conference*, New York, NY, USA, 2012, pp. 89–94.
[290] VideoLAN, "H.264/AVC encoder." [Online]. Available: http://www.videolan.org/developers/x264.html. [Accessed: 17-Mar-2013].
[291] C. Müller and C. Timmerer, "A VLC media player plugin enabling dynamic adaptive streaming over HTTP," in *Proceedings of the 19th ACM international conference on Multimedia*, New York, NY, USA, 2011, pp. 723–726.
[292] "Transform Manager : The Official Microsoft IIS Site." [Online]. Available: http://www.iis.net/downloads/microsoft/transform-manager. [Accessed: 22-Mar-2013].
[293] "Osmo4 | GPAC." [Online]. Available: http://gpac.wp.mines-telecom.fr/player/. [Accessed: 20-Mar-2013].
[294] C. Mueller, S. Lederer, J. Poecher, and C. Timmerer, "Demo paper: Libdash - An open source software library for the MPEG-DASH standard," in *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2013, pp. 1–2.

[295] "Media Source Extensions." [Online]. Available: https://dvcs.w3.org/hg/html-media/raw-file/tip/media-source/media-source.html. [Accessed: 04-Apr-2013].
[296] "IIS Smooth Streaming Server Manifest (On-Demand)." [Online]. Available: http://msdn.microsoft.com/en-us/library/ee230817%28v=vs.90%29.aspx. [Accessed: 22-Jan-2013].
[297] R. Pantos, "HTTP Live Streaming, IETF draft," 15-Oct-2012. [Online]. Available: http://tools.ietf.org/html/draft-pantos-http-live-streaming-09. [Accessed: 11-Apr-2013].
[298] S. Akhshabi, S. Narayanaswamy, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptive video players over HTTP," *Signal Processing: Image Communication*, vol. 27, no. 4, pp. 271–287, Apr. 2012.
[299] R. Kuschnig, I. Kofler, and H. Hellwagner, "Evaluation of HTTP-based request-response streams for internet video streaming," in *Proceedings of the second annual ACM conference on Multimedia systems*, New York, NY, USA, 2011, pp. 245–256.
[300] K. Evensen, T. Kupka, D. Kaspar, P. Halvorsen, and C. Griwodz, "Quality-adaptive scheduling for live streaming over multiple access networks," in *Proceedings of the 20th international workshop on Network and operating systems support for digital audio and video*, New York, NY, USA, 2010, pp. 21–26.
[301] 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.

Appendix A

A.1. Technology Context – Video

A.1.1 Methods for Objective Estimation of Video Quality

This section provides an overview of objective methods for video quality estimation.

Objective video quality assessment methods are methods that use automated computational signal processing techniques to predict subjective quality assessment of human viewers. There are several objective methods which may be employed to measure the quality level and detect impairments such as blocking, blurring, contrast and colour errors as well as jerkiness, frame skips and freezes in the video playout sequence. These quality metrics usually compare the original (distortion-free) image and the distorted image. They can be classified according to the availability of an original image to full-reference (complete reference image known), reduced-reference and no-reference or "blind" quality assessment approach (reference image not available). While the latter approach does not require access to the original image, such computational methods are both resource and time intensive. The ITU has adopted a three stage approach to recommending objective perceptual assessment methods for multimedia. The first two stages identify perceptual quality tools appropriate for measuring video and audio individually, while the final, third stage identifies objective assessment methods for composite audiovisual media. ITU R.J.247 [280] focuses on the first stage and defines a number of appropriate objective perceptual video quality measurement methods, given the availability of a full reference signal, for both Internet multimedia streaming and for mobile video streaming over telecommunications networks.

Peak Signal to Noise Ratio (PSNR) is an example of an objective, pixel based, QoE metric based on a simple mathematical model. It is used to predict the quality level of multimedia services according to the estimated user’s perception. This full-reference metric compares processed and original video using Mean Square Error (MSE), and due to its conceptual and computational simplicity [281], is one of the most popular metrics and is still widely used in video networking studies. It should be recognised that any pixel error, visually perceivable or not, decreases PSNR. A PSNR to MOS mapping with the equivalent ITU-T R. P.910 quality and impairment scale [93] is given in Table 2-1 (page 23).

While PSNR is very simple and easy to use, it does not consider a very important factor – the **Human Vision System (HVS).** HVS approaches are alternatives to pixel based methods and include:

- Psychophysical approach: based on models of HVS which abstract estimated sensitivity to contrast and orientation, frequency selectivity, colour perception, etc. The HVS approach is generic and may be used in a wide variety of video applications, however, HVS models tend to be complex and computationally demanding. These are typically full-reference models.

- Engineering Approach: based on image analysis and the extraction of video characteristics and errors, not excluding aspects of HVS. Most of reduced- and no-reference metrics fall into this category.

An example of the HVS approach is the **Moving Picture Quality Metric (MPQM)** [282], which is an objective quality metric that considers contrast sensitivity and masking. Human eye sensitivity depends on the spatial/temporal frequencies present in an image where a signal is perceived if the signal contrast is higher than a threshold value. The human response to combined signals exhibits so-called masking phenomena, where for example, the foreground sensitivity might be impacted by the contrast of the background. MPQM-based assessment begins with the decomposition of the original sequence and distorted sequence into perceptual channels and contrasting sensitivity and masking are accounted for using a channel-based distortion measure. Finally, a quality rating ranging from 1 (bad) to 5 (excellent) is calculated based on mathematical data analysis.

Perceptual Video Quality Measure (PVQM) [283] focuses on the most dominant cognitive effects (e.g. the human eye is more sensitive to sharp transitions in the luminance component than to changes in chrominance components in quality measurements). The same approach was adopted for a speech quality measurement system Perceptual Speech Quality Measure, PSQM [284]. PVQM uses a linear combination of three indicators: the “edginess” of the luminance, the normalised colour error and the temporal decorrelation. The method achieves a full reference metric and thus takes two video sequences as input (reference and delivered).

The **Structural SIMilarity (SSIM)** [89] index measures the similarity between two images. The SSIM index indicates a quality measure of one of the images being compared (test video), provided the other image is regarded as of perfect quality (original video). It is designed to improve on PSNR/MSE as it measures the change in structural information and "the HVS is highly adapted for extracting structural information" [89, p. 600].

This is a well researched area and comprehensive surveys may be found in [89], [90].

A.2. MPEG-DASH

This section introduces segment types in MPEG-DASH and presents an outline of popular MPEG-DASH related tools and datasets. It also provides a comparison of MPEG-DASH enabled players.

A.2.1 MPEG-DASH Segment Types

A segment is a fundamental element of the DASH standard. It is a unit of data associated with an HTTP URL and requested by DASH clients. Optionally, a segment can be associated with a byte range which may be requested individually. The MPEG-DASH standard introduces four types of segments, namely **Media Segments**, **Intialization Segments**, **Index Segments** and **Bitstream Switching Segments**. This section provides an outline of each.

Media Segments contain and encapsulate media streams complying with the media format in use and enable playback when combined with zero or more preceding segments, and an Initialization Segment (if any). These segments are independent of previous/successive segments in terms of decoding, as a segment contains a portion of the stream that begins at video GOP (see Section 2.1 introduction) boundaries starting with an I-frame. The segments contain accurate Media Presentation timing information enabling synchronisation of components and seamless switching. They may be further subdivided into ***Subsegments,*** each of which contains a whole number of complete ***Access Units*** (AU). A Subsegment is a unit within Media Segments that is described by a Segment Index, whilst an AU is a unit of a media stream with an assigned Media Presentation time. DASH fully supports two segment types: ISO/IEC 14496-12 ISO Base Media File Format (ISO BMFF) [285] (currently used by Smooth Streaming and HDS) and ISO/IEC 13818-1 MPEG-2 Transport Stream (MPEG-2 TS) [286] (currently used by HLS). This lends itself to an easy use of existing adaptive streaming content by MPEG-DASH, where the index files need to be migrated to an MPD format, while the media segments can frequently be easily reused.

Initialization Segment contains metadata describing the encoding of the media content necessary to initialise the Media Engine and enable playout. The Initialisation Segment is media format specific. Each Representation either contains an Initialization Segment or each Media Segment in the Representation is self-initialising.

Each Media Segment is indexed; it either contains a Segment index within the Media Segment (typically at the beginning of the file) or utilises separate ***Index Segment*** providing indexing information for the Media Segment. A Segment Index provides timing and stream access information for the Representation and corresponding byte range in the Segment occupied by each Subsegment for one or more media streams. Timing information includes: presentation time range; the earliest presentation time of access units in each Subsegment of an indexed media stream; and the presentation time of the first Stream Access Point (SAP), if present.

Each Representation is assigned at most one ***Bitstream Switching Segment*** (that contains essential data to perform a switch to the Representation to which it is assigned) which is relevant when Segments from different Representations are sequenced.

A.2.2 MPEG-DASH Related Tools and Data Sets

There are a variety of freely available tools that are DASH-enabled, and this section identifies a number of important examples. These tools range from MPD validators (which check if the structure and content of a given MPD file conforms to the standard), to DASH content generators.

ITEC MPEG-DASH MPD validator [287] supports ISO/IEC 23009-1:2012, ISO/IEC 23009-1:2012/COR.1 (w13282), and ISO/IEC 23009-1:2012/AMD.1 (w.13284).

GPAC MP4Box. [288] is an MP4 multiplexer used for video conversion, splitting as well as video hinting and dumping. This multi-purpose command line tool can also be used to import different (e.g. H.264 AVC) video and audio streams into the .MP4 container to produce compliant MP4 (MPEG-4 System) streams. This tool is a part of the GPAC Project framework and generates both segment (fragmented MP4) files and corresponding MPD files. However, the generated MPD files must be manually merged to a resulting MPD file describing multiple representations.

DASHEncoder [289] is an open source tool that generates representations, fragmented MP4 files, and an MPD file according to an input configuration file or command line parameters. This tool uses x264 [290] for the video encoding (H.264 AVC format) and GPAC MP4Box [288] for the multiplexing and the MPD generation (on representation level) to build a combined MPD file describing all representations. The resulting MPD file does not require manual editing and the content generated is compatible with the DASH VLC plugin [291] (handles decoding and playout).

IIS Transform Manager [292] is an extensible media transform engine that enables queuing, management, integrated media transcoding/transmuxing, and batch-encryption of on-demand audio and video files. It handles for example, conversion from Windows Media-formatted and MP4-formatted files to on-demand Smooth Streams for delivery to Smooth Streaming-compatible clients (e.g. Silverlight). The generated segments with an appropriate MPD file are suitable for DASH-enabled streaming.

DASH-formatted Video Content. There are few freely available DASH datasets. Many researchers consequently resort to using short, freely available, video sequences which are concatenated multiple times to achieve longer test sequences. This process results in a video sequence with a limited variety of scenes (fade in, fade out, low and high motion, etc.), which do not correspond accurately with real world settings [289]. The DASH Dataset [289] is the first freely available DASH Dataset that provides various full-length videos in a variety of genres, resolutions, bitrates and segment lengths. The current implementation - D-DASH [124] is mirrored across five different European locations to enable CDN-like scientific evaluations. The dataset MPDs are managed in a repository at Alpen-Adria University and are periodically replicated at the different mirrors by the remote site owners. An Ultra High Definition DASH-formatted dataset[21] [268] of HEVC video content, including multiple encoding bitrates and packaging options is hosted by the Signal and Image Processing Department of Telecom ParisTech. This data set provides streams with bitrates appropriate for UHDTV display resolution (e.g. 3840x2160).

A.2.3 HTTP-based Adaptive Streaming Players

This section identifies a selection of open-source clients which support DASH video playback.

GPAC Osmo4[22] [293] is a highly configurable multimedia player that supports many existing delivery protocols including DASH. It is also capable of playing back audiovisual content mixed with 2D or 3D content. This multiplatform player is integrated with the majority of Web browsers and supports both MPEG-DASH and Apple HLS playback. Osmo4/MP4Client plays back from a HTTP(s) server or from local storage (for testing purposes). It supports much of the MPD syntax; different media segment types, multiple periods, group selection, independent (audio and video) component download.

VLC Media player with DASH plugin[23] [291], is a DASH plugin for VLC[24] (open source media player). This implementation is built with libdash [294] (a DASH client library).

DASH-JS on HTML5[25] [118] is a JavasScript based DASH library for Google Chrome. This is an integration of the DASH standard into the Web environment using the HTML5 video element. The Google Chrome Media Source Application Programming Interface API [295] provides access to the HTML5 video element directly, allowing the use of its decoder unit.

Popular commercial media players supporting adaptive streaming over HTTP are outlined below. These players support both on-demand and live adaptive bitrate streaming and the differences between these proprietary solutions are indicated in Table A.2-1 (page 6).

Microsoft Silverlight Smooth Streaming player [105]**.** This player is an IIS Media Services extension which optimises content playback by switching video quality in real-time. An IIS Smooth Streaming Server manifest file specifies media files that comprise the presentation, heuristic parameters, such as bitrate and fragment duration (e.g. 2 seconds) and quality index for each track (adaptation set) [296]. This proprietary video player application can be downloaded on-demand by the Web browser. The player generates HTTP requests for audio and video fragments (based on the manifest file) which contain the content name, requested bitrate, and fragment start identification (a timestamp based on the per-fragment information in the

[21] Available from http://download.tsi.telecom-paristech.fr/gpac/dataset/dash/uhd/ [Accessed: 4-Jan-2016]
[22] Available from http://gpac.wp.mines-telecom.fr/player/ [Accessed: 4-Jan-2016]
[23] Available from http://www-itec.uni-klu.ac.at/dash/?page_id=10 [Accessed: 4-Jan-2016]
[24] Available from http://www.videolan.org/vlc/index.html [Accessed: 4-Jan-2016]
[25] Available from http://www-itec.uni-klu.ac.at/dash/?page_id=746 [Accessed: 4-Jan-2016]

manifest). Microsoft Smooth Streaming Client version 2.5 supports DASH for Silverlight on-demand scenarios.

Apple HTTP Live Streaming (HLS) [297] is integrated in the Safari Web browser on the Mac platform (but not yet supported on Windows and Linux), where the m3u8 manifest file is defined as the source of the HTTP5 video element, allowing manifest parsing and segment download to be performed within the Web browser.

Adobe Flash (Adobe HTTP Dynamic Streaming [3]) is an example of a proprietary video player that is downloaded on-demand by the Web browser but which has only limited support on mobile platforms.

Feature/ Solution	**Apple HLS** [100]	**Adobe HDS** [99]	**MS IIS** [101]	**MPEG-DASH Standard**
Server	Web Server	Adobe Media Server	MS IIS Server	Web Server
Client	HTTP5 Video Element, Apple QuickTime player	Flash Player	Silverlight	Variety
File on Server	Fragmented	Contiguous	Contiguous	Fragmented
Video Manifest	m3u8	f4m file	Smooth Streaming Manifest file	MPD
Default segment	10 seconds	2 seconds	4 seconds	Variable
Media Container	MPEG-2 TS	MP4-part 12, FLV	MP4-part 12 (fragmented MP4)	MPEG-2 TS, ISO BMFF
Video Standard	H.264 Baseline Level	H.264	Agnostic	Agnostic

Table A.2-1: Cross-comparison of HTTP Streaming Solutions

There have been a number of case studies in which HTTP media players were evaluated. One such study [298] details an experimental evaluation of two commercial players and one open source player. The authors focused on: player reaction to persistent or short-term throughput changes; the ability to perform on a shared network path; and the performance with live streamed content. Significant inefficiencies (e.g. oscillations, unnecessary bitrate reductions, etc.) were identified with regard to each of the players under investigation. A further study [174] experimentally investigated HD video distribution performance employing HTTP-based adaptive streaming using the Akamai CDN. Results showed that short interruptions of the video playback can occur due to a sudden drop in the available bandwidth as the client contacts the server on average every 2 seconds. Alternatively, approximately 150 seconds were required to request higher quality subsequent to a sudden increase in bandwidth.

A.2.4 Other Issues in HTTP Streaming

This section presents a selection of issues considered in HTTP-based streaming algorithms, focusing on multiple TCP connections and request timing.

A.2.4.1 Multiple TCP Connections

The rapid deterioration in performance of a single TCP connection with increasing packet loss is noted in [299] where experimental evidence identified multiple HTTP-based request-response streams (each implemented by a separate TCP session) to be a good alternative to classical TCP streaming as they maintain satisfactory performance despite increasing packet loss conditions. Another example of the merits of multiple TCP sessions over a single one is presented in [120]. Here a segment is retrieved in parallel via a number of independent paths (the bitrate of the next segment requested is determined on the basis of the aggregate of the estimates for individual paths). A multilink extension of an adaptive, segmented video streaming system implementing core MPEG-DASH functionality, is proposed in [123], [300]. The approach taken divides video segments into subsegments, which are then requested over multiple paths and interfaces simultaneously. However, the evaluation presented in [112] indicates that a single connection is better than two in the case of bottlenecks. While one connection was used for video, and the other for audio, they shared the same endpoints and bottleneck and were consequently competing for the same bandwidth.

A.2.4.2 Request Timing

A number of studies investigate the scheduling of content requests. For example, the study conducted in [299] examines inter-request gap times (the artificial gap between the requests) and adjusts them to achieve TCP fairness. It is concluded that smaller inter-request gaps lead to higher throughput. However, gap times have a greater influence on small segments and increase transmission latency. An evaluation of client-side request strategies for live adaptive HTTP segment streaming [301] shows that the strategy of segment requests can have a considerable impact on bandwidth utilisation and attained video quality. The synchronisation of client requests leads to competition for bandwidth, and has a negative impact on router queues. Since this results in increased packet loss and severe underutilisation of bandwidth it is recommended that synchronisation should be avoided to achieve a high goodput.