Cognitive algorithms and the optimisation of Quality of Experience and bandwidth utilisation

Trinity College

Cognitive algorithms and the optimisation of Quality of Experience and bandwidth utilisation

Mark Stokell

15th August 2015

Author

Mark Stokell

A thesis submitted to the University of Dublin in fulfilment of the requirements for the degree of MSc (Hons) Computer Science.

Date of Submission: 15th August 2015

Permission to lend and/or copy

I agree that the School of Computer Science and Statistics, Trinity College may lend or copy this dissertation upon request.

Signed: ______________________________________

Mark Stokell

15th August, 2015

Acknowledgements

I wish to thank my project supervisor, Professor Declan O'Sullivan of the Knowledge and Data Engineering Group of the School of Computer Science and Statistics, for his invaluable advice and insights, which had an enormous impact on the direction and outcome of my research.

I am indebted to Dr Ray Richardson for both for the practical advice provided in his capacity as an experienced researcher in Computer Science, and for his support in a professional capacity. This also had a significant impact on the approach taken.

I am grateful to my employers, Daysha Consulting Ltd., for their support of my study during the previous three years and, not least, for funding the activity

My family have shown tremendous understanding and supported me consistently throughout this research study and the previous B. Sc. Programme; for this I will always be grateful

Signed: ____________________________________

Mark Stokell

15th August 2015

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Abstract

This thesis presents an investigation of the effect of machine learning algorithms on the Quality of Experience (QoE) for users when downloading or streaming video to network connected devices such as PCs, Mobile phones, game consoles and smart TVs while optimizing network load

Building on Scalable Video Coding (SVC) as the mechanism to maximize intrinsic video quality within the constraints of network bandwidth and device capability, cognitive algorithms have been evaluated to determine their ability to maximize QoE whilst maintaining a minimum network load.

The environments evaluated are based on peer-to-peer or client server networks where the final user is on a home network with wired and wireless devices competing for bandwidth. Consequently, there is congestion in both the backbone (internet) and access (home) networks and uncertainty as users and peers connect and disconnect from the network.

Methods of objectively estimating QoE in real time are investigated and key influencing parameters are identified. These parameters are used to evaluate cognitive algorithms in a simulation environment.

Contents

Tables and Figures

1 Introduction

1.1 Problem Statement

From a technology perspective the modern family home constitutes a heterogeneous network of devices accessing many and varied content from the World Wide Web. The home network is invariably connected to the internet via a router provided by an internet service provider (ISP) and/or a satellite TV provider. The content accessed typically includes simple text files, audio files and video files.

At the time of writing the availability of video content, its quality, and the demand for it, is increasing rapidly (Zambelli, 2013). Whilst broadband connection speeds are increasing – currently, up to 200Mbs bandwidth is available via optical fibre in the Dublin area (UPC, 2015) – the quality and therefore file size of the video content is also increasing: HD video file sizes range from 1 to 5 GBytes. Consequently, the competition for home network bandwidth is high despite improving broadband speeds. In a family of 3 teenage children and 2 working parents it is not unusual to find all family members individually and concurrently downloading or streaming large files to different devices.

Such a family usually accesses content via Wi-Fi which effectively limits the maximum available bandwidth to 100Mbs (IEEE Standards Authorty, 2012). In reality, this is usually reduced to less than 50 MBs due to competition for the radio frequency with other devices, such as remote controls and wireless surround sound, and the physical architecture of the home in question. Due to the fact that 8011 protocol demands that all users remain connected and device bandwidth is stepped down to satisfy this requirement (IEEE Standards Authorty, 2012), each user will experience an actual bandwidth much lower than 50Mbs. As a result the quality of video steaming and the rate of download is often less than satisfactory.

When considering the different types of content (movie, sport, news) and the different devices (iPad, HDTV, laptop and so on) drops in performance have a different effect on the quality of experience (QoE) for different users. For example, a user experiencing buffering whilst watching a HD movie on a 55inch wide screen TV will be far more dissatisfied than a user experiencing pixilation watching a news item on their iPhone. Thus, the question arises; how is it possible to seamlessly optimise the QoE for all users in a network where devices are connecting and accessing content in an ad hoc manner?

1.2 Research Question

The research question investigated in this thesis is to what extent can cognitive algorithms[1] support the optimisation of Quality of Experience (QoE)[2] and bandwidth utilisation for users when downloading or streaming video to connected devices (such as PCs, Mobile phones, game consoles and smart TVs), in a resource contended network[3]? Furthermore, how effectively can such algorithms cope with the uncertainty[4] of the quality on the device generated due to corruption[5] or interruption[6]?

Exploring these questions the research seeks to address some of the following:

- Can QoE be used to optimise content delivery in near real-time[7]?
- Can QoE be estimated from simple indicative parameters?
- Can devices be self-aware[8] with respect to the content they receive and the QoE they deliver?
- Can networks be optimised with respect to QoE and network load in real time?
- Can this be achieved using distributed cognitive algorithms with minimal reliance on central policy?

Building on Scalable Video Coding (SVC) (Schwarz, et al., 2007) as the mechanism to maximize intrinsic video quality within the constraints of network bandwidth and device capability, cognitive algorithms will be evaluated to determine their ability to maximize QoE whilst maintaining a minimum network load. It is intended to investigate the possibility to deploy such algorithms at the device level such that the devices be self-aware (in terms of QoE capability and bandwidth consumption) and content aware (in terms of video category such as sport, news, movie and so on). In addition, it is necessary to evaluate the ability to

[1] Cognitive Algorithms are processes that model some degree of human reasoning based on storing and using information to affect a change (Russell & Norvig, 2010)

[2] Quality of Experience is human subjective and objective quality needs and experiences arising from the interaction of a person with technology (Laghari & Connelly, 2012)

[3] Resource contended network is a collection of connected devices that share common resources such as bandwidth. When demand for the resource exceeds supply the network is said to be contended

[4] Uncertainty of quality refers to the unpredictable nature of the network in question where devices may join or leave and data streams may vary in quality, speed and availability

[5] Video data may be damaged in a such a way as to make it unreadable

[6] Video streams may be interrupted due to device failure or network communication failure

[7] The delay introduced, by automated data processing, between the occurrence of an event and the use of the processed data, e.g., for display or feedback and control purposes. (Telecommunications Industry Association, 2015)

[8] Self-Aware in the context of a connected device means that an Artificial Intelligence agent has access to data regarding the device, content and the environment

cope with the uncertainty of devices being added to or removed from the network, and variable prevailing bandwidth.

Resource contended network scenarios include content delivery via peer to peer or client server networks where the final users are in a home network with wired and wireless devices competing for bandwidth. Consequently, there will be congestion in the access (home) network and uncertainty as users and peers connect and disconnect form the network.

Methods of objectively estimating QoE in real time will be established and key influencing parameters will be identified. These parameters will be used to evaluate cognitive algorithms in a simulation or test bed environment. It is intended that the actual QoE is perceived and acted upon at device level. This type of solution is considered preferable to a central policy derived solution based on intelligent routing for example (Strassner, 2003).

1.3 Impact Summary

The results of the research will provide mechanisms that home users could use to optimise video streaming quality at minimum network load. In addition, the methodologies developed could be extended to encompass all content (flat file, video and audio) competing for bandwidth in a multi device/user environment. Furthermore, it is envisaged that the solutions could be applicable in commercial contended networks such as corporate LANs.

Researchers in network management will benefit from the results. Mapping of objective measures to subjective estimates of QoE will be of particular benefit to researchers in video quality domains.

In order to maximise the impact, the research will be published as widely as possible, beginning with the International Workshop on Design of Reliable Communication Networks[9]. In addition, industrial contacts will be exploited with a view to developing commercial prototypes.

1.4 Thesis Structure

The study comprised two broad streams of activity: a literature survey and a technical study. The study was undertaken part-time, as the author of the thesis is employed full-time. The initial exploratory research phase was completed during the first 6 months of the study, the

[9] International Workshop on Design of Reliable Communication Networks – March 14th-March 17th 2016, Paris

outcome of which was a target research area. The subsequent 12 months was dedicated to literature review and continuing refinement of the research question. The final period of study was dedicated to developing the simulation experiments. This began with the proof of concept based on an Excel model of a home network and concluded with a computer simulation model that was used to analyse the effect of the chosen algorithms on QoE.

This thesis comprises two main elements, the Literature Review and the Technical Approach. The former describes the state of the current research literature with respect to the research objective and the latter describes the experimentation performed to address the research questions and relevant findings. The thesis is completed with conclusions and recommendations for further work. The document structure:

- Section 2 – Literature Review – provides a discussion of the relevant studies
    - Approach
    - video streaming,
    - Quality of Experience,
    - Artificial Intelligence and network management
- Section 3 - Methodology – describes the experimental approach and simulation architecture
- Section 4 – Implementation - describes the experiments and findings from network simulation
- Section 5 – Conclusions – provides a discussion of the findings of both the literature review and technical study and suggests areas of further investigation that could build on this study

2 Literature Review

This section constitutes the review of literature relevant to research in the three main domains pertinent to the research question; Quality of Experience (QoE) for users obtaining and viewing video content via the internet, video streaming technology, and the use of artificial intelligence in the management of content delivery over networks. The body of knowledge relevant to this study was analysed and the findings are given in this section. The review was performed in the context of technology, theory and application to video streaming and comprised the study of seminal books, textbooks, and academic papers. Trinity College library resources, physical and on-line, were utilised to discover and analyse works by key contributors in both the academic and commercial spheres. The most important online databases consulted are described below (EBSCO Industries, 2014)

- Academic Search Complete
  - 8,500 full-text periodicals
  - 7,300 peer-reviewed journals with full text articles
  - Abstracts for a further 12,500
  - PDF content going back as far as 1887
- Business Source Complete
  - abstracts and full text back as far as 1886
  - more than 1,300 journals
- Business Abstracts with Full Text
  - full text of articles from 510 key publications
  - PDF content going back as far as 1995
- Social Sciences Full Text
  - 625 periodicals
  - PDF content going back as far as 1983

2.1 Approach to Literature Review

The review is an attempt to establish and report on the key concepts in the field of artificial intelligence and its application in the delivery of QoE for consumers of video streaming over networks.

The field of artificial intelligence is large and has been the subject of research since the 1950s. So a general review of the broad concepts has been undertaken to establish a context for the review. This was followed by a deeper exploration of the application of machine learning techniques in the area of telecommunications and data network management.

Work in the area of QoE for video is relatively immature. QoE is by definition subjective but for real-time adaptation of video service over networks some objective methods of estimating QoE must be applied. As a result, much focus has been given to the state of the art with respect to objective models and mapping to subjective evaluation of QoE.

Delivery of video over the internet has grown significantly over recent years (Zambelli, 2013) and much work has been undertaken in development of methods to efficiently deliver content. This work has been reviewed through the lens of user perceived quality and consumption of network resources with particular emphasis on Scalable Video Coding (SVC) and Dynamic Adaptive Steaming over HTTP (DASH).

The structure of the literature review was formed around seminal books and academic articles (Sections 2.1.1 and 2.1.2). The results of the literature review are presented as a semi-structured progression: beginning with a review of the core concepts of video delivery (Section 2.2); progressing to a discussion of issues related to Quality of Experience (Section 2.3), and closing with a review of artificial intelligence and network management (Section 2.4). However, given the complexity of interdependencies and cross-linkages between these themes it was not possible to examine each in isolation.

2.1.1 Core Texts

Artificial Intelligence: A Modern Approach (Norvig & Russel, 2003)

2.1.2 Other Sources

1. IEEE Communications Magazine
2. IEEE Transactions on Wireless Communications
3. Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)
4. ACM SIGCOMM Conference
5. IEEE Transactions on Vehicular Technology
6. IEEE Journal on Selected Areas in Communications
7. IEEE Communications Surveys and Tutorials
8. IEEE Transactions on Communications
9. IEEE/ACM Transactions on Networking (TON)

10. IEEE Transactions on Mobile Computing
11. Computer Networks
12. Internet Measurement Conference
13. Annual International Conference on Mobile computing and networking
14. arXiv Networking and Internet Architecture (cs.NI)
15. IEEE Wireless Communications
16. IEEE GLOBECOM Workshops
17. IEEE International Conference on Communications
18. Computer Communications
19. Ad Hoc Networks

2.2 Video Delivery over the internet

Video delivery over the Internet has been of serious interest since the proliferation of computers in the home, and growth of online video: CISCO predict one million minutes of video content will be transmitted over the internet every second by 2018 (The Economist, 2014). It was not until the 2000s that the capabilities of the hardware, with respect to CPU[10] power and network bandwidth, were sufficient to support downloading in reasonable time for subsequent play back. Later still, when the bandwidth of the last mile finally reached multiple megabits per second, real time streaming was possible.

2.2.1 Mechanisms and Protocols

A video stream is a composite of audio and video streams. The original source is compressed using a compression mechanism called a *codec* (Gantenbein & Robinson, 2008). The audio stream is compressed using an audio codec such as MP3 and the video stream is compressed using a video codec such as H.264. The encoded audio and video streams are assembled in a container bit stream such as MP4. The bit stream is delivered from a streaming server to a streaming client using a transport protocol, such as MMS[11] or RTP[12]. Newer technologies such as HLS, Microsoft's Smooth Streaming, Adobe's HDS and finally MPEG-DASH have emerged to enable adaptive bitrate streaming over HTTP as an alternative to using proprietary transport protocols.

[10] Central Processing Unit
[11] Multimedia Message Service
[12] Real Time Transport Protocol

Video compression for subsequent transmission began in the mid-1980s with the definition of the ITU (International Telecommunications Union) H.120 digital coding standard (Jacobs & Probell, 2007). Practical video compression really began in 1990 with the ITU H.261 standard targeted for transmission at 64kb/s over ISDN with CIF[13] resolution. Shortly after this, Motion JPEG (MPEG) began to be used for video streams because it offered editing efficiency. This led to the MPEG-1 standard of 1992 aiming to achieve acceptable quality at 1.5Mb/s and CIF resolution. In the following year MPEG-2/H.262 standard was developed jointly by ISO and ITU. This became the standard for "Standard Definition" (SD) video, 720x576 pixels at 3 to 10 Mb/s. In 1993 the practical limit of the access network was far less than 3 Mb/s with ISDN 128kb/s being the highest available bandwidth, this was often referred to as the "last mile problem" (Tucker & Westereveld, 2015).

Development of the now ubiquitous MPEG-4 codec began in 1995. Over the next decade development continued to improve resolution and quality, and take advantage of the improving bandwidth in the access network. This culminated in the development of the Advanced Video Coding (AVC) standard; H.264/MPEG4/AVC (Wiegand, et al., 2003). H.264/AVC is the basis for high definition (HD) video delivery we enjoy over the broadband access network today.

The contribution to video streaming of developments in video coding cannot be considered in isolation. The transport mechanism is equally important. In the early days of video streaming many proprietary mechanisms were prevalent, each using different approaches to deliver content to the users.

In the mid-2000s the vast majority of the Internet traffic was HTTP-based and content delivery networks (CDNs) were increasingly being used to ensure delivery of popular content to large audiences. (Zambelli, 2013). In the early days, streaming media was mostly based on UDP[14]. However, in 2007 HTTP-based adaptive streaming was introduced by a company named Move Networks. They used the dominant HTTP protocol to deliver media in small file chunks that could be sequenced for playback at the device level. In addition, the device player application could be used to monitor download speeds and request chunks of varying quality (size) in response to changing network conditions and caching could be

[13] 352x288-pixel resolution
[14] User Datagram Protocol

employed to improve efficiency. The technology had a huge impact because it allowed streaming media to be distributed using Content Delivery Networks (CDNs).

This first method of delivery became known as progressive video. (Sandvine, 2015). In this case the large file is pushed onto the network regardless of available bandwidth. The user determines the quality of video desired, usually in coarsely defined resolution options; HD, SD and so on. In effect, users watching at different quality levels are accessing different files from different sources (servers or nodes). Progressive video streams contain the video and audio codec files and some information that can be used for QoE assessment:

- **IP information**: Subscriber, CDN
- **Subscriber information**: physical location on network, service plan, device type
- **TCP information**: transport layer information - not of relevance to QoE
- **HTTP information**: asset (used to link multiple chunks together), duration, stall information (transport quality)
- **Container information**: codec, resolution, bit rate (display quality), CDN
- **Elementary stream**: bytes transferred

HTTP-based adaptive streaming solutions (Stockhammer, 2011) followed: adaptive video protocols effectively modulate the display quality based on the network's available transport capacity (i.e., bandwidth). It achieves this by breaking the video file into "chunks" and transmitting the chunks of the video in a piecemeal fashion. Figure 2-1 represents the adaptive streaming process. At each stage of playback the chunk is chosen such that the playback is at the maximum deliverable display quality for the prevailing network conditions at that time. The process begins with the client device's video playback application requesting the first chunk. Once the chunk is decoded playing commences. However, if the first chunk takes too long to deliver, then the next chunk will be requested at lower display quality; conversely, if this initial chunk delivered especially quickly, exceeding some delivery rate parameter, then the next chunk will be requested at a higher display quality. In this way the system iterates to the maximum video quality. Of course, changes in available bandwidth at the client can occur dynamically and QoE may be affected by perceived changes in quality due to adaptation and, in severe, cases buffer run out.

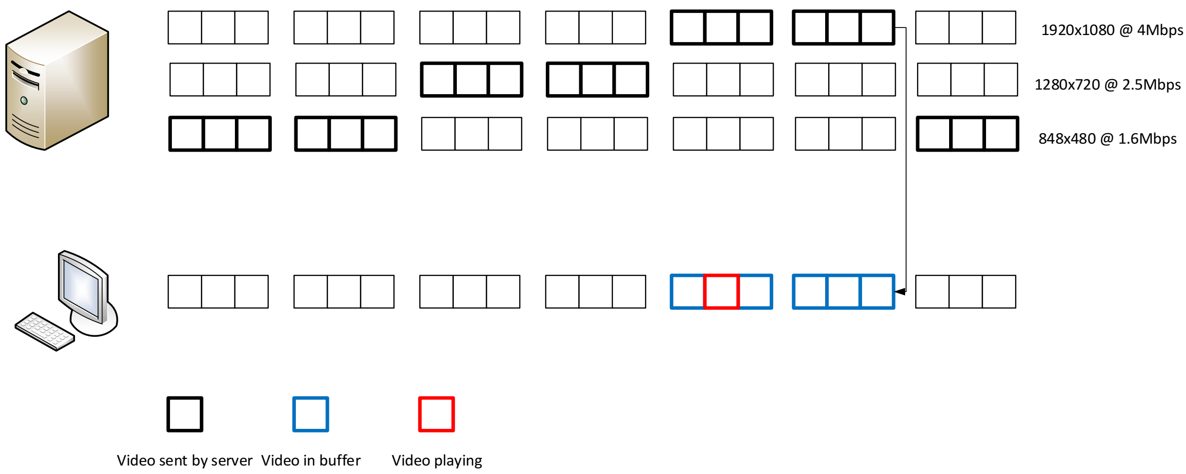

FIGURE 2-1 - ADAPTIVE VIDEO STREAMING

Once again the adaptive stream contains information about the quality of the video and its delivery:

- **IP information**: Subscriber information, CDN
- **Subscriber information**: physical location on network, service plan, device type
- **TCP information**: transport layer information - not of relevance to QoE
- **HTTP information**: asset (used to link multiple chunks together), 'protocol', CDN
- **Protocol information**: duration, stall information (transport quality)
- **Container information**: codec, resolution, bitrate (display quality)
- **Elementary stream**: bytes transferred

Adaptive streaming began around 2008 when Microsoft launched its Smooth Streaming technology (Zambelli, 2013). In the same year Netflix developed its own technology to power its *Watch Instantly* streaming service. Apple soon followed in 2009 with HTTP Live Streaming (HLS) designed for delivery to iOS devices - Mac, iPad, iPhone. Adobe released its version of HTTP Dynamic Streaming (HDS) in 2010. HTTP-based adaptive streaming soon became the medium of choice for high-profile live streaming events (London Olympics, Wimbledon and similar). HTTP adaptive streaming now forms the transmission model for all premium on-demand services (Netflix, LoveFilm, Amazon Instant Video and so on).

Over this relatively short period it became evident that a large number of different proprietary protocols for HTTP steaming was a threat to the stability of an emerging entertainment industry. Consequently, efforts to establish industry wide standards for streaming protocols began. In 2009 the 3rd Generation Partnership Project (3GPP) began work on a new industry standard for adaptive streaming (3rd Generation Partnership Project, 2015). Early 3GPP standardisation work was absorbed into the International Standards Organisation (ISO) MPEG working groups in 2010. Work on the standards was completed relatively quickly. Proposals based on the 3GPP work were moved through draft to ratified status by 2012. The new standard was named Dynamic Adaptive Streaming over HTTP, more commonly referred to as MPEG-DASH.

Zambelli (2013) asserts that the original specification for MPEG-DASH suffered from excessive ambiguity and, as a result, the majority of the companies involved in MPEG-DASH formed a *DASH Industry Forum* with the goal of promoting DASH adoption and establishing a well-defined set of interoperability constraints. In 2013 the DASH-IF published a draft (version 0.9) of its *DASH264 Implementation Guidelines* (DSH-IF, 2015). The DASH264 guidelines provide important interoperability requirements such as support for the H.264 video codec (Sodagar & Giladi, 2015). DASH264 defines other essential interoperability requirements such as support for HE-AAC v2 audio codec, ISO base media file format, SMPTE-TT subtitle format, and MPEG Common Encryption for content protection (DRM[15]). The forum continues to publish guidelines as new complementary developments arise. These guidelines are open for community review. DASH264 provides the details needed for adoption of MPEG-DASH.

Significantly, Zambelli (2013) elaborates the quality gap that is the major hurdle facing streaming media industry. In a short space of time streaming media technology has progressed from less than SD video to 720p HD video, but the quality of even the best video-on-demand still falls short of broadcast television and Blu-ray quality. While most HD video delivered over satellite is 1080i video H.264-compressed at 17-37 Mbps (DVB Standatds Organisation, 2014), most HD streamed video is only 720p, encoded at a modest 3-4 Mbps. Broadcast television is always delivered at 50Hz in Europe, whereas streaming video is nearly always delivered at half the frame rate – 25Hz. Finally, broadcast audio is typically mixed and delivered in 5.1 surround (Dolby, 2014) whereas streaming audio is still largely simple stereo. Zambelli considers this quality gap to be a barrier that streaming media

[15] Digital rights management

companies must overcome to compete with traditional media delivery. However, the staggering growth of Netflix subscribers would suggest that this barrier has been breached. According to a Forbes report (Forbes, 2015), Netflix had gathered 57.4 million subscribers by the end of 2014.

Video streaming quality is primarily dependent on bandwidth and the method of increasing quality is by increasing the bandwidth and/or by improving the compression efficiency. Increasing available bandwidth is dependent on the "last mile" (access network) and subscriber (in the home) infrastructure. Currently, improved codec technologies are delivering higher quality video over the existing infrastructure (Gitman, et al., 2015) (Ceglie, et al., 2014). The latest ISO/IEC collaboration has resulted in the H.265 codec. This allows delivery of 1080p (Full HD) video at the same 3-4 Mbps currently used for 720p video delivery, or increased frame rate to 50/60Hz without requiring a proportional increase in bandwidth.

Video delivery mechanisms have grown from the traditional client server architecture, where the source of the bit stream is the provider's server and the consumer's device is the client. In the late 1990s an alternative delivery mechanism was created. MP3 file sharing was promoted by Napster and Gnutella for music file distribution (Carlsson & Gustavsson, 2001). This became known as peer-to-peer (P2P) file sharing and the network is referred to as the P2P network, where each peer or node, is the client <u>and</u> the server. With improvements in network bandwidth P2P file sharing became used for video content as well as audio content.

According to Steinmetz and Wehrle (2005), a P2P system is "*a self-organising system of equal, autonomous entities which aims for the shared usage of distributed resources in a networked environment avoiding central services*". Clearly, some central resource must be the source of the original file that is shared among the peers. Of course, when a small number of peers hold copies of the source file then P2P sharing is possible.

P2P networks rely on a peer's ability to identify other peers, determine the files that other peers have available for sharing, and decide from which peers to download content. Distributed Hash Tables (DHTs) are used to enable peers to query each other (Harren, et al., 2002). A DHT is a table of keys derived via a hash function from data about the peer. Each peer publishes its key on the network. A peer can query the network by examining the keys and using them to determine efficient routing paths between peers. A peer can then maintain a routing table which is a description of the peer's links to neighbouring peers in its network.

A peer-to-peer network is an example of an overlay network in which a node (peer) chooses its neighbours according to the DHT topology. This is achieved by use of a greedy algorithm where the peer choses the neighbours based on the goal of the shortest routing path (Zhang & Hassanein, 2012). This means that a peer obtains content from another peer over the lowest possible number of routing steps based on the key values in the routing tables. The peer searches for a neighbour with a key value that represents the desired separation minimum distance between the peers and a route is established for content transmission. If no such neighbour exists the peer will search for the next closest neighbour that has the desired key value and so on. In addition to selecting neighbours for routing in a P2P overlay network, greedy algorithms have also been employed to determine chunk selection in video file sharing P2P networks (Zhou, et al., 2011).

2.2.2 Conclusions

The access network bandwidth reached a level that could sustain real time streaming of HD video content in about 2010/2011. This coincided with coding developments that led to true HD codecs that are intrinsically adaptive such as MPEG-DASH

By 2014 HD (720p) video could be streamed at 4Mb/s at 25Hz frame rates but full HD (1080p) at current HDTV framerates (50Hz or 100Hz) still requires much higher bandwidth; for example, 720p at 100Hz would need approximately 16Mb/s. When considering the current 4k HD available from Netflix the demand for bandwidth is higher still. Even though access network bandwidth of 100Mb/s plus are now common, the practical limit to wireless home networks is approximately 50Mb/s. Thus, contention for bandwidth in the modern home is still an issue.

Adaptive steaming facilitates some degree of quality control. By buffering different layers of content chucks, with differing quality levels, real-time adaptation of video steam quality to changes in bandwidth is possible. In addition, adaptive streaming protocols necessitate the inclusion of data about the content within the packet containing the video file. This data could be queried to obtain information relevant to QoE in almost real-time. Consequently, it should be possible to make decisions about chunk and layer selection based on QoE related parameters also in near real-time.

2.3 Video Quality of Experience

2.3.1 Background

In the late 1990s the internet was perceived as providing only a best efforts service (Xiao, 1999). Traditional telecommunication network performance measures have focused on Quality of Service (QoS). From the late 1990s QoS metrics began to be applied to internet services. QoS parameters are objective and easily quantifiable, such as packet loss, time outs etc. However, in the context of video on demand (VoD) there may not necessarily be a direct correlation between QoS and the service quality perceived by the user, the Quality of Experience (QoE)

A definition of QoE is *"the overall acceptability of an application or service, as perceived subjectively by the end-user."* (Stankiewicz & Jajszczyk, 2011). QoE for video is a subjective estimate for the perceived quality of the user when accessing video content over a network and is a function of two components; obtaining and watching. Factors influencing QoE when obtaining video content include delay in starting, ease of sourcing and the network architecture. There are three major factors influencing QoE whilst watching video content; intrinsic quality, device capability, and network effects:

- **Intrinsic quality** – Temporal (frame rate), Spatial (Resolution), and fidelity (original coding quality – Mpeg3, Mpeg4 etc.)
- **Device capability** – available bandwidth, buffer size, and device resolution
- **Network effects** – all QoS parameters that may affect streaming capability (packet loss, bandwidth, routing protocols), topology (P2P, Client server and all architectural implications), and local congestion (home network competition for bandwidth)

Research in video streaming QoE is relatively new. The earliest papers found date back to 2003. The subjective nature of QoE necessarily means it is difficult to measure directly (Lindeberg, et al., 2011). There is a body of work on video QoE consisting of user surveys and subsequent interpretation of their results. In such studies video clips of varying quality are presented to viewers and their perception of quality is assigned some arbitrary score. (Fu, et al., 2010) The QoE for a video clip is then inferred from the Mean Observation Score (MOS).

2.3.2 Objective Estimation of QoE

Clearly, it is a significant challenge to adapt devices or networks in real time to subjective measures of QoE and some work has been done on mapping of QoE to objective (QoS) parameters that can be measured in real time (Alreshoodi & Woods, 2013). In addition, studies have been performed to determine the relative influence of different QoS parameters on QoE. Venkataraman & Chatterjee (2011) propose a model for the subjective assessment of QoE using QoS parameters. A good correlation between the model and survey results is demonstrated. Furthermore, the model was used to develop a framework for measuring real time QoE for video streaming over a network and suggests a method for adaptation of key parameters to optimise QoE. The outcome of this work suggests internet service providers (ISPs) potentially have the option to affect QoE at the application layer by controlling QoS parameters at the network layer.

In general, objective quality assessment methodologies can be categorized into five types. (NTT Network Technology Laboratories, 2015). These are media-layer models, parametric packet-layer models, parametric planning models, bit stream layer models, and hybrid models as shown in Figure 2-2.

Media-layer models use video signals to predict QoE. They do not require knowledge about the system under test, such as QoS parameters like codec type or packet loss. Therefore, they can be applied to the evaluation of unknown systems for the purpose of optimisation.

Parametric packet-layer models predict QoE from packet-header information. Such models are limited because they do not read the payload information. Therefore, these models cannot evaluate the content dependence of QoE.

Parametric planning models use quality planning parameters for networks as inputs. As such a priori information is required about the system under evaluation. Such models are more prevalent in fixed and mobile telephone network planning.

Bit stream-layer models are a combination of media-layer models and parametric packet-layer models. They read encoded bit stream information and packet-layer information so that they can incorporate the content-dependent quality evaluation characteristics.

The fifth model is the hybrid model, which is a combination of the previous models. Hybrid models are perhaps the most powerful models since they evaluate as much information as possible to estimate QoE. Necessarily, this comes with a computational overhead.

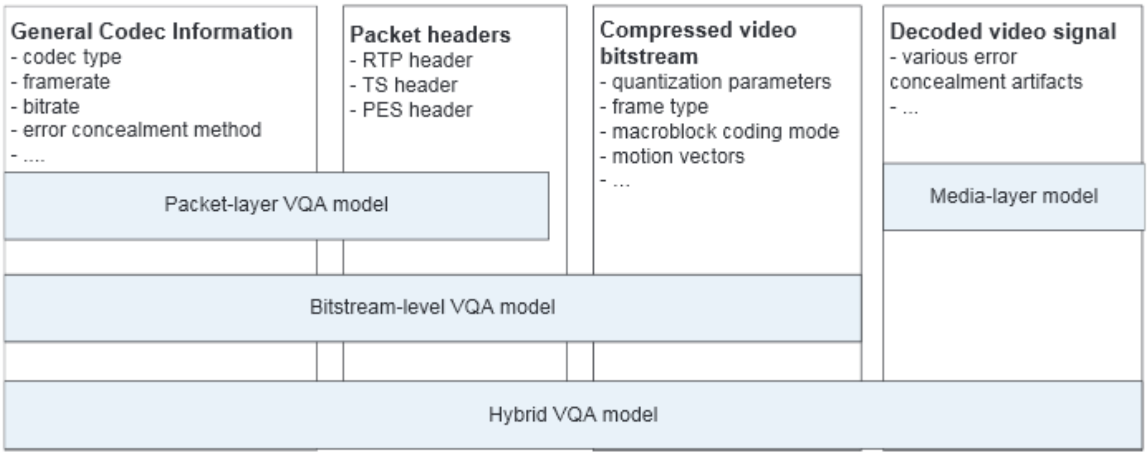

FIGURE 2-2 - DIFFERENT OBJECTIVE MODELS FOR QOE (LIAO & CHEN, 2011)

The choice of model lies with the researcher and, in the context of assessing a QoE, the most important task is to validate the model against subjective studies (Li & Lee, 2015) (Hoßfeld, et al., 2015). When using multi-layer models the task of identifying the most effective QoS parameter adaptation to optimise QoE is the key one. Statistical models for mapping QoS to QoE have been used as a basis for control parameter policy setting in "*QoE-aware QoS management"* (Agboma & Liotta, 2008). In this study a modelling technique is employed which correlates QoS parameters with estimates of QoE perceptions and identifies the degree of influence of each QoS parameter on the user perception. It is proposed that this methodology is applied towards QoE-aware QoS management for networks.

There are intrinsic video quality parameters that directly affect QoE (Ljubojevic, et al., 2014). The basic fidelity of the video and the resolution of the display are key parameters (Joskowicz & Ardao, 2010). Clearly, a high definition video codec (e.g. MPEG-4) has a higher intrinsic quality than a standard definition codec (e.g. MPEG-2). Joskowicz & Ardao (2010) propose a model for real time assessment of QoE for different codecs, bit rates, and display formats. Using subjective assessment of 1500 processed video clips (coded in MPEG-2 and H.264/AVC, in bit rate ranges from 50 kb/s to 12Mb/s, in SD, VGA, CIF and QCIF display formats) a MOS was derived for each clip viewed at different bit rates. The videos chosen represented 3 levels of movement; low, medium and high. It was shown that for all codecs and resolutions that a MOS of greater than 4. 5 on a scale of 0-5, was achieved at bit rates of 4Mb/s. Further, the rate of change in MOS declines rapidly after 5 Mb/s. This is illustrated in Figure 2-3 and Figure 2-4

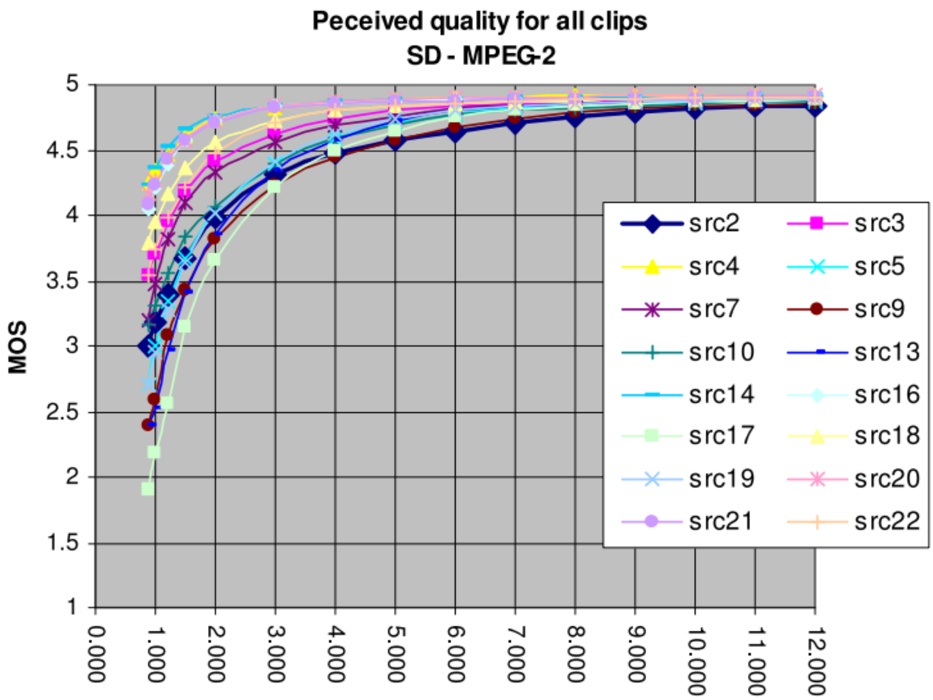

FIGURE 2-3 - MOS FOR ALL STANDARD DEFINITION CLIPS (JOSKOWICZ & ARDAO, 2010)

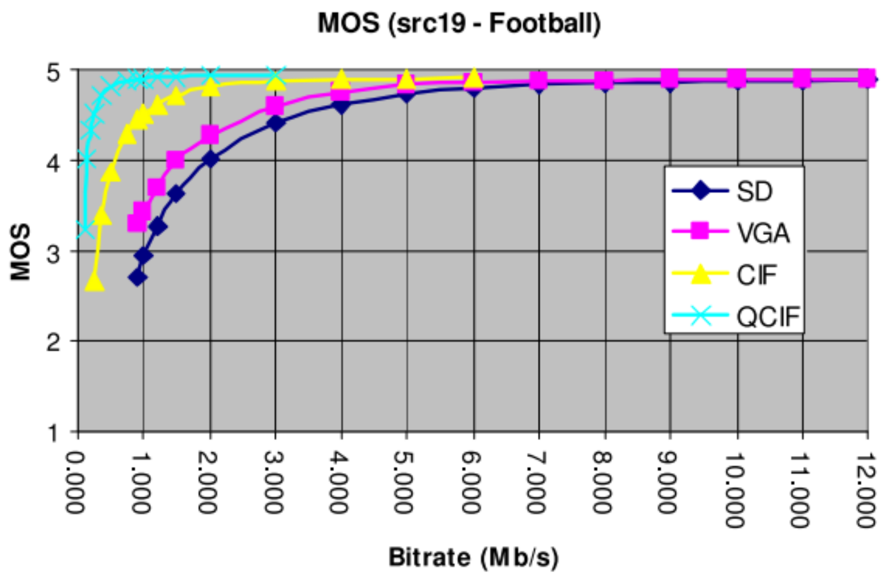

FIGURE 2-4 - MOS FOR HIGH MOVEMENT CLIP IN 4 CODECS (JOSKOWICZ & ARDAO, 2010)

By establishing a model using coefficients based on codec type, video display format, key frame interval, and video display size Joskowicz & Ardao (2010) were able to replicate the MOS scores derived subjectively. The proposed model builds on established standards for modelling video quality, such as ITU-T G1070, and allows real time estimation of QoE based on QoS parameters and content based coefficients:

$$Vq = 1 + 4k\left(1 - \frac{1}{1 + (ab/v_4)^{v_5}}\right)$$

Where *Vq is the Video Quality Metric,*

b is the bit rate, in Mb/s,

a is related to the display format,

v_4 and v_5 are coefficients related to the movement content,

k is the Codec Enhancement Factor based on coefficients k_1 and k_2,

k = 1 for MPEG-2,

$k = 1+k_1.e^{-k_2.a.b}$ for H.264.

Native video quality alone is not enough to guarantee QoE; slow loading and buffer run-out will lead to a reduced QoE regardless of the codec or resolution. However, if some basic information about video content is known (codec, content type, display resolution) and delivery bit rate can be derived then it is possible to determine the QoE of video streams without subjective assessment. Clearly, this is potentially quite powerful and if one could control the bit rate for an individual stream, one could control QoE. Furthermore, regardless of the codec quality, resolution, and level of movement in the video, it appears that satisfactory QoE can be achieved at relatively modest bit rates.

2.3.3 Adapting QoE

As discussed in Section 2.2.1, current video delivery mechanisms include some form of adaptation, the most prevalent being MPEG-DASH. The adaptation is essentially a method of maximising the quality of video (in terms of resolution and frame rate) within the prevailing available bandwidth. MPEG-DASH is a more sophisticated adaptive mechanism than that shown in Figure 2-1 and is represented in Figure 2-5. It provides a method of maximising the quality of video playback for a single device during streaming by adapting to fluctuations in available network bandwidth.

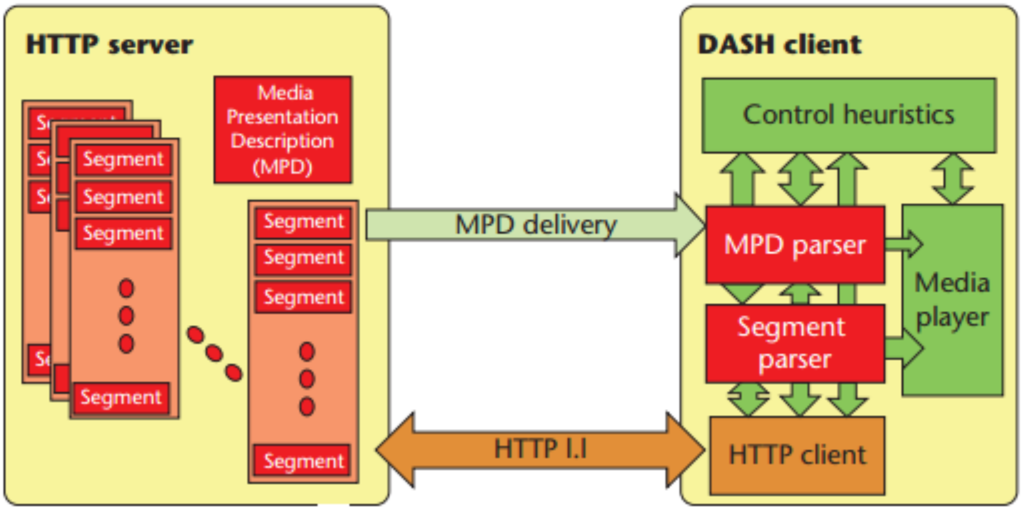

FIGURE 2-5 - MPEG-DASH (THANG, ET AL., 2012)

The video content is stored on a HTTP server in 2 parts. (Thang, et al., 2012). The first part, Media Presentation Description (MPD), describes the available content and characteristics. The second part contains the media segments which are the actual multimedia bit stream files in the form of chunks. The playback device hosts the DASH client which first obtains and parses the MPD so the device learns the content characteristics; availability, media types, resolutions, minimum and maximum bandwidths, the existence of encoded alternatives of multimedia components, accessibility features, digital rights management (DRM), and locations on the network.

The DASH client selects the appropriate media file and starts streaming the content by fetching the segments using HTTP GET requests. After initial buffering, the client obtains the subsequent segments and monitors the network bandwidth fluctuations. Depending on bandwidth, the client adapts by fetching segments of different alternatives (with lower or higher bitrates) to maintain an adequate buffer based on the measured TCP throughput of recent chunks. The DASH client chooses a quality level that has a lower bit rate than the measured throughput. In this way buffer run out is avoided because the download rate is always higher than the playback rate.

From a QoE perspective, the assumption that underpins MPEG-DASH is that the user is satisfied with quality on the basis that buffer run outs are minimised or eliminated. However, this assumption relies on the user being satisfied with lower framerate and resolution for some portions of playback and is not concerned with the transitions between quality levels as the bandwidth fluctuates. Key factors here are the prevailing bandwidth, the rate of fluctuation, and the speed of transition between quality levels. If the bandwidth is large and stable and transition rates high enough to be imperceptible then it follows that QoE will be high. The converse is also true.

A QoE aware DASH system has been proposed in order to address the issue of perceptible transition between intrinsic video quality levels (Frame rate and resolution) (Mok, et al., 2012). The principle of DASH and other adaptation schemes is to select the most suitable quality based on the received bit rate as measured by TCP throughput. The streamed video clips are divided into chunks that are effectively files consisting of groups of pictures. The chunk size usually represents between two to ten seconds of video. QDASH works by reducing the choice of chunk download rate. This accelerates the selection process making QDASH more sensitive to the change in available bandwidth. QoE was estimated using a small survey and it was found that users perceived higher QoE with QDASH than with other adaptation schemes. Mok et al (2012) suggest that the algorithm could be adapted such that download rate may be adjusted to match the available bandwidth and device buffer rate according to the intermediate quality levels preferred by the users surveyed. However, this is based on a very small sample size (19 users) and the algorithm would have to be re-configured by repeat subjective assessment.

Scalable video coding (SVC) is another scheme for adapting playback quality to available bandwidth (Huysegems, et al., 2012). It is the Annex G extension to the H.264/AVC coding standard (Schwarz, et al., 2007). The objective of SVC is encoding of a high-quality video bit stream that also contains one or more subset bit streams of lower quality. Subset video streams are derived from the high quality stream by dropping packets, this reduces the bandwidth required for the subset bit stream. The subset bit stream comprises a combination of one or more reduced dimensions; a lower spatial dimension (lower resolution), a lower temporal dimension (lower frame rate), and/or a lower quality video signal (higher PSNR etc.). Any combination of these dimensions corresponds to a “layer” and it is a fundamental assumption that layers may be mapped to QoE. SVC is based on the delivery of a base layer that conforms to H.264/AVC and the addition of improved layers when bandwidth allows. In Figure 2-6 the base layer is represented by the block in the bottom left corner. Each block above, to the right, and to the rear represents a layer of higher quality than the base datum. Each higher quality layer consumes greater bandwidth.

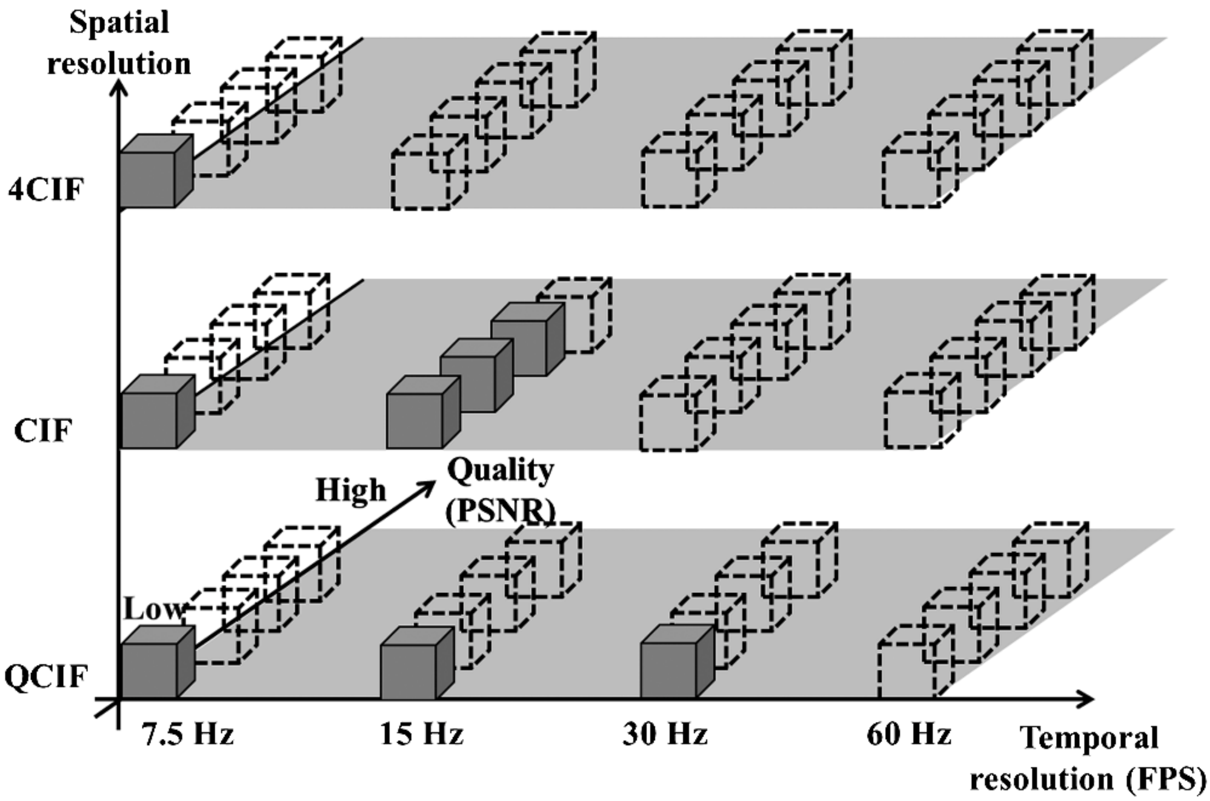

FIGURE 2-6 - SVC LAYER MODEL

Like DASH, SVC is based on the transmission of video files by breaking them into partial bit streams or chunks where the chunks have different levels of the three quality dimensions. This enables functionalities such as graceful degradation in "lossy" transmission environments i.e. the quality of the video stream may be reduced to adapt to inconsistent network conditions without suffering catastrophic QoE degradation such as buffer run-out. According to Schwarz (2007) *"SVC has achieved significant improvements in coding efficiency with an increased degree of supported scalability relative to the scalable profiles of prior video coding standards."* In addition, SVC has been found to improve the QoE of real-time video streaming over wireless networks because it robust in terms of network environment changes and transmission errors. (Schierl, et al., 2007)

Also like DASH, SVC uses layers of chunks of video at varying quality and is based on the assumption maximising quality of video via adaptation to varying bandwidth equates with improved QoE. Using streamed layers for adapting to bandwidth was proposed as early as 2000 (Rejaie, et al.) SVC and DASH are the current popular standards for real-time adaptation.

As with other video systems, measurement of QoE is a challenge for SVC. Subjective assessment of QoE is of no benefit to adaptation and objective real time estimation of QoE may not correlate well with perceived video quality. Recently however, attempts have been made to develop models of objective QoE estimation that could be used to adapt SVC streaming to optimal QoE. Using a hybrid subjective/objective assessment tool, PSQA (Pseudo Subjective Quality Assessment tool), Singh et al (2011) produced an accurate model for QoE estimation which could be used for real time QoE feedback for layer switching adaptation. The model is based on a Random Neural Network (RNN) algorithm used to map QoS and network parameters to QoE. In doing so the most influential parameters on QoE were identified, the chief influence being packet loss. In SVC the video transport packet, NALU (Network Abstraction Layer Unit), can only carry information relevant to a single layer of SVC. Unsurprisingly, it was found that loss of a base layer NALU had a larger impact on QoE than loss of a NALU from higher layers, since loss of a base layer NALU is propagated throughout all other higher quality layers. The successful use of RNN to model subjective assessment does pose a challenge to real time adaptation; it is unclear how the initial algorithm "training" period would be accommodated in real-time, nor whether training would be required for each stream.

In the development of Scalable Video Delivery system using Peer-to-Peer Networks (SVDN) (Qiao, et al., 2015) it has been demonstrated that devices can be made to adapt QoE by SVC layer selection appropriate to the prevailing network conditions. Using peer-to-peer networks, algorithms were designed to improve the efficiency of chunk selection and packetisation[16]. SVDN was shown to reduce the overhead associated with packetisation prevalent in SVC systems from the typical 9 to 15% down to virtually zero. Whilst this does not specifically adapt to QoE any differently to SVC, it frees up resources that facilitate less perceptible layer switching so QoE is maintained during active adaptation.

In further work on SVC in the peer-to-peer environment it has been shown that it is possible to use QoE estimation for real-time adaptation decision making (Ruckert, et al., 2012). In this work the combination of application of centralised QoE policy and distributed adaptation algorithms was used to adapt SVC layer selection based on the maximum QoE achievable at the device node. Periodic monitoring of the parameters affecting QoE took

[16] Division of the video stream in to discrete chunks containing video files and packet headers

place during video streaming and algorithms were employed to decide on layer switching implementation.

DASH, SVC, and the adaptive derivatives described above, base adaptation on video stream quality parameters for one device within the constraints of network parameters. An alternative QoE adaptation approach, for network management with multi devices and video streams, is to adapt based on content. From a network management perspective the goal is to optimise QoE at the lowest possible network cost. In effect, this means that the aim of the network manager is to deliver video stream of a quality that just matches each users different requirement, so maximising the network capacity for streaming. To satisfy the network manager Agboma and Liotta (2008) propose a QoE aware QoS management framework. After determination of the QoS parameters that influence QoE for different content types, movie, news etc., they devise a method of degrading the quality of a video stream to match the user expectation at the minimum bandwidth consumption. This resulted in a 15% reduction of network resource consumption overall and significantly more for some content types as illustrated in Figure 2-7.

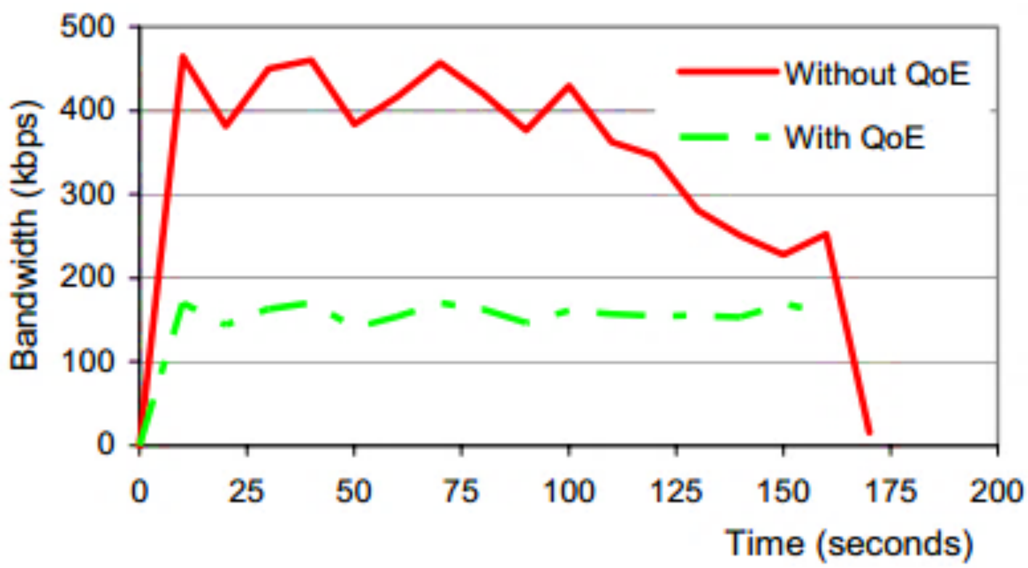

FIGURE 2-7 - THE EFFECT OF QOE AWARE NETWORK MANAGEMENT (AGBOMA & LIOTTA, 2008)

Solutions discussed thus far are based on adapting the video quality to the prevailing network conditions. The QoS parameter that has a major detrimental influence on QoE is packet loss or error. An interesting alternative approach to adapting to measure QoE is to detect and repair the errors as they occur. (Asghar, et al., 2009). It was found that the streaming IPTV encounters access network (last mile) challenges; network congestion, bit errors on access lines, slow channel change times, and limited ability to monitor per-subscriber video quality. By using a model for subjective QoE assessment it was possible to improve quality by deploying intelligence at device level. In this case, between the aggregation networks and IP set-top box (STB) to repair packet losses and speed up channel change times.

2.3.4 Conclusions

In the context of this study, Quality of Experience (QoE) is a subjective assessment of the user perceived quality of video playback during the steaming process. QoE from an objective standpoint is an assessment of quality that extends to factors beyond the native quality of the video clip to encompass other quality parameters.

QoE can be objectively estimated using models based on native video quality and transmission QoS parameters. Many models have been proposed and developed for the estimation of QoE in real-time and the best of these have been shown to map well to the subjective QoE assessment.

It has been generally shown that the most influential QoS parameter is bit rate which translates to available bandwidth with which to deliver the video. For SVC the most influential QoE parameter is loss of SVC base layer NALU which propagates through all higher SVC layers.

Algorithms have been developed and used to adapt QoE during playback. Many of these algorithms are based on adapting the download rate to the available bandwidth by selecting chunks with video quality (resolution and frame rate) appropriate to the bandwidth

Random Neural Network (RNN) algorithms have been shown to be successful in real time estimation of QoE. For SVC the PSQA model has been shown to be the most accurate.

Using Bitrate, packet layer and codec information it is possible to produce objective models of QoE. Thus if a device can be made “aware” of this information it is possible to define a set of expected QoE (MOS) scores for the device any given matrix of causal parameters; bitrate, content, codec, frame rate and so forth. A bit stream-layer model can be used to accurately estimate QoE based on both encoded bit stream information and packet-layer information such that the content-dependent quality evaluation characteristics are taken into account

Options for adaptation include single device video stream adaptation or multi device network management. In both cases AI algorithms are employed to model QoE and, in the case of network management algorithms, they are employed to detect network conditions and adapt or repair the video stream.

Work on P2P streaming has shown that SVC layer section algorithms can be used to adapt QoE to prevailing network conditions and that it is possible to combine such adaptation with device QoE capability policy rules

2.4 Artificial Intelligence and Network Management

Thus far consideration has been given to video delivery mechanisms and their influence on quality of experience. In this section the ability to control QoE from a network parameter perspective is examined.

The concept of network management is founded on the network manager's need to control cost whilst providing a service to a level that satisfies consumers of network services (Boutaba & Xiao, 2009). From this perspective, user QoE targets are subservient to maximising the use of network resources, i.e. the network manager tries to degrade the quality of the service provided to a minimum acceptable level and thus make the most efficient use of bandwidth. Of course, there are other considerations for the network manager such as fault detection and repair. Over the last decade there has been a growth in the use of technology to create self-managing (self-monitoring, self-detecting, and self-repairing) networks (Boutaba & Xiao, 2009).

In 2001 Paul Horne of IBM stated that autonomic computing *"is the single most important challenge facing the IT industry"* (Ganek & Corbi, 2003). Autonomic computing is defined as *"computer systems that regulate themselves much in the same way as our autonomic nervous system"* (Nguengang, 2011). This concept of self-regulation was beginning to be applied to communications networks in 2004. Figure 2-8 shows how research in this area has matured quickly with significant FP6, FP7 and EU research projects being undertaken from 2006 to the present (Kuklinski, 2012).

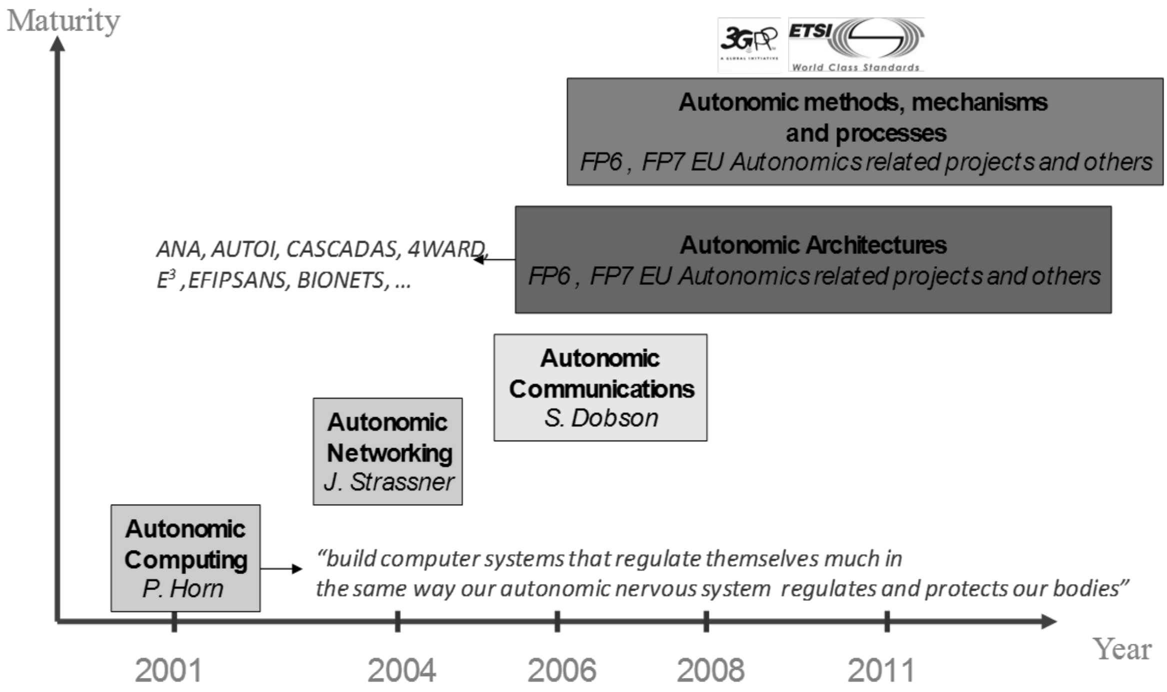

FIGURE 2-8 - GROWTH OF AUTONOMIC NETWORK MANAGEMENT (NGUENGANG, 2011)

A key feature of autonomic networks is the application of policy that controls the behaviour of the network in response to prevailing conditions and changes to those conditions. Changing the behaviour of the network is achieved by reconfiguring network parameters by using control loops (Kephart & Chess, 2003).

Policy will determine what the target KPIs (Key Performance Indicators) are for the network and what to do when they are breached. Traditionally, the KPIs are based on network parameters that influence QoS. The challenge for autonomic networks is implementing central policies across a distributed network with many different network and end user devices – network elements. One solution to this challenge is the closed control loop applied to each network element as shown in Figure 2-9 (Kuklinski, 2012). The control loop consists of a 3 components; sensing, acting and knowledge. The sensing part detects changes to the network parameters, "knowledge" of the systems is used to analyse the changes and implement policy decisions, the acting part of the loop implements the decisions by reconfiguration.

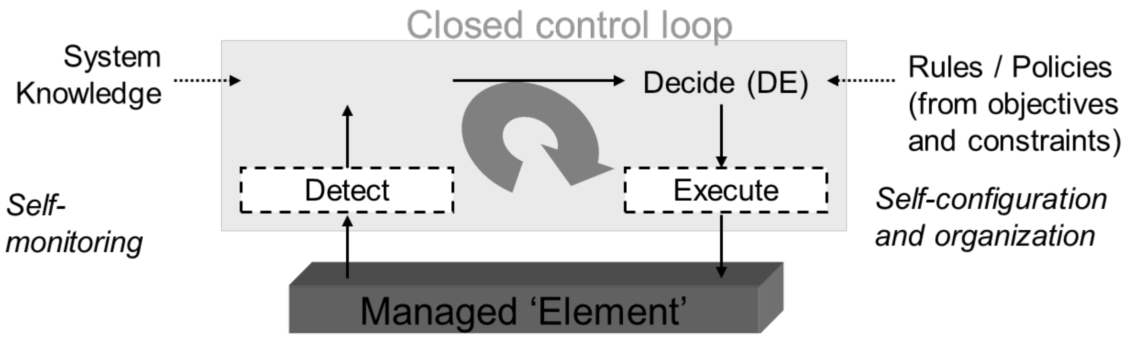

FIGURE 2-9 - AUTONOMIC NETWORK ELEMENT MANAGEMENT (KUKLINSKI, 2012)

There is a contrast between the role of the adaptation algorithm and the role of the learning algorithm in the context of network management. Adaptation agents, whether deployed centrally, locally or hybrid fashion are based on application of policy to affect changes based on measured conditions (Russell & Norvig, 2010). Leaning algorithms can be employed to measure the effect of adaptation and change the rules in order to improve adaptation (Kuklinski, 2012). There have been a number of recent collaborative studies focused on the application of machine learning to network management (Celtic Plus, 2014) (Univerself, 2015). These studies have proposed learning algorithms for future telecommunications networks and demonstrate the ability to cope with network uncertainties, such as nodes dropping in and out of service or incomplete sensor data, and still perform positive adaptation functions.

In complete contrast to the network management approach, P2P content sharing networks focus entirely on distributed agents to affect adaptation. Work on P2P networks has an increasing focus on QoE based adaptation and design. Learning algorithms have been proposed for QoE based P2P video content optimisation (Menkovski, et al., 2010). Ruckert et al (2012) have extended the algorithm for chunk selection to include QoE based agent decision making and claim improvements on QoE at little or no bandwidth cost. QoE has even been proposed as the driver for design of P2P video streaming networks (Couto da Silva, et al., 2008)

The layer selection process proposed by Ruckert et al (2012) is significant because it potentially conveys efficiencies in the process. Currently, SVC single device adaptation is based on "*the goal to maximize the bandwidth utilization at the peers and chooses the layer, out of the compatible ones, that has the highest bit rate.*" This strategy is implicitly QoE dependent since it is reasonably assumed that higher bit rate equates to higher quality.

However, Ruckert et al (2012) propose a 2 step strategy: The first step enables the selection of the SVC layer at the start of the steaming session. The SVC layer chosen is based on prior knowledge of QoE associated with each layer. This knowledge is the result of previous QoE assessment experiments and is applied at the server that is the source of the video stream. This enables layer selection based on device capability such as resolution and available bandwidth. Once streaming starts the second adaptation step takes over and the available resources at the device are sampled and layer changes are applied accordingly. In order to smooth the transition from one SVC layer to another, and minimise the impact on QoE during transition, switching is not performed in a single step. Rather the switching is performed in a series of steps by implementing the Dijkstra algorithm to produce an SVC graph and calculate the target layers for each step required for transition that have the least disruption on QoE (Cormen, et al., 2009). It was found that it was necessary to limit the number of steps per transition and 3 seconds was chosen per switch with a total adaptation interval of 10 seconds; allowing three steps per adaptation.

In both P2P and client server architectures examples discussed, the adaptation of QoE either by policy application, local device adaptation or a combination of both, relies on some form of artificial intelligence. The intelligence is implemented by one or more machine learning algorithms.

2.4.1 Comparison of Learning Algorithms

This section presents an overview of common machine learning algorithms and the situations in which they are applicable. Particular attention is given to those algorithms applicable to network management and adaptation. At the time of writing there are many machine learning algorithms in use today and they can be grouped into learning categories based on the desired outcome of the algorithm and/or the type and structure of input data.

There are the four common learning classifications; supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning (Russell & Norvig, 2010). When analysing data to model business processes supervised and unsupervised learning methods are employed. Currently, Big Data is an area of high research activity and semi-supervised learning methods are employed; for example, satellite image classification where there are large datasets with very few labelled examples (European Space Agency, 2014). Reinforcement learning is more likely used in system control and robotics and is most applicable to network management scenarios.

Selecting the algorithm classification that most fits the problem is the first step to choosing the correct algorithm. In this section machine learning algorithms are associated with the learning type and are explained in terms of their functionality

2.4.1.1 Supervised Learning

Supervised learning algorithms are trained on *labelled* data i.e. input data has a known result (label) such as SVC layer or network QoS parameter. The supervised learning algorithm attempts to create a general function or mapping from inputs to outputs which can then be used to generate an output for new inputs. The algorithm creates a model via a *training process*. During training the model is fed with training data in the form of input-output pairs. The model is then updated based on the accuracy of the outputs until the model achieves a desired level of accuracy (Russel & Norvig, 2010):

Given a training set of N example input-output pairs:

$$(x_1, y_1), (x_2, y_2), \dots . (x_N, y_N)$$

Where each y_j *was generated by an unknown function* $y = f(x)$

Discover the function h that approximates to the true function f

When the outputs are Boolean values the learning problem is one of Classification, when the outputs are numbers the problem is one of Regression.

Regression is concerned with solving a problem by establishing an average value of the output *y* by using a measure of error in the predictions made by the model. Estimation of a the equation for a straight line is using the Ordinary Least Squares method is probably the best known regression algorithm (Moore, et al., 2012):

The equation for the straight line:

$$\hat{y} = a + bx_i$$

Where:

$\hat{y}$ *is the predicted value of dependent variable* y_i

a is the intercept on the y axis when x = 0

b is coefficient that represents the slope of the line

x_i *is the observed value of the independent variable for the* i^{th} *case.*

The OLS method of calculating slope b is based on mean values of x and y:

$$b = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

and intercept a;

$$a = \bar{\bar{y}} - b\bar{\bar{x}}$$

Other examples of regression algorithms include:

- Logistic Regression - which is used to solve problems of classification, essentially by defining a boundary between binary classification options (Penga, et al., 2002)
- Multivariate Adaptive Regression Splines (MARS) – which is a regression method used to predict output values for data with non-linear relationships between input-output variables. The non-linearity is expressed as a “hinge” between two contiguous best fit lines (Friedman, 1991).
- Locally Estimated Scatterplot Smoothing (LOESS) – which is used to model complex processes for which no theoretical models exist (Cleveland & Devlin, 1988).

Instance based learning model a decision problem with instances or examples of training data that are deemed important to the model. They are examples of non-parametric models; *models that cannot be characterised by a bounded set of parameters* (Russel & Norvig, 2010). In contrast, the models discussed previously are parametric models where the parameters are known; for example x and y for linear regression. Instance based learning methods typically build up a database of examples and compare new data to the database using a similarity measure in order to find the best match and make a prediction. A simple example is classification based on a look-up table where for a value x the corresponding value *x* in the table relates to a value *y* which is returned. A disadvantage of the look up table is that if *x* does not exist in the table then no value of *y* can be returned. Examples of algorithms that improve on the look up table are:

- k-Nearest Neighbour (kNN) – which improves on classification by using not only the look up value but other values close to it. Regression is applied to improve the classification prediction (Guo, et al., 2003)

- Learning Vector Quantization (LVQ) – which is used to model pattern recognition based classification (Kohonen, 1995)

Decision tree methods are the simplest and most successful machine learning processes (Russel & Norvig, 2010). Decision trees are models that use a range of input attributes to return a single output value, the decision. Decision trees are trained on data for classification and regression problems and there are many variants:

- Classification and Regression Tree (CART) (Berk, 2008)
- Iterative Dichotomiser 3 (ID3) (Quinlan, 1986)
- Random Forest (Breiman, 2001)
- Multivariate Adaptive Regression Splines (MARS) (Friedman, 1991)
- Gradient Boosting Machines (GBM) (Friedman, 2002)
- Bayesian methods – which are regression and classification algorithms where the predictions are weighted based on Bayesian probabilities (Russel & Norvig, 2010)

Kernel Methods, sometimes referred to as Kernel tricks (Russel & Norvig, 2010), are methods that aid classification and regression by mapping input data that is not easily separated or correlated, into a higher dimensional space where some classification or regression problems are easier to model.

- Support Vector Machines (SVM) – very useful methods if there is no specialist prior knowledge about the problem domain (Hearst, et al., 2002)
- Linear Discriminant Analysis (LDA) - methods of classification based on a linear combination of data attributes (Natha, et al., 1992)
- Radial Basis Function (RBF) – methods of determining classification based on separation distances from some centre point and the basis of learning networks (Broomhead & Lowe, 1988)

Artificial Neural Networks are models that are mathematical interpretations of brain neurons. They are a class of pattern matching that are commonly used for regression and classification problems. The networks simulate the neuron by combination of an input function and an activation function. The activation function is only implemented when the *weight* of inputs exceeds a threshold. Neurons in the network are connected by input and output links (Russel & Norvig, 2010). Some popular methods include:

- Perceptron – An early classification network (Rosenblatt, 1958)
- Back-Propagation (Reidmiller, 1994)
- Hopfield Network (Behl, et al., 2013)
- Self-Organizing Map (SOM) (Kohonen, 1995)
- Learning Vector Quantization (LVQ) (Kohonen, 1995)

A difficulty with all algorithms is model selection. Some algorithms will under-fit the training data and some will over-fit the training data. The challenge is selecting the model that provides optimum fit with the training data (Russel & Norvig, 2010). **Regularization Methods** are a set of algorithms (typically regression methods) that help decide on the optimum model. They operate by penalising models based on their complexity and favouring simpler models. Regularisation methods are generally modifications made to other methods.

- Ridge Regression (Hoerla & Kennard, 1970)
- Least Absolute Shrinkage and Selection Operator (LASSO) (Tibshirani, 1996)
- Elastic Net (Zou & Hastie, 2005)

2.4.1.2 Unsupervised learning

Unsupervised learning algorithms operate on *unlabelled* data and learn patterns in the input data without feedback. The desired outcome is to discover some groupings and relations in the data (Russel & Norvig, 2010). Example problems are association rule learning and clustering.

Clustering methods are typically organized by the modelling approaches, such as centroid-based and hierarchal models. They are chiefly concerned with using the inherent data structures to organize the input data into groups of maximum commonality such that the output are classifications based on commonality. (Russel & Norvig, 2010)

- k-Means (Wong, 1979)
- Expectation Maximisation (EM) (North & Blake, 1998)

Dimensionality Reduction methods are like clustering methods, in that they seek out and exploit the inherent structure in the data to summarise or describe data using less information. This can be useful to visualize dimensional data or to simplify data which can then be used in a supervised learning method. (Russel & Norvig, 2010)

- Principal Component Analysis (PCA) (Mackiewicz & Ratajczak, 1993)

- Sammon Mapping (Sammon, 1969)
- Self-Organizing Map (SOM) (Kohonen, 1995)

2.4.1.3 Semi-supervised learning

Semi-supervised learning methods combine both labelled and unlabelled inputs to generate an appropriate function. A desired outcome exists but the data structures must be learned in order to organize the data so that predictions can be made. As with supervised and unsupervised learning, the algorithms rely on classification and regression functions. Assumptions are made about how to model the unlabelled data and training outcomes are based on some derivation of labelled input data (Russel & Norvig, 2010). Algorithms tend to be extensions to other methods discussed in the previous 2 sections.

Deep Learning methods are updated methods based on Artificial Neural Network. They are concerned with building much larger and more complex neural networks. Many methods are concerned with semi-supervised learning problems where large datasets contain very little labelled data (Deng & Yu, 2014).

- Restricted Boltzmann Machine (RBM) (Salakhutdinov, et al., 2007)
- Deep Belief Networks (DBN) (Lopes & Ribeiro, 2015)

2.4.1.4 Reinforcement learning

Reinforcement learning consists of methods where the outcome of the algorithm is defined in terms of *reward* or *punishments*. Feedback is not the result of a training process but as punishments and rewards in the environment. The agent performs actions which cause the observable state of the environment to change. Through a sequence of actions, the agent reinforces its knowledge about how the environment responds to its actions. Finally, the agent will select the sequence of actions that maximises a cumulative reward (Russel & Norvig, 2010).

Markov Decision Process (MDP) methods are the most well-known reinforcement learning algorithms (Szepesvari, 2009) where transitional probabilities are calculated to estimate the rework at each state. Markov Decision Processes are a tool for modelling sequential decision making problems where a decision maker interacts with a system in a sequential fashion.

Monte Carlo methods (or Monte Carlo experiments) are a class of computational algorithms that rely on repeated random sampling to obtain numerical results. Simulations are performed many times in order to obtain the distribution of an unknown probabilistic entity (Gentle, 2003). The name comes from the resemblance of the technique to the act of playing and recording your results in a real gambling casino. They are often used in physical and mathematical problems when it is difficult or impossible to obtain an analytical expression, or to apply a deterministic algorithm. Monte Carlo methods are useful in solving optimisation problems.

Temporal difference (TD) learning is a prediction method that resembles a Monte Carlo method because it learns by sampling the environment according to some *policy*. TD approximates an estimate of utility at each state transition using a learning rate parameter to update the estimate at each stage. The learning rate parameter is derived from previously learned estimates (Russel & Norvig, 2010). TD is a dynamic prediction method; a prediction is made and when a new observation is available, the prediction is adjusted to better match the new observation.

Q-learning is a reinforcement learning technique that can be used to find an optimal action-selection policy for any given (finite) Markov Decision Process (MDP) (Watkins & Dayan, 1992). It works by learning an action-value function that will produce the expected utility of a given action in a given state. Q-learning then applies the learned optimal policy thereafter. When such an action-value function is learned, the optimal policy can be constructed by simply selecting the action with the highest utility value in each state.

SARSA (State-Action-Reward-State-Action) is an algorithm for learning a Markov Decision Process policy and is similar to Q-Learning. However, it differs in a subtle way (Russel & Norvig, 2010); Q-Learning is an *off-policy* algorithm whereas SARSA is an on-policy learning algorithm where the agent will interact with the environment and update the policy based on actions taken.

2.4.1.5 Comparison of Reinforcement learning Algorithms

The advantages of Monte Carlo Methods (Denning, 2012) are as follows:

- ***Complexity:*** *Simulation often gives better physical visibility of a complex system analysis than a set of equations, aiding interpretation of the output.*
- ***Scope:*** *For example, complex policies are easier to deal with in simulations than analytical models.*

- ***Accuracy:*** *Although analytical models are deterministic, they usually involve simplifying assumptions to make the model analytically tractable. Such assumptions have to be justified.*
- ***Future development:*** *If a model is likely to be further refined and developed, an initial model that may be initially tractable analytically may not be so when further development requirements are placed. A simulation model may therefore be appropriate from the start.*
- ***Application:*** *For quick look analysis, analytical models may be preferred, because of their speed of execution. The repeated running involved in Monte-Carlo simulation can cause long execution times before estimates of system parameters of interest are obtained.*

According to Denning (2012) the disadvantages of Monte Carlo Methods are as follows:

- ***Processing:*** *Usually requires a computer.*
- ***Calculations:*** *Can take much longer than analytical models.*
- ***Precision and accuracy****: Solutions are not exact, but depend on the number of repeated runs used to produce the output statistics. That is, all outputs are estimates.*

One of the strengths of Q-learning is that it is able to compare the expected utility of the available actions without requiring a model of the environment (Chris Gaskett, 2005). Additionally, Q-learning can handle problems with stochastic transitions and rewards, without requiring any adaptations. Gaskett et al (2005) suggest that the main advantage of Q-Learning is exploration insensitivity; the ability to learn without necessarily following the current policy. Q-learning is the reinforcement learning algorithm most widely used for addressing the control problem because of its off-policy update, which makes convergence control easier (Woergoetter, 2008). It has been proven that for any finite MDP, Q-learning will converge to the optimal policy if all state-action pairs are visited infinitely often.

The disadvantages of Q-Learning (Gaskett, et al., 2005) is generally considered in the case that states and actions are both discrete. In some real world situations, and especially in control, it is advantageous to treat both states and actions as continuous variables.

TD-learning will converge to the final value function assigning to each state its final value, if all states have been visited "often enough". This can, however, lead to very slow convergence if the state space is large. (Woergoetter, 2008) For large state spaces and/or sparse rewards convergence may require many steps and can be very slow (Barto, 1983)

2.4.2 Conclusions

When trying to make predictions with known (labelled) data it is best to use supervised learning algorithms. When dealing with big data or unknown (unlabelled) data the unsupervised algorithms are best suited to discover data structure and associations. In the cases of control systems or robotic systems reinforcement learning is the best approach.

For network management problems a key driver is latency of the feedback loop when sensing the environment and triggering actuation (Kuklinski, 2012). Both Monte Carlo methods and Q-Learning are ideal for network control and management. In network environments that have distributed or hybrid agent structure the discrete states of the network nodes and environment make Q Learning or SARSA the most useful algorithms. This is on the basis that the utility of each state could be set as a target QoE. The initial policy could be reinforced during a video streaming session with sampling of the environment providing observations with which to update the policy.

The Ruckert (2012) model of QoE based adaptation could be modified for use in simulation of QoE aware device adaptation in a resource contended home network. Their method of layer selection is based on application of prior mapping of QoE to SVC layer. This could be altered to make use of assumptions regarding device state mapping to a *QoE capability* (Resolution, available bandwidth, content type and so forth) as measured during streaming.

For simulations of SVC layer selection, instance based algorithms may be suitable for learning. Given the dynamic nature of network environment and the potential inaccuracies of QoE estimation in real-time, it may be possible to utilise k-nearest neighbour algorithms to look up target SVC layers for streaming at optimum QoE.

2.5 Conclusions from the literature review

Demand for video streaming is increasing rapidly and is being supplied by large commercial enterprises such as Netflix and Amazon. As a medium of entertainment, video streaming is now a serious competitor to traditional broadcast media. Access network and subscriber (Home) networks are improving with respect to bandwidth, which is one of the key constraints on video QoE. However, the suppliers continue to strive to improve quality in terms of video definition, such as 4k HD from Netflix thus the demand for bandwidth is also increasing. With more family members connecting simultaneously to the internet, contention for bandwidth in the modern home is still an issue.

Video QoE is a subjective assessment of the user perceived quality of video playback during the steaming process but QoE can be objectively estimated using models based on native video quality and transmission QoS parameters. Adaptive steaming allows a degree of quality control in a dynamically changing network environment and real-time adaptation of video steam quality to changes in bandwidth is possible. In addition, adaptive streaming protocols necessitate the inclusion of data about the content within the packet containing the video file. This data could be queried to obtain information relevant to QoE in almost real-time. Consequently, it should be possible to make decisions about chunk and layer selection based on QoE related parameters also in near real-time

The Ruckert model of QoE based adaptation could be modified for use in simulation of QoE aware device adaptation in a resource contended home network. Their layer selection algorithms can be used to adapt QoE to prevailing home network conditions and it is possible to combine such adaptation with device QoE capability policy rules. For simulations of SVC layer selection, instance based algorithms may also be suitable for learning. It may be possible to utilise k-nearest neighbour algorithms to look up target SVC layers for streaming at optimum QoE.

Both Monte Carlo methods and Q Learning are ideal for network control and management. In network environments that have distributed or hybrid agent structure the discrete states of the network nodes and environment make Q Learning or SARSA the most useful algorithms. These algorithms could be investigated as potential solutions to policy adaptation during streaming as devices connect/disconnect from the network.

3 Methodology

Recalling the research question:

"To what extent can cognitive algorithms support the optimisation of Quality of Experience (QoE) and bandwidth utilisation for users when downloading or streaming video to connected devices in a resource contended network?"

It has been shown in the literature review that QoE can be estimated from measurable parameters associated with network conditions, QoS, and intrinsic video quality parameters. It has also been shown that such parameters can be measured in near real-time. Further, it has been shown that modern video streaming mechanisms can be adaptive, leading to some optimisation of video quality parameters in response to bandwidth constraints.

3.1 Quantitative Method Strategy

When the research question was first posed the instinctive philosophical approach was positivist; an attempt was to be made to determine the effect on device level QoE by the implementation of algorithms using the simulation of a resource contended home network, through the use of a controlled experiment.

The quantitative approach was based on the production of home network environment simulators within which device level agents could be used to implement algorithms designed to optimise device level QoE. A baseline of QoE would be established via simulation experiments for several different device and video content combinations. Agent algorithms would then be employed and the simulation experiment would be repeated. The differences between the baseline and the repeat experiments would be analysed to quantify the effectiveness of the algorithms in optimising QoE.

3.2 Architecture

This section describes the logical design of a device manager agent (section 3.2.1) for use in simulation and/or experiment testbeds and the physical architecture (section 3.2.2) in which such an agent could be deployed. The logical design includes the descriptions of the environment (section 3.2.1.1), the device manager agent (section 3.2.1.2), and the high level adaptation algorithm (section 3.2.1.3).

3.2.1 Logical Design

In this section the logical design of the experiment is described. This design is proposed as the basis of both simulation and any further experiment implementation.

3.2.1.1 Task Environment

In order to characterise the environment the approach of Russell and Norvig is adopted (Russell & Norvig, 2010) and the summary description is given in Table 3-1.

TABLE 3-1 - TASK ENVIRONMENT DESCRIPTION

Agent	Performance Measure	Environment	Actuators	Sensors
Device Manager	Device QoE Maximised Device Bandwidth consumption Minimised For all connected devices	Home Network Multiple device types Multiple video streams Other network traffic	SVC Layer selector	Device QoE capability monitor. SVC layer detector Content descriptor Bandwidth detector

The environment is partially observable by the device manager agent, because the agent cannot perceive the number or type of other devices connected to the network, nor can it detect the type of content delivered to, or bandwidth "consumed" by other devices. Since every device connected to the network will have a device manager the environment can be described as multi-agent and competitive. This is desirable for the overall goal is satisfaction for all users by achieving optimum QoE at each device.

There is a degree of uncertainty in the home network environment, such as total available bandwidth (fluctuating due to ISP contention), connection/disconnection of devices, and type and quantity of other data transfer over the network. As such, the environment may be said to be nondeterministic because the subsequent environment states depend on factors other than the current state and the agent actions.

Furthermore, the environment is episodic due to the fact that the device manager actions are not dependent on previous actions nor do they influence future actions; for example, there is no sequence of actions for the device manager, it must simply perceive the QoE layer, device capability, content, and available bandwidth then select the appropriate QoE layer: this is independent of previous selection actions.

Key to the operation of the device manager is the fact that the environment is dynamic and the state can change during the device manager decision making process. This dynamism is continuous and there are no discrete states because bandwidth consumption by all competing devices is continually changing.

The environment characteristics are summarised as follows:

- Partially Observable
- Multi-agent
- Nondeterministic
- Episodic
- Dynamic
- Continuous

3.2.1.2 The Device Manager Agent

The device manager is a model based, reflex agent represented in Figure 3-1. The Device Manager Agent (DVM) must be capable of sensing the device resolution capability, the video quality, content type, and the available bandwidth. The DVM receives this as sensing data.

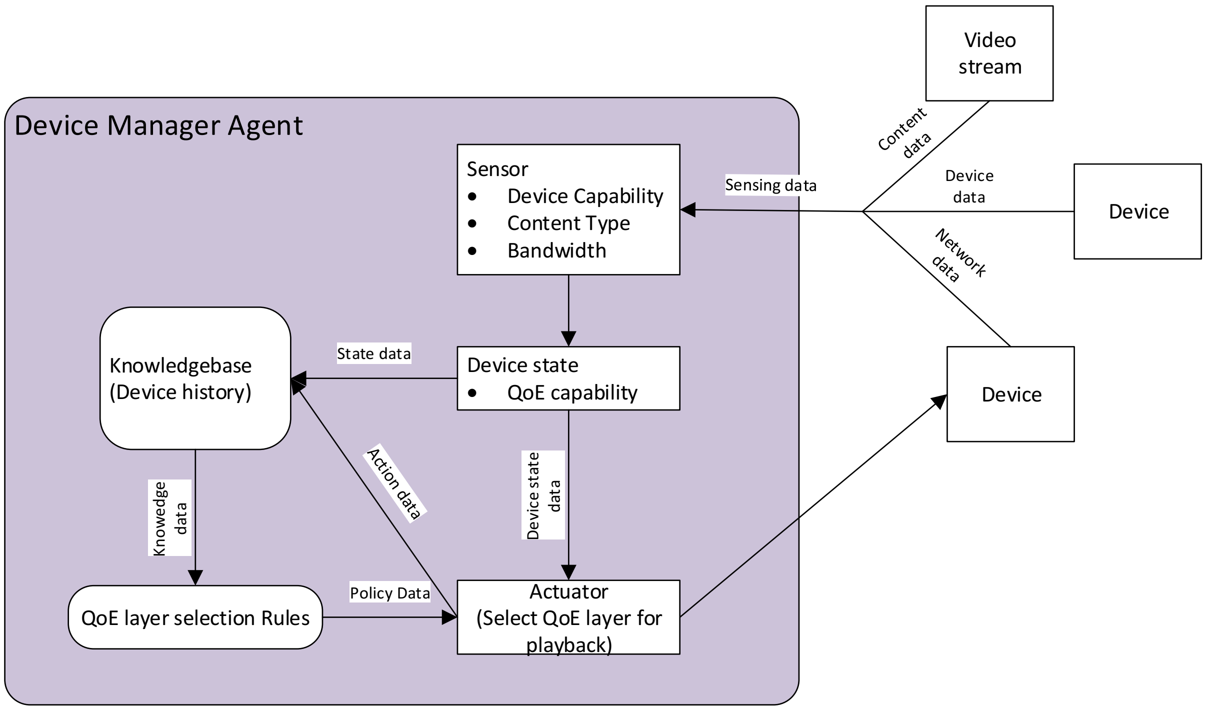

FIGURE 3-1 - DEVICE MANAGER AGENT

The agent is based on the closed control loop as described in Figure 2-9. At a point in time, the actual values of the sensing parameters define the device state relative to a desired QoE target state. The agent must then make changes based on the current state and rules defined relative to target QoE. Action is defined by the local implementation of policy rules which are based on a SVC layer selection algorithm which will enable adaptation of QoE by changing the SVC layers that are downloaded. The DVM must be able to record the sensing/acting decisions in a knowledge base such that future actions could be defined by reference to the knowledge base and policy. In this way the DVM could learn to respond more efficiently and potentially adapt the policy rules.

The DVM rules that follow have been developed based on a baseline device QoE capability - Q_C. In order to simulate $Q_{C,}$ the following assumptions have been made:

Q_C is a function of the device maximum resolution - R_C, the total bandwidth capability of the device network interface card (NIC) - B_C, and the quality of the content being streamed to the device - C_C:

$$\textit{Equation 1.} \qquad Qc = f(Rc, Bc, Cc)$$

During streaming the device state will be measured and can be defined in terms of QoE. The actual QoE - Q_A is a function of the resolution of the video clip – R_A, the actual bandwidth consumed by the download – B_A, and the actual content quality – C_A:

$$\textit{Equation 2.} \qquad Q_A = f\,(R_A,\, B_A,\, C_A)$$

The baseline state of the device can be expressed as actual QoE as a simple proportion of the device capability:

$$\textit{Equation 3.} \qquad \textit{Device State} = Q_A \,/\, Q_C$$

From a SVC perspective, the actual SVC layer L_A is a function of 3 dimensions; resolution, frame rate and native video quality, so;

$$\textit{Equation 4.} \qquad Q_A = f\,(R_A,\, B_A,\, C_A) = f\,(L_A)$$

Furthermore, as each dimension is increased, the file size (clip chunk) increases. Thus the consumption of bandwidth increases and bandwidth consumption – W_A is proportional to the actual SVC layer L_A:

$$\textit{Equation 5.} \qquad W_A \,\alpha\, L_A \,\alpha\, Q_A$$

Similarly, the maximum bandwidth consumed by a device is function of the device capabilities so that;

$$\textit{Equation 6.} \qquad W_C \,\alpha\, L_C \,\alpha\, Q_C$$

Therefore the device state could be expressed as;

$$\textit{Equation 7.} \qquad \textit{Device State} = Q_A \,/\, Q_C = L_A \,/\, L_C$$

It follows that the actual QoE for the user could be estimated from the actual SVC layer being streamed. This layer could be inferred by the DVM based on the bandwidth being consumed and decisions could be made to consume more or less layers as a function of the device capability and content

3.2.1.3 High level Algorithm

Following on from the assumptions in section 3.2.1.2, it is possible to derive a simple layer selection algorithm for the DVM. At the basic level the DVM for a single device will grab chunks of video as SVC layers based on this simple algorithm: If the actual QoE of the device is less than its capability, download the next highest SVC level. Repeat this until the maximum device QoE capability is reached:

- If $L_A < L_{C,}$
- Then select next highest L_A layer
- Do until $L_A = L_C$

In essence this is a simple statement of SVC adaptation for a device. For SVC the device capability changes as a function of available bandwidth. As more bandwidth becomes available higher quality SVC layers are streamed and vice versa.

Central to the research question is the ability to optimise QoE for each device in a network when more than one device is competing for resources. If the SVC based algorithm above is implemented by DVMs for each device in the network each device would grab video chunks at the same rate until the total available bandwidth was consumed. Such a scenario could result in a paradoxical situation where the user watching low quality video in a low capability device perceives higher QoE than the user steaming high quality video on a high capability device. For example, if we assume we have 2 devices connected to a network with total available bandwidth of x Mb each device will consume 0.5x Mb. Further, we can assume x Mb equates to y SVC layers so each device would stream 0.5y SVC layers. If device 1 has a SVC layer capability of y and device 2 is higher quality device of capability 2y then the device states would be:

- Device 1 (low quality) = L_A / L_C = 0.5y/y = 50%
- Device 2 (high quality) = L_A / L_C = 0.5y/2y = 25%

When we consider user expectation, the device 2 user would be unhappy with the quality of the video and his/her real QoE would be low. For instance, if user 2 was watching a movie on a HDTV set and user 1 is watching a news clip on a mobile device then the DVMs fail to achieve the desired optimum QoE for the users. In order to address this paradox the DVM must implement policy rules that reflect the user expectation in terms of content and device capability. Building on a layer selection algorithm the DVM can implement file/chunk grabbing at a device level based on a hierarchy of device capability and content quality. This could be expressed as SVC layer grabbing where the device of higher resolution, streaming content of higher quality grabs more SVC layers than a device of lower capability and/or streaming content of lower quality.

At a high level the DVM algorithm can be broken into to three distinct areas: Sensor action, policy application and actuator action:

- Sensor action
    - Establish device capability
        - Determine device resolution range
        - Determine connection type
        - Determine NIC maximum bandwidth
    - Detect content type
        - Query tags or chunk packet headers
    - Detect available bandwidth
        - Sample NIC every x seconds
    - Determine device state
        - Calculate maximum svc quality layer capability
        - Measure actual SVC layer capability
            - Sample every x seconds
- Policy application
    - Read SVC layer selection rules
        - Set content priority
        - Set device type priority
- Actuator action
    - Set bandwidth *GRAB* priority ranking
    - Stream video at maximum capability
        - Read device history
        - Set target SVC layer
        - Stop at max capability SVC layer (capping bandwidth)
        - If insufficient bandwidth to achieve max svc layer then grab maximum bandwidth available to achieve highest possible SVC layer according to selection priority
        - Update device policy knowledgebase
- Repeat sensing

3.2.2 Design of Simulated Physical Architecture

The platform for the evaluation is a simulator designed to replicate a resource contended home network with multiple users connected. The typical Irish home has at least four devices on line (Eircom, 2014). The example home network is heterogeneous with respect to content delivery protocols combining both 802.11N WIFI via access points with fixed line 1000Base-T Ethernet over Category 5e cable. The network topology is shown in Figure 3-2.

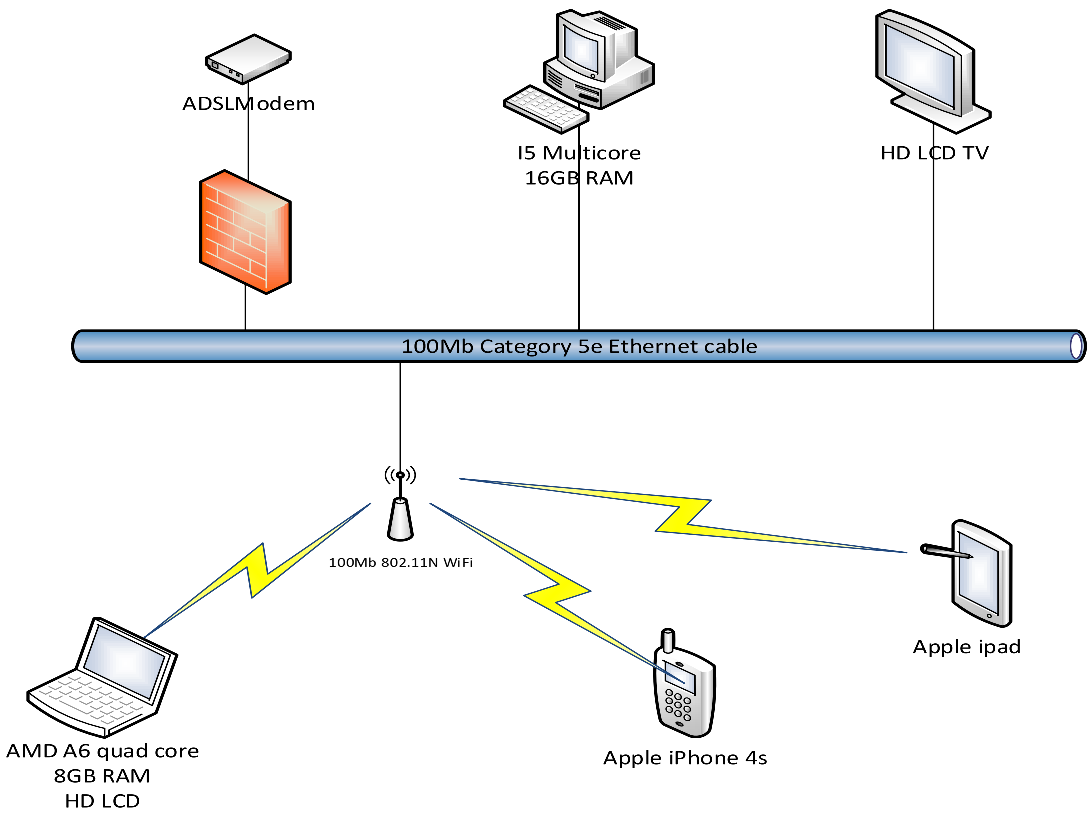

FIGURE 3-2 – NETWORK TOPOLOGY

The users can access content via different devices connected to the home network. Each user can stream video content of differing type. The simulator would be set up to represent streaming of video to devices selected from the actual devices in an example home network as shown in Table 3-2. Studies have shown that user perceived QoE is influenced by expectation based on content type (Agboma & Liotta, 2008). Three different content types were simulated, Movie, News, and Sport.

TABLE 3-2 - HOME NETWORK DEVICES

Device	Description	Access Type	Resolution	SAR	Pixels
Tablet	iPad (3rd Generation) QXGA	WIFI	2048 x 1536	4:3	3145728
HDTV	HDTV	Ethernet	1920 x 1080	16:9	2073600
desktop	PC 900p Monitor	Ethernet	1600 x 900	16:9	1440000
Laptop	720p/1080i displays Windows 8 netbooks	WIFI	1366 x 768	683:384	1049088
Smart phone	Apple iPhone	WIFI	480 x 380	3:2	153600

4 Implementation

The goal of simulation was to establish that a device could adapt QoE based on available resources and device capability. Furthermore, simulation should demonstrate that multiple devices could adapt their QoE when competing for resources. The intention to simulate SVC layer selection was based on the fact that it was possible to correlate QoE to the layer being delivered to the device using the relation developed in section 3.2.1.2. This simple measure of device state as expressed as a proportion of the device QoE capability would be the metric used to compare the effectiveness of the adaptation algorithm.

Simulation would be performed in two phases; a proof of concept using an iterative Excel model and a simulator based on a Java program. Each phase would include a baseline evaluation, where each device competes for resources on an equal basis, and an evaluation of the implementation of the DVM layer selection algorithm described in section 3.2.1.3. In this way the effectiveness of the DVM algorithm could be evaluated.

4.1 Excel Proof of Concept

This section describes a proof of concept (PoC) that was performed in order to test the feasibility of optimising perceived QoE in a home network of contended bandwidth. The PoC was designed to demonstrate the ability to measure QoE and implement a simple distributed algorithm that could enforce a policy of differential bandwidth consumption based on device type and video content hierarchies.

The PoC was based on simple model of SVC video streaming where a key assumption was that there is a linear relationship between QoE and the SVC layer being played by the receiving device. This is a safe assumption since SVC layers comprise 3 fundamental QoE parameter types, temporal (frame rate), spatial (resolution), and video quality (SNR). Each of these groups represent a measurable set of parameters related to bandwidth (temporal), device (spatial) and content (video quality). As such, it is not necessary to measure the parameters but simply infer the quality on the basis that the higher the layer, the greater the QoE.

The goal of the PoC was to demonstrate that a simple layer selection algorithm could be employed at device level allowing devices of higher resolution, that were streaming content of higher intrinsic quality, to secure a greater share of available bandwidth than those devices of lower resolution and video quality.

A Microsoft Excel model was constructed to simulate a home network of restricted bandwidth with 3 devices connected. Three DVM agents were constructed to represent each device. Figure 4-1 shows the simulated sensing part of the DVM. The device type can be selected from a dropdown list and the maximum capability of the device is calculated.

Device Capability	HDTV			
Max Resolution	Connection type	Max bandwidth (NIC) (mbs)	Max SVC Layer Index	
1440000	ethernet	100	297	
Content Sensor			**Bandwidth Sensor**	
Type (tag)	Quality (PSNR) index (1,2,3)		Total network bandwidth	Effective Bandwidth availability
Movie	3		100	24.17

FIGURE 4-1 - EXCEL DVM FROM PROOF OF CONCEPT

Max Resolution – This value is read from the table of devices when the device type is selected and is shown as the number of pixels

Connection type – This value is read from the table of devices when the device type is selected

Max Bandwidth (NIC) – This value is read from the table of devices when the device type is selected and is shown as the number of Mbs. If Ethernet is shown the maximum bandwidth available to the device is 100Mb; if WiFi is selected the maximum bandwidth available is limited to 40Mbs

Max SVC Layer - An SVC layer capability is expressed as the "Max SVC Layer" index. This is a value based on a combination of resolution, maximum bandwidth, and content type. The layer is selected from a 3D matrix of spatial, temporal and content quality values (Figure 4-2). An arbitrary value of 0.3mb was chosen to represent the packet size of a single SVC layer. The total network bandwidth is read from environment variables

Content sensor – The user can chose the content type from a drop down menu of News, Movie or Sport

Quality (PSNR) Index – is a value assigned to content type that is used to determine the SVC layer selection from the SVC matrix (Figure 4-2)

Bandwidth Sensor – shows the total network bandwidth, read from the environment variables in the 3 dimensional SVC layer table shown in Figure 4-3, and the bandwidth available to the device. The latter value is calculated based on the total network bandwidth minus the bandwidth consumed by other devices.

News	Temporal Mbps									
Spatial Pixels	**5**	**10**	**15**	**20**	**25**	**80**	**85**	**90**	**95**	**100**
999999	1	16	31	46	61	226	241	256	271	286
1999999	2	17	32	47	62	227	242	257	272	287
2999999	3	18	33	48	63	228	243	258	273	288
3999999	4	19	34	49	64	229	244	259	274	289
4999999	5	20	35	50	65	230	245	260	275	290

Sport	Temporal Mbps									
Spatial Pixels	**5**	**10**	**15**	**20**	**25**	**80**	**85**	**90**	**95**	**100**
999999	6	21	36	51	66	231	246	261	276	291
1999999	7	22	37	52	67	232	247	262	277	292
2999999	8	23	38	53	68	233	248	263	278	293
3999999	9	24	39	54	69	234	249	264	279	294
4999999	10	25	40	55	70	235	250	265	280	295

Movie	Temporal Mbps									
Spatial Pixels	**5**	**10**	**15**	**20**	**25**	**80**	**85**	**90**	**95**	**100**
999999	11	26	41	56	71	236	251	266	281	296
1999999	12	27	42	57	72	237	252	267	282	297
2999999	13	28	43	58	73	238	253	268	283	298
3999999	14	29	44	59	74	239	254	269	284	299
4999999	15	30	45	60	75	240	255	270	285	300

FIGURE 4-2 - 3D SVC LAYER MATRIX

The model is an iterative one based on the assumption that each device starts accessing content by downloading one SVC layer at time until either the device capability is reached or the bandwidth available to the device is exhausted. In this way basic SVC adaptation is modelled; each device accesses SVC layers of increasing quality until bandwidth is exhausted using a simple *instance based algorithm* where the SCV layer is obtained from look-ups from the 3 dimensional SVC table in Figure 4-2 according to three input parameters:

- Device resolution
- Available bandwidth
- Content type

The baseline algorithm is a conditional one of the general form:

Let device capability = n SVC layers

Do until actual SVC layer = n, or available bandwidth = 0

If resolution = x and if content = y then return SVC layer z

Available bandwidth = Starting bandwidth – f(z)

Device State = z/n

In the PoC f(z) = 2 x count (z)

The environment screen allows the user to set the total network bandwidth and initiate the test by setting the reset value to "0". After the test the results are cleared by setting the reset value to "1". To run a simulation three DVMs were configured with the desired device and content choices and the environment variables were set. The dashboard that represents the simulated home environment is shown in Figure 4-3.

Environment					
Total bandwidth available	150				
Number of devices connected	3				
Net bandwidth available	76.25	88%			
Reset	0				
	Device	**Connection**	**Type**	**Layer**	**% QoE**
Device 1	HDTV	ethernet	Movie	73.00	24%
Device 2	desktop	Wifi	Sport	61.00	54%
Device 3	iphone	Wifi	News	21.00	20%

FIGURE 4-3 - SIMULATED HOME NETWORK ENVIRONMENT

During the test the device state for each DVM is updated at each iteration. This is shown in Figure 4-4**Error! Reference source not found.** . Here the device state shows the effective capability expressed as the maximum SVC layer achievable, in this example layer 73 represents 24.5% of the device QoE capability. This value is calculated based on the available bandwidth and the bandwidth consumed by each layer. During a test the effective capability of the device decreases during each iteration as the available bandwidth reduces due to consumption by each of the three devices.

Device State		
Effective Capability	73.00	
Starting layer	1.00	0
Tracker	2000.00	**100**
Current layer	73.00	24.5%

FIGURE 4-4 - DVM DEVICE STATE

In order to complete the proof of concept a series of tests were run with different device/content configurations in baseline mode. The tests were then repeated applying the DVM layer selection algorithm and differences were analysed.

The DVM layer selection algorithm rules were constructed to represent a hierarchy of device type and content type. The content type is representative of native video quality:

- Tablet > HDTV > desktop > Laptop > Smart phone
- Movie > Sport > News

The algorithm is a simplification of the SVC selection algorithm proposed by Ruckert et al., (2012) in which they propose SVC layer selection based on a priori mapping of QoE capability to the SVC layers. In the PoC streaming is represented by iterative layer selection starting at layer one and sequentially increasing layers with each iteration. In order to simulate QoE based layer selection policy rules are applied where the higher QoE capability combinations select higher numbers of SVC layers with each iteration according to the rules in Table 4-1.

The DVM layer selection algorithm is a conditional one of the general form:

Let device capability = n SVC layers

Let QoE coefficient of layer selection = p

Let the device type = a

Let the content type = b

Do until actual SVC layer = n, or available bandwidth = 0

For a, b look up p and look up z

Return p.count (z)

Available bandwidth = Starting bandwidth – f(p.count (z))

Device State = (pz)/n

In the PoC f(z) = 2 p.count (z)

This simplification is a sound one: Ruckert et al., propose selection of the layer most appropriate to QoE capability in one initial streaming step which consumes bandwidth commensurate with the quality of the layer. In the POC, the iterative consumption of bandwidth based on stepwise layer selection arrives at a similar adaptation end state in terms of bandwidth consumption.

TABLE 4-1 - LAYER SELECTION RULES

Selection Rules	Number of SVC layers accessed per iteration		
Device/Content	Movie	Sport	News
HDTV	5	4	3
Desktop	4	3	2
iPad	3	2	1
Laptop	2	1	1
iPhone	1	1	1

4.1.1 PoC Null Hypothesis

At a basic level the first objective is to demonstrate that QoE can be measured at the devices and that some change can be affected based on the measured level of QoE. On this basis the simple null hypothesis may be expressed as follows:

H_0 = No measurable effect on QoE can be detected following the implementation of the network layer selection algorithms. Of course, with a simulation there is no statistical method of establishing support for the null hypothesis because each repeat test gave exactly the same result. Hence no confidence limits can be estimated for any differences demonstrated.

4.1.2 Results

To set a baseline a network with 3 devices was configured with device parameters that define the maximum resolution and bandwidth capability as shown in Table 4-2. The total network bandwidth was set to 50Mb/s. Three connected devices were chosen to limit the number of possible experimental combinations.

TABLE 4-2 - POC DEVICE PARAMETERS

Device	Access Type	Max bandwidth capability (Mb/s)	Resolution (Pixels)
iPad	WIFI	40	3145728
HDTV	Ethernet	100	2073600
Desktop	Ethernet	100	1440000
Laptop	WIFI	40	1049088
iPhone	WIFI	40	153600

4.1.2.1 Experiment 1

The objective of the experiment was to demonstrate the ability of the simulator to model the consumption of bandwidth as SVC layers are downloaded with predictable results for shared bandwidth without the application of the layer selection algorithm. The four tests shown in Table 4-3 were performed. In each case each of the three connected devices were of the same type and each device was accessing the same content.

TABLE 4-3 - POC EXPERIMENT 1: TEST MATRIX

Test	Device 1	Content1	Device 2	Content2	Device 3	Content3
1.1	iPad	Movie	iPad	Movie	iPad	Movie
1.2	Desktop	Movie	Desktop	Movie	Desktop	Movie
1.3	HDTV	Movie	HDTV	Movie	HDTV	Movie
1.4	Laptop	Movie	Laptop	Movie	Laptop	Movie

The results of experiment are shown in Figure 4-5. The device state is recorded when the available bandwidth is exhausted. This represents the proportion of the device QoE attained. It is clear that the bandwidth is shared evenly between each device as expected. The device state of higher resolution devices is lower than that of the lower resolution devices. This provides an index of user satisfaction and indicates users watching the content on a high capability device would be likely to be less satisfied than those watching the same content on a low capability device. This is a reasonable model of user QoE based on expectation; a user would expect high quality when watching a movie on a HDTV but would be satisfied with lower quality when watching on a laptop.

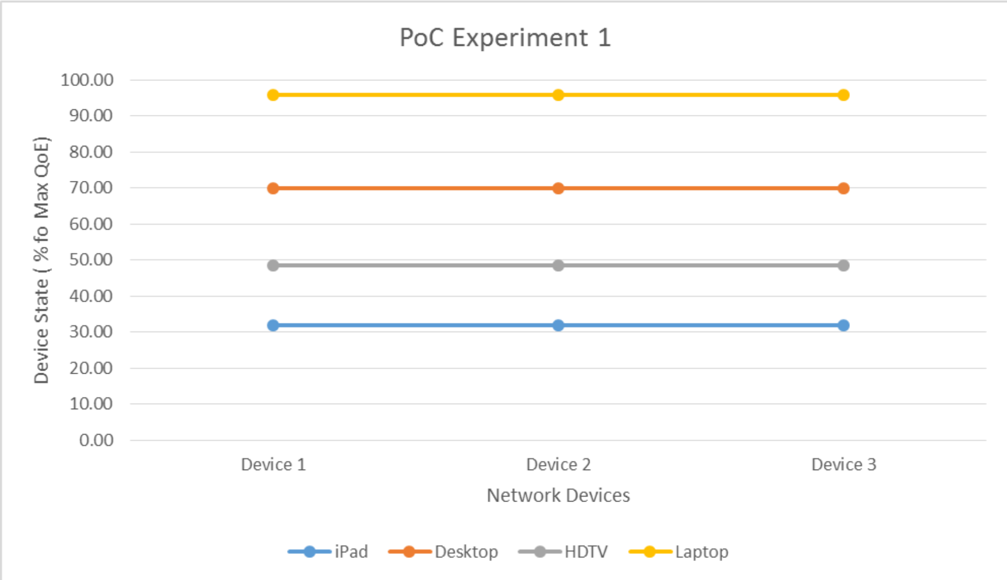

FIGURE 4-5 - POC - EXPERIMENT 1 RESULTS

4.1.2.2 Experiment 2

The objective of the experiment was to show the effect of different content on the device state without the layer selection algorithm, i.e. when each device accesses content at the same rate. The test matrix is shown in Table 4-4.

TABLE 4-4 - POC EXPERIMENT 2: DEVICE CONTENT MATRIX

Test	Device 1	Content1	Device 2	Content2	Device 3	Content3
2.1	iPad	Movie	iPad	Movie	iPad	Movie
2.2	iPad	Movie	iPad	Sport	iPad	News

The results of experiment are shown in Figure 4-6. The effect of lower quality content can be seen. It is manifest as much higher QoE% because the maximum capability for the device/content combination is much lower. So without the application of the layer selection algorithm the user watching the higher quality content is likely to be dissatisfied. Paradoxically the user watching low quality content will be likely to be satisfied. However, it is possible that such users would be equally satisfied with lower QoE% due to lower expectations.

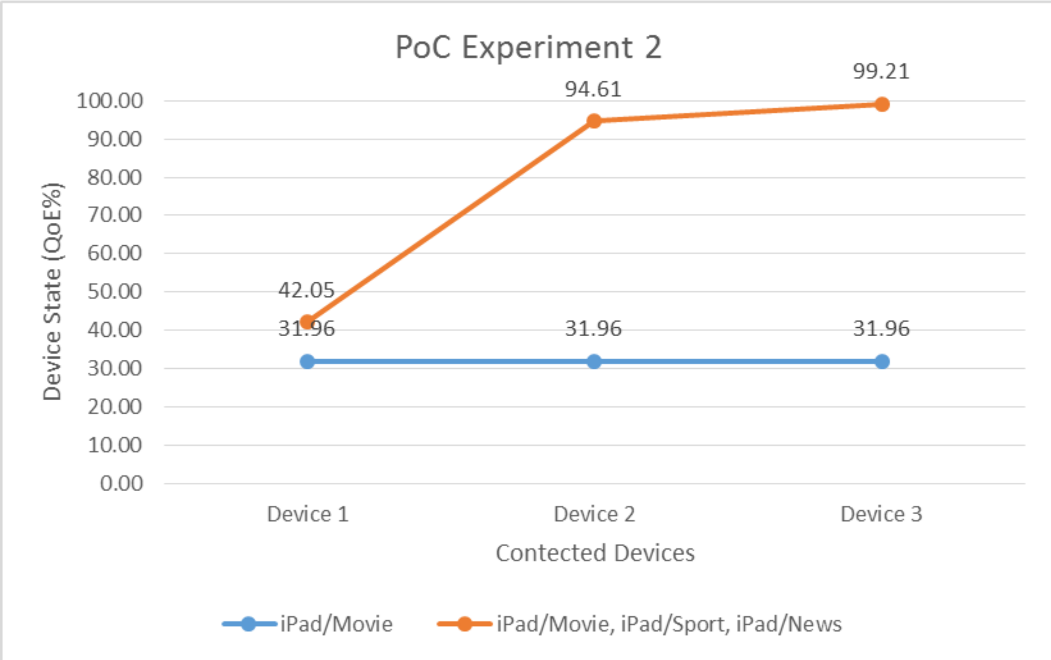

FIGURE 4-6 - POC - EXPERIMENT 2 RESULTS

4.1.2.3 Experiment 3

The objective of the experiment was to demonstrate the effect of the application of a simple layer selection algorithm to facilitate favourable SVC layer grabbing by higher capability devices accessing higher quality content.

- iPad > HDTV > desktop > Laptop > iPhone
- Movie > Sport > News

For the PoC the algorithm was modelled by each device/content combination grabbing different numbers of SVC layers per iteration as shown in Figure 4-7

Selection Rules			
	Movie	Sport	News
HDTV	5	4	3
Desktop	4	3	2
Ipad	3	2	1
Laptop	2	1	1
iPhone	1	1	1

FIGURE 4-7 - LAYER GRABBING SELECTION RULES

The device/content combinations shown in Table 4-5 were selected to demonstrate the change is device state due to the application of the layer selection algorithm selection rules.

The goal of the function is to improve the device state for the higher quality device/content combinations.

TABLE 4-5 - POC EXPERIMENT 3: DEVICE CONTENT MATRIX

Test	Layer selection algorithm	Device1	Content1	Device2	Content2	Device3	Content3
3.1	No	iPad	Movie	HDTV	Movie	Laptop	Movie
3.2	Yes	iPad	Movie	HDTV	Movie	Laptop	Movie
3.3	No	iPad	Movie	Desktop	Sport	HDTV	News
3.4	Yes	iPad	Movie	Desktop	Sport	HDTV	News

The results of experiment are shown in Figure 4-8. The result is a significant improvement in the iPad device state at the expense of the other two devices. This coarse adaptation across the three devices could be improved by application of learning; if a reasonable measure of user satisfaction was the length of time of a single streaming session of a content type (assuming users would stop watching the poorer QoE video sooner) then the size of the layer selection parameters could be refined to further optimise the device state across the connected devices.

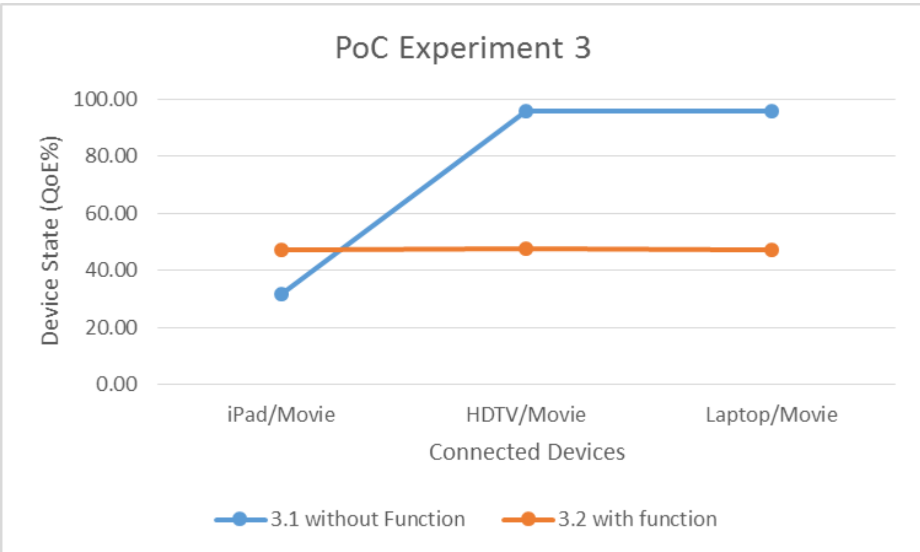

FIGURE 4-8 - POC EXPERIMENT 3 - CONSTANT CONTENT

The results shown in Figure 4-9 clearly demonstrate the effect of the weighting applied to content type. Without the Layer selection algorithm the maximum capability of both the desktop and HDTV is reached which allows the iPad to grab the remaining available bandwidth and achieve a device state of over 60%. Consequently, the improvement to the iPad device state on application of the function is relatively modest

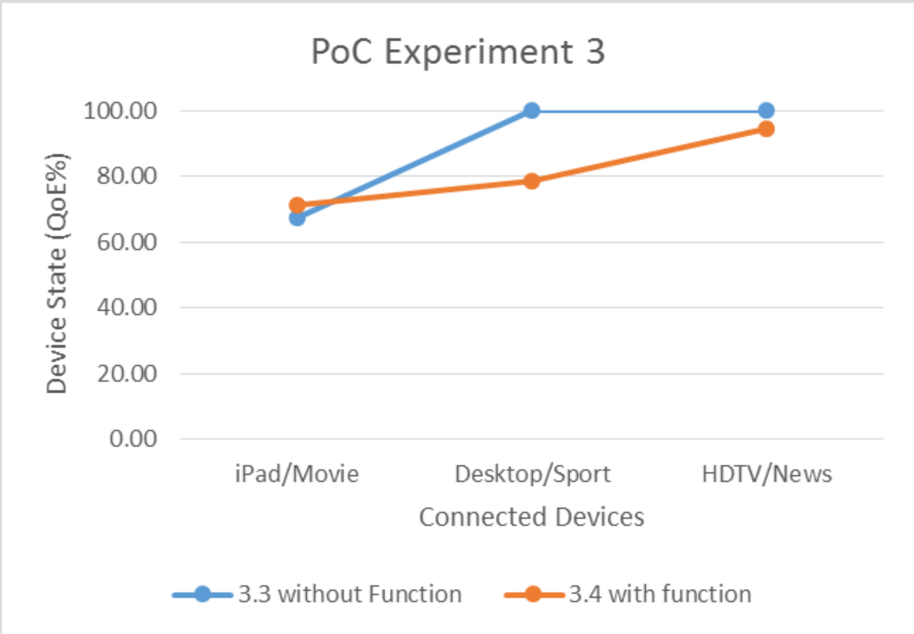

FIGURE 4-9 - POC - EXPERIMENT 3 – MULTI DEVICE/CONTENT COMBINATIONS

4.1.3 Conclusions

The proof of concept was successful in as far as it was possible to demonstrate the tangible effect of the application of an SVC layer selection rules based algorithm. The device state provides an index of user satisfaction. Improvements in the device state of the high quality device/content combinations come at the expense of the other two devices. Without the application of the layer selection algorithm the user watching the higher quality content is likely to be dissatisfied. Paradoxically the user watching low quality content will be likely to be satisfied. However, it is possible that such users would be equally satisfied with lower QoE% due to lower expectations.

4.2 Java Simulator

Having satisfied the basic goal of the PoC; to demonstrate an appreciable change due to the application of the Layer selection algorithm, the next step was to create a simulator that could be configured with sensing parameters that would map to parameters that would be measured by an operational device manger agent:

Sensing Parameters

- Available network bandwidth
- Connection type
- Device type - from which device capability is derived
- Content type – from which content quality is derived

The simulator was coded in Java using the NetBeans IDE version 8.01. The Java development kit was jdk1.7.0_67 with Java Runtime Environment JRE 7. Java was chosen as the coding language because Java is device environment independent, due to the fact that code is executed in the Java Virtual Machine, JVM. In the event that the simulation proved encouraging, the code could be deployed in home network devices with maximum code re-use. The generalised class model for the simulator is shown in Figure 4-10.

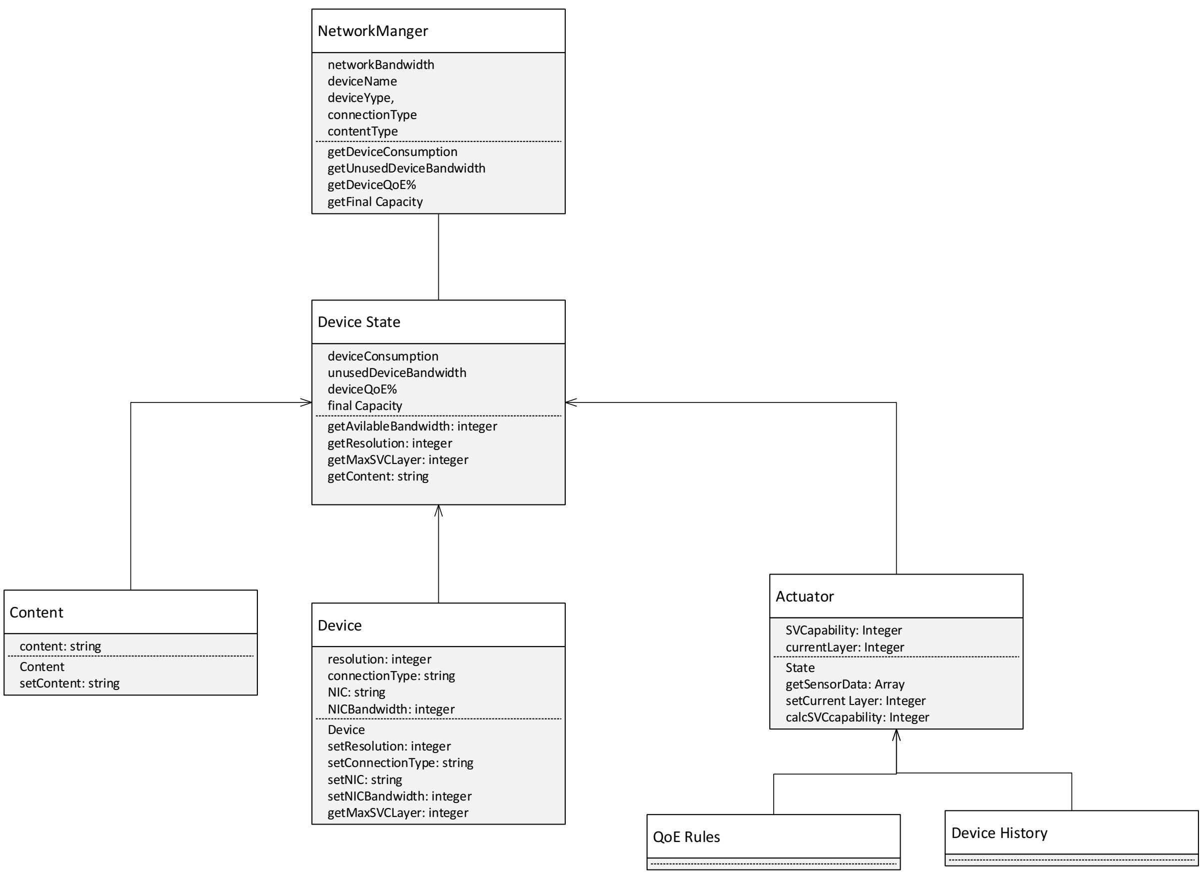

FIGURE 4-10 - SIMULATOR JAVA CLASS MODEL

Network and device parameters were entered via a text editor. The network manager was designed to configure as many devices as possible. However, the number of devices per experiment was limited to three to minimise the experimental permutations. The experiment configuration dialogue was as follows:

```
run:
Enter network capacity (total bandwidth for streaming)
50
Network Capacity is 50.0Mbps
************************************************
Enter Device name
My TV
Enter Device type,
HDTV
Enter connection Type, 'WiFi' or 'Ethernet'
Ethernet
Enter Content Type, 'Movie', 'Sport' or 'News'
Movie
```

After the device is configured the following calls are made to the device object:

```
DeviceMapPolicy device1 = new DeviceMapPolicy();
    device1.mapDevice(content, deviceType, connection);
    System.out.println("Device 1 NIC bandwidth is " + device1.getMaxBandwidth());
    System.out.println("Device 1 is playing " + device1.getContent() + " with PSNR Index " + device1.getPSNRIndex());
    System.out.println("Device 1 Resolution is " + device1.getResolution());
    System.out.println("Device 1 Max SVC Layer is " + device1.getLayer());
    System.out.println("Device 1 Effective device bandwidth is " + Math.min(device1.getMaxEffectiveBandwidth(),device1.getMaxBandwidth()));
    System.out.println("Device 1 is playing " + device1.getContent());
```

A confirmation message is generated thus:

```
Device 1 is My TV
Device 1 is a(n) HDTV
```

Device 1 NIC bandwidth is 100.0

Device 1 is playing Movie with PSNR Index 3.0

Device 1 Resolution is 2073600.0

Device 1 Max SVC Layer is 139.968

Device 1 Effective device bandwidth is 34.992

Device 1 is playing Movie

After configuring 3 devices the program can be run. A starting value of 1 is entered and the simulator iterates through each DVM simultaneously until bandwidth is exhausted:

```
double remainingCapacity = capacity;

        int layer = in.nextInt();

        while(remainingCapacity>0 && layer <500){

device1.BandwidthConsumption(device1.getContent(),device1.getResolution(),Math.min(d
evice1.getMaxEffectiveBandwidth(),device1.getMaxBandwidth()),device1.getLayer(),layer)
;

device2.BandwidthConsumption(device2.getContent(),device2.getResolution(),Math.min(d
evice2.getMaxEffectiveBandwidth(),device2.getMaxBandwidth()),device2.getLayer(),layer)
;

device3.BandwidthConsumption(device3.getContent(),device3.getResolution(),Math.min(d
evice3.getMaxEffectiveBandwidth(),device3.getMaxBandwidth()),device3.getLayer(),layer)
;

        //remainingCapacity = capacity - (device1.getConsumption()+
device2.getConsumption());

        remainingCapacity = networkCapacity(capacity,(device1.getConsumption()+
device2.getConsumption()+ device3.getConsumption()));

        layer ++;

        }

        // Get QoE as percent of potential layers for a device object

        System.out.println("Device 1 consumption is "  + device1.getConsumption());

         System.out.println("Device 1 Unused device Bandwidth is "  +
(device1.getMaxBandwidth() - device1.getConsumption()));

         //System.out.println("Device 1 Actual SVC Layer is " + device1.getActualLayer());
```

```
//layer++;
System.out.println("Device 1 QoE % " + device1.getQoE());
  // Get QoE as percent of potential layers for a device object
System.out.println("Device 2 consumption is "  + device2.getConsumption());
System.out.println("Device 2 Unused device Bandwidth is "  +
(device2.getMaxBandwidth() - device2.getConsumption()));
//System.out.println("Device 2 Actual SVC Layer is " + device2.getActualLayer());
System.out.println("Device 2 QoE % " + device2.getQoE());
  System.out.println("Device 3 consumption is "  + device3.getConsumption());
System.out.println("Device 3 Unused device Bandwidth is "  +
(device3.getMaxBandwidth() - device3.getConsumption()));
//System.out.println("Device 2 Actual SVC Layer is " + device2.getActualLayer());
System.out.println("Device 3 QoE % " + device3.getQoE());
System.out.println("Final Capacity " + remainingCapacity);
```

The results are displayed as follows:

```
Device 1 consumption is 22.5
Device 1 Unused device Bandwidth is 17.5
Device 1 QoE % 42.38
Device 2 consumption is 22.5
Device 2 Unused device Bandwidth is 17.5
Device 2 QoE % 42.38
Device 3 consumption is 15.0
Device 3 Unused device Bandwidth is 25.0
Device 3 QoE % 63.57
Final Capacity -1.03
BUILD SUCCESSFUL (total time: 53 seconds)
```

For each device the device state (*QoE%*), the bandwidth consumed and the unused device bandwidth is shown. In addition the final network capacity, expressed as unused bandwidth, is shown. It should be noted that this is sometimes negative. This is because each SVC layer consumes a fixed bandwidth and the program terminates when the final capacity falls to zero or below. In order to compare results with and without application of the algorithm the results are normalised to a concluding value of zero remaining bandwidth

A number of experiments were conducted as paired configurations, i.e. three device content configurations were repeated with and without the layer selection algorithm employed. The pairings were grouped into experiments to determine the changes in QoE% due to the Layer selection algorithm with the following focus:

- Change in lowest quality device regardless of content
- Change in high quality device with highest quality content
- Change in in device state based on content type

Where those devices achieved 100% QoE the results were ignored as the change was capped by device capability and would give a false estimation of the change.

4.2.1 Results

4.2.1.1 Experiment 1

The first experiment was designed as a control. Each device content combination was identical before and after applying the function as shown in Table 4-6.

TABLE 4-6 - EXPERIMENT1 TEST MATRIX

Device 1	Content1	Device 2	Content2	Device 3	Content3	Layer selection algorithm
Desktop	Movie	Desktop	Movie	Desktop	Movie	-
Desktop	Movie	Desktop	Movie	Desktop	Movie	+
HDTV	Movie	HDTV	Movie	HDTV	Movie	-
HDTV	Movie	HDTV	Movie	HDTV	Movie	+
iPad	Movie	iPad	Movie	iPad	Movie	-
iPad	Movie	iPad	Movie	iPad	Movie	+

Device state change in high quality device regardless of content = -0.883

Device state change in lowest quality device regardless of content = -0.883

Device state change in high quality device with highest quality content = -0.883

No change in device state should be expected after application of the layer selection algorithm. The fact that there is a change is a result of the method of calculation of the SVC layer at each iteration when applying using the layer selection algorithm. For all other experiments these values were subtracted from the changes measured.

4.2.1.2 Experiment 2

The second experiment, shown in Table 4-7, was designed to show the effect of the Layer selection algorithm on the device states of a network with two high quality devices and one lower quality device all access the same content type:

TABLE 4-7 - EXPERIMENT 2 TEST MATRIX

Device 1	Content1	Device 2	Content2	Device 3	Content3	Layer selection algorithm
iPad	Movie	iPad	Movie	iPhone	Movie	-
iPad	Movie	iPad	Movie	iPhone	Movie	+
iPad	Movie	iPad	Movie	Laptop	Movie	-
iPad	Movie	iPad	Movie	Laptop	Movie	+
iPad	Movie	iPad	Movie	HDTV	Movie	-
iPad	Movie	iPad	Movie	HDTV	Movie	+
iPad	Movie	iPad	Movie	Desktop	Movie	-
iPad	Movie	iPad	Movie	Desktop	Movie	+

Change in lowest quality device regardless of content = -29.990 - (-0.833) = - 29.157

Change in high quality device with highest quality content = 5.027 – (-0.833) = 5.860

There was a positive change in the device state for the highest quality device of 5.027 QoE%. The null hypothesis was that there was no significant change due to the application of the Layer selection algorithm

H_0 *: mean difference = 0*

H_A *: mean difference > 0*

Using the paired test with 3 degrees of freedom for the highest quality device:

t = 2.417

t from table = 4.541 at 99% confidence limit

Similarly for the lowest quality device:

t = 2.768

t from table = 4.541 at 99% confidence limit

On this basis there is no reason so reject H_0 and therefore the layer selection algorithm has no real effect. This may be explained mainly by the small sample size, the fact that all devices access the same content, and that device 1 and 2 "share" the increased bandwidth provided by the action of the Layer selection algorithm.

4.2.1.3 Experiment 3

The third experiment was designed to show the effect of the Layer selection algorithm on the device states of a network with one high quality device/content combination, one high quality device with a mix of content, and a lower quality with a mix of content. The experiment design is given in Table 4-8.

TABLE 4-8 - EXPERIMENT 3 TEST MATRIX

Device 1	Content1	Device 2	Content2	Device 3	Content3	Layer selection algorithm
iPad	Movie	iPad	Sport	HDTV	News	-
iPad	Movie	iPad	Sport	HDTV	News	+
iPad	Movie	iPad	Sport	iPhone	News	-
iPad	Movie	iPad	Sport	iPhone	News	+
iPad	Movie	iPad	Sport	Laptop	News	-
iPad	Movie	iPad	Sport	Laptop	News	+
iPad	Movie	iPad	Sport	Desktop	News	-
iPad	Movie	iPad	Sport	Desktop	News	+
iPad	Movie	iPad	Movie	Desktop	Sport	-
iPad	Movie	iPad	Movie	Desktop	Sport	+
iPad	Movie	iPad	Sport	iPhone	Movie	-
iPad	Movie	iPad	Sport	iPhone	Movie	+
iPad	Movie	iPad	News	iPhone	Sport	-
iPad	Movie	iPad	News	iPhone	Sport	+
iPad	Movie	iPad	Sport	Desktop	Movie	-
iPad	Movie	iPad	Sport	Desktop	Movie	+
iPad	Movie	iPad	Sport	Laptop	Movie	-
iPad	Movie	iPad	Sport	Laptop	Movie	+
iPad	Movie	iPad	News	Laptop	Sport	-

Device 1	Content1	Device 2	Content2	Device 3	Content3	Layer selection algorithm
iPad	Movie	iPad	News	Laptop	Sport	+
iPad	Movie	iPad	News	Desktop	Sport	-
iPad	Movie	iPad	News	Desktop	Sport	+
iPad	Movie	iPad	Sport	HDTV	Movie	-
iPad	Movie	iPad	Sport	HDTV	Movie	+
iPad	Movie	iPad	News	HDTV	Sport	-
iPad	Movie	iPad	News	HDTV	Sport	+

Change in lowest quality device regardless of content = -12.652-(-0.833) = -11.819

Change in high quality device with highest quality content = 8.614-(-0.833) = 9.473

There was a positive change in the device state for the highest quality device of 9.473 QoE%. The null hypothesis was that there was no significant change due to the application of the layer selection algorithm

H_0 : mean difference = 0
H_A : mean difference > 0

Using the paired test with 12 degrees of freedom for the highest quality device:

t = 3.711

t from table = 2.681 at 99% confidence limit

Similarly for the lowest quality device:

t = 2.649

t from table = 2.681 at 99% confidence limit

On this basis there is evidence of the positive effect of the layer selection algorithm on the QoE for the high quality device/content combination. There is weaker evidence for the reduction in QoE for the lowest quality device. This may be explained by the mix of quality of the content for this device; the better content will consume more bandwidth, thus the average change will be reduced.

4.2.1.4 Experiment 4

The fourth experiment was designed to show the effect of the layer selection algorithm on the device states of a network with one high quality device/content combination and a mix of other lower quality device and content types. The test matrix for this experiment contained 46 pairs and can be summarised as shown in Table 4-9.

TABLE 4-9 - EXPERIMENT 4 TEST MATRIX

Device 1	Content1	Device 2	Content2	Device 3	Content3	Layer selection algorithm
iPad	Movie	All combinations except iPad	All combinations	All combinations except iPad	All combinations	-
iPad	Movie	All combinations except iPad	All combinations	All combinations except iPad	All combinations	+

Change in lowest quality device regardless of content = -21.787-(-0.833) = -20.994

Change in high quality device with highest quality content = 11.559-(-0.833) = 12.392

There was a positive change in the device state for the highest quality device of 12.392 QoE%. The null hypothesis was that there was no significant change due to the application of the layer selection algorithm

H_0 : mean difference = 0
H_A : mean difference > 0

Using the paired test with 45 degrees of freedom for the highest quality device:

t = 9.561

t from table = 2.423 at 99% confidence limit

Similarly for the lowest quality device:

t = 8.850

t from table = 2.423 at 99% confidence limit

On this basis there is a strong positive effect of the Layer selection algorithm on the QoE for the high quality device/content combination, in this case the iPad with Movie. This demonstrates the strong bias of the algorithm towards the high quality device type. Equally, there is a strong bias against the lower quality device types regardless of the content being played

4.2.1.5 Experiment 5

The fifth experiment was designed to show the effect of the layer selection algorithm on the content type. Tests were made with 3 devices of the same type each with different content. The test matrix for this experiment is shown in Table 4-10.

TABLE 4-10 - EXPERIMENT 5 TEST MATRIX

Device 1	Content1	Device 2	Content2	Device 3	Content3	Layer selection algorithm
iPad	Movie	iPad	Sport	iPad	News	-
iPad	Movie	iPad	Sport	iPad	News	+
HDTV	Movie	HDTV	Sport	HDTV	News	-
HDTV	Movie	HDTV	Sport	HDTV	News	+
Desktop	Movie	Desktop	Sport	Desktop	News	-
Desktop	Movie	Desktop	Sport	Desktop	News	+
Laptop	Movie	Laptop	Sport	Laptop	News	-
Laptop	Movie	Laptop	Sport	Laptop	News	+
iPhone	Movie	iPhone	Sport	iPhone	News	-
iPhone	Movie	iPhone	Sport	iPhone	News	+

Change in high quality content = 7.900 - (-0.833) = 8.733

Change in lowest content = -18.494- (-0.833) = 19.327

Change in mid quality content = -6.794 - (-0.833) = 5.961

The null hypothesis was that there was no significant change due to the application of the Layer selection algorithm

H_0 : *mean difference = 0*

H_A : *mean difference > 0*

Using the paired test with 4 degrees of freedom for the highest quality content:

t = 19.42

t from table = 3·747 at 99% confidence limit

Similarly for the mid quality content:

t = 3.947

t from table = 3·747 at 99% confidence limit

Similarly for the low quality content:

t = 23.21

t from table = 3·747 at 99% confidence limit

On this basis there is a strong positive effect of the layer selection algorithm on the QoE for the high quality content. Equally, there is a strong negative effect for the lowest quality content. There is also a significant negative effect on the mid quality content. This demonstrates the strong bias of the algorithm towards the high quality content type at the expense of QoE for other content types.

4.2.2 Conclusions

The experiment results demonstrate the ability of the DVM to simulate the effect of the layer selection algorithm on QoE for devices accessing SVC video content. The DVM improves the QoE for high quality device and content combinations by reducing the QoE for lower quality combinations. There is a small error when the layer selection algorithm is applied. The error was quantified as 0.88% of the maximum achievable QoE for a device. An offset correction could be applied to all simulations to compensate for the error.

When simulating a network with two highest quality device/content combinations the layer selection algorithm had no statistically significant effect. This was due, in part, to the small sample size. Also, the fact that all devices accessed the same content means that there was no change affected due to content type. More importantly, two highest quality device/content effectively “share” the increased bandwidth provided to them by the action of the layer selection algorithm. Thus the potential improvement in QoE is divided between the two devices.

When the network was configured with just two high quality devices, only one of which accessed the highest quality content, the effect of the layer selection algorithm was clearly demonstrated. There was a positive effect on the QoE for the high quality device/content combination and a reduction in QoE for the lowest quality device. Although the scale of the reduction in QoE% was less significant because, for some combinations, high quality content compensated for the low quality device.

When just one highest quality device content combination was examined in networks with all other combinations. The effect of the layer selection algorithm was most dramatic with a 12% improvement of the high quality device state and a 20% reduction in the lowest quality device state.

There is a strong positive effect of the layer selection algorithm on the QoE for the high quality content. Equally, there is a strong negative effect for the lowest quality content. There is also a significant negative effect on the mid quality content. This demonstrates the strong bias of the algorithm towards the high quality content type at the expense of QoE for other content types

5 Conclusions

The research question was *"to what extent can cognitive algorithms support the optimisation of Quality of Experience and bandwidth utilisation for users when downloading or streaming video to connected devices in a resource contended network?"* The question was prompted by personal experience of poor QoE when accessing video content in a home network with many connected devices. Despite the fact that access network bandwidth reached a level that could sustain real time streaming of HD video content in about 2010/2011, advances in device technology have seen the advent of high definition display capability that puts more demand on the network capacity due to the need to stream ever larger files.

The first challenge in this study was to understand the various video streaming delivery mechanisms and discover what scope there was for influencing QoE. It became apparent that the modern delivery methods such as DASH and SVC were designed with quality adaptation in mind. However, the quality in question for these protocols is limited to picture and sound quality and not the entire user experience. In considering QoE other factors must be taken into account, such as loading rate, transitions between picture quality states, buffer run out, pixilation and so on. Such factors are a function not of the video file or the delivery mechanism, but of the environment; specifically the network state.

Thus the second challenge was to determine if QoE could be estimated from indicative parameters based on the video file, the delivery mechanism, and the network state. To meet this challenge it was necessary to investigate methods of estimating QoE. It was clear that QoE is subjective assessment of a user's perception of the experience in question. Fortunately, much work had been done in establishing objective measures of QoE for video streaming. Many researchers have demonstrated that subjective QoE assessment could be mapped to objective parameters with some degree of accuracy.

So, the question became focused on whether QoE could be used to optimise content delivery in near real-time. The adaptive steaming protocols such as SVC showed that real-time adaptation of video steam quality to changes in bandwidth is possible. Therefore, it should be possible to make decisions about SVC chunk and layer selection based on QoE related parameters also in near real-time. During the study it became apparent that a combination of two types of algorithm would be required to optimise QoE. The first would be an adaptation algorithm that would be capable of measuring key parameters and affecting some change on the video stream that influences QoE. The second would be a cognitive algorithm that was capable of changing the adaptation algorithm on the basis of learning from previous adaptations.

Researchers have also demonstrated that algorithms can be used to adapt QoE during playback. Many of these algorithms are based on adapting the download rate to the available bandwidth by selecting chunks with video quality appropriate to the bandwidth. Using bit rate, packet layer and codec information it is possible to produce objective models of QoE. Thus if a device can be made "aware" of this information it is possible to adapt playback to optimise QoE.

The proof of concept and simulation were successful in as far as it was possible to demonstrate the tangible effect of the application of an SVC layer selection rules based algorithm. The device state provides an index of user satisfaction. The experiment results demonstrate the ability of the DVM to simulate the effect of the layer selection algorithm on QoE for devices accessing SVC video content. The DVM improves the QoE for high quality device and content combinations by reducing the QoE for lower quality combinations.

Thus, adaptation based on QoE is possible based on the application of policy based algorithms at the device level. Research suggests that learning algorithms could be developed to adapt the policy algorithm in order to optimise QoE for the device users. Both Monte Carlo methods and Q Learning are ideal for network control and management. In network environments that have distributed or hybrid agent structure the discrete states of the network nodes and environment make Q Learning potentially the most useful algorithm.

5.1 Future Work

The simulator exhibits relatively coarse adaptation across the connected devices. This could be improved by application of learning algorithms such as Q learning. For example, if a reasonable measure of user satisfaction was the length of time of a single streaming session of a content type (assuming users would stop watching the poorer QoE video sooner) then the size of the layer selection parameters could be refined to further optimise the device state across the connected devices. Furthermore it would be valuable to examine the effect of sudden changes to available bandwidth; using learning to determine responses based on accumulated device knowledge could improve the efficiency of layer selection.

The next step would be to produce a home network testbed with real time parameter sensing. SVC video streams would be set up such that QoE% could be calculated. Methods of determining available bandwidth and device parameters would have to be established. QoE data for the content type could be derived from packet data. Attempts should then be made to repeat the simulation experiments and compare the calculated QoE% to the simulation results. It would be valuable to then repeat the test bed experiments with subjective QoE assessment and map MOS to calculated QoE%.

Researchers studying content delivery, network management, and quality of experience could find this work useful when considering the challenges of balancing QoE with limited network resources. The content could be any file types and the network could be any network. The work could be of particular interest to those managing enterprise networks with many connected devices accessing many types of content.

5.2 Closing Remarks

This work demonstrates the potential to optimise QoE for users accessing video content in a resource contented network. Device awareness, in terms of content and capability, is fundamental to the successful application of the algorithms. When considering the Internet of Things (IOT) it is easy to imagine a modern home with tens of connected devices all competing for bandwidth. It should be possible to make all these devices self-aware with respect to their function, environment and capability. On that basis it should be possible to implement distributed agents that facilitate cross-network optimisation of functionality and resource consumption with low computational overhead. In this scenario, when any device is connected to a network it could *know* what it is, what it does, and *learn* where it fits in the hierarchy of demand for resources. Thus, in the world of IOT, artificial intelligence could increase utility at minimum resource cost.

References

3rd Generation Partnership Project, 2015. *3GPP.* [Online]
Available at: http://www.3gpp.org/
[Accessed 28th June 2015].

Agboma, F. & Liotta, A., 2008. *QoE-aware QoS management.* New York, ACM, pp. 111-116.

Alreshoodi, M. & Woods, J., 2013. Survey on QoE\QoS Correlation Models For Multimedia Services. *International Journal of Distributed and Parallel Systems,* 4(3).

Asghar, J., Le Faucheur, F. & Hood, I., 2009. Preserving Video Quality in IPTV Networks. *IEEE Transactions on Broadcasting,* 55(2), pp. 386-395.

Barto, A. G. S. R. S. a. A. C. W., 1983. Neuronlike elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics,* Volume 13, pp. 835-846.

Behl, I. S., Saini, A. & Verma, J., 2013. Hopfiled Networks. *International Journal Of Scientific Research And Education,* 1(7), pp. 154-162.

Berk, R. A., 2008. Classification and Regression Trees. In: R. A. Berk, ed. *Statistical Learning from a Regression Perspective.* s.l.:Springer, pp. 1-65.

Boutaba, R. & Xiao, J., 2009. An Introduction to Network and Servive Management. In: M. H. Sherif, ed. *Handbook of Enterprise Integration.* Boca Raton, Florida: Talor and Francis, pp. 159-181.

Breiman, L., 2001. Random Forests. *machine Learning,* 45(1), pp. 5-32.

Broomhead, D. H. & Lowe, D., 1988. Multivariable Functional Interpolation and Adaptive Networks. *Complex Systems,* 2(3), pp. 321-355.

Carlsson, B. & Gustavsson, R., 2001. *The Rise and Fall of Napster - An Evolutionary Approach.* Hong Kong, Springer-Verlag Berlin Heidelberg, pp. 347-354.

Ceglie, C., Piro, G., Striccoli, D. & Camarda, P., 2014. Performance evaluation of 3D video streaming services in LTE-Advanced networks.. *Wireless Networks,* 20(8), pp. 2255-2273.

Celtic Plus, 2014. *Cognitive network management under uncertainty.* [Online]
Available at: http://projects.celtic-initiative.org/commune/index.html
[Accessed 24 June 2015].

Chris Gaskett, D. W. a. A. Z., 2005. *Q Learninmg in Continuous State and Action SpecesCanberra,* s.l.: Australian national University.

Cleveland, W. S. & Devlin, S. J., 1988. Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting. *Journal of the American Statistical Association,* 83(403), pp. 596-610.

Cormen, T., Leiserson, C. & Rivest, R. S. C., 2009. *Introduction to Algorithms.* 3rd ed. s.l.:The MIT Press.

Couto da Silva, A., Irisa, R. & Rodriguez-Bocca, P. R. G., 2008. *Optimal Quality-of-Experience Design for a P2P Multi-Source Video Streaming.* Beijing, IEEE.

Deng, L. & Yu, D., 2014. Deep Learning: Methods and Applications. *Foundations and Trends in Signal Processing,* 7(3-4), pp. 197-387.

Denning, R., 2012. *Applied R&M Manual for Defence Systems.* s.l.:Ministry of Defence.

Dolby, 2014. *Dolby Digital 5.1.* [Online]
Available at: http://www.dolby.com/us/en/technologies/dolby-digital.html
[Accessed 5 june 2014].

DSH-IF, 2015. *Implementation Guidelines.* [Online]
Available at: http://dashif.org/guidelines/
[Accessed 15 May 2015].

DVB Standatds Organisation, 2014. *Digital Video Broadcasting Standards.* [Online]
Available at:
http://www.etsi.org/deliver/etsi_en/302300_302399/30230701/01.04.01_60/en_30230701v010401p.pdf
[Accessed 2 January 2015].

EBSCO Industries, 2014. *EBSCO Host.* [Online]
Available at: http://web.ebscohost.com.elib.tcd.ie/ehost/selectdb?sid=a755d31d-6879-

49ae-988a-7f83e2d27d58%40sessionmgr11&vid=1&hid=15
[Accessed 18th December 2014].

Eircom, 2014. *Ireland - A Tech Savvy Nation that Needs to be Constantly Connected.* [Online]
Available at: http://pressroom.eircom.net/press_releases/article/ireland_-_a_tech_savvy_nation_that_needs_to_be_constantly_connected/
[Accessed 25 July 2015].

European Space Agency, 2014. *2014 conference on Big Data from Space (BiDS'14).* Frsacatti, Italy, Eurpoean Union.

Forbes, 2015. *Netflix Soars On Subscriber Growth.* [Online]
Available at: http://www.forbes.com/sites/laurengensler/2015/01/20/netflix-soars-on-subscriber-growth/
[Accessed 23 February 2015].

Friedman, J. H., 1991. Multivariate Adaptive Regression Splines. *Annals of Statistics,* 19(1), pp. 1-67.

Friedman, J. H., 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis,* 38(4), pp. 367-378.

Fu, T. Z. J., Chiu, D. M. & Lei, Z., 2010. *Designing QoE Experiments to Evaluate Peer-to-Peer.* Huangshan, International Society for Optical Engineering.

Ganek, A. G. & Corbi, T. A., 2003. The Dawning of the Autonomic Computer Era. *IBM Systems Journal,* 42(1), pp. 5-18.

Gantenbein, R. & Robinson, B., 2008. Decodin CODECs. *Journal of Telemedicine & Telecare,* 14(2), pp. 59-61.

Gaskett, C., Wettergreen, D. & Zelinsky, A., 2005. *Q Learninmg in Continuous State and Action SpecesCanberra,* s.l.: Australian national University.

Gentle, J. E., 2003. Monte Carlo Methods. In: Chambers, et al. eds. *Random Number Generation and Monte Carlo Methods.* New York: Springer, pp. 229-281.

Gitman, Y. et al., 2015. Delivering Enhanced 3D Video. *Intel Technology Journal,* 19(1), pp. 162-200.

Guo, G. et al., 2003. *KNN Model-Based Approach in Classification.* Catania, Scicily, Springer (Beelrlin, Heidelberg).

Harren, M. et al., 2002. *Complex Queries in DHT-based Peer-to-Peer Networks.* Cambridge Massechussetts, Springer-Verlag.

Hearst, M. et al., 2002. Support Vector Machine. *IEEE Intelligent Systems and their Applications,* 13(4), pp. 18-28.

Hoerla, A. E. & Kennard, R. W., 1970. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics,* 12(1), pp. 55-67.

Hoßfeld, T. et al., 2015. Identifying QoE optimal adaptation of HTTP adaptive streaming based on subjective studies.. *Computer Networks,* Volume 81, pp. 320-332.

Huysegems, R., De Vleeschauwer, B., Wu, T. & Van Leekwijck, W., 2012. SVC-Based HTTP Adaptive Streaming.. *Bell Labs Technical Journal ,* 16(4), pp. 25-41.

IEEE Standards Authorty, 2012. *IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* s.l.:IEEE.

Jacobs, M. & Probell, J., 2007. A Brief History of Video Coding. *ARC International..*

Joskowicz, J. & Ardao, J. C. L., 2010. A parametric model for perceptual video quality estimation. *Telecommunications Systems,* 49(1), pp. 49-62.

Kephart, J. O. & Chess, D. M., 2003. The vision of autonomic computing. *Computer,* 36(1), pp. 41-50.

Kohonen, T., 1995. Learning Vector Quantization . In: T. Kohonen, ed. *Self-Organizing Maps.* s.l.:Springer, pp. 175-189.

Kuklinski, S., 2012. *Comparative analysis of different autonomic management architectures.* Krakow, ITCP.

Laghari, K. & Connelly, K., 2012. Toward total quality of experience: AQoE model in communication ecosystems. *IEEE Communications Magazine,* 50(4), pp. 58-65.

Liao, N. & Chen, Z., 2011. A packet-layer video quality assessment model with spatiotemporal complexity estimation. *Journal on Image and Video Processing,* Volume 5.

Li, M. & Lee, C.-Y., 2015. A cost-effective and real-time QoE evaluation method for multimedia streaming services.. *Telecommunication Systems,* 59(3), pp. 317-327.

Lindeberg, M., Kristiansen, S., Plagemann, T. & Goebel, V., 2011. Challenges and techniques for video streaming over mobile ad hoc networks.. *Multimedia Systems,* 17(1), pp. 51-82.

Ljubojevic, M., Vasković, V., Babić, Z. & Starčević, D., 2014. INFLUENCE OF RESOLUTION AND FRAME RATE ON THE LINEAR IN-STREAM VIDEO AD QOE.. *Journal of Information Technology ,* 4(1), pp. 15-22.

Lopes, N. & Ribeiro, B., 2015. Deep Belief Networks (DBNs) . In: *Machine Learning for Adaptive Many-Core Machines - A Practical Approach.* s.l.:Springer International Publishing, pp. 155-186.

Mackiewicz, A. & Ratajczak, W., 1993. Principal components analysis (PCA). *Computers and Geosciences,* 19(3), pp. 303-342.

Menkovski, V., Exarchakos, G. & Liotta, A., 2010. *Machine Learning Approach for Quality of Experience Aware Networks.* Thessaloniki, IEEE.

Mok, R., Luo, X., Chan, E. & Chang, R., 2012. *QDASH: A QoE-aware DASH system.* Chapel Hill, North Carolina, ACM New York.

Moore, D. S., McCabe, G. P. & Craig, B. A., 2012. Inference for Regression. In: K. Carson, ed. *Introduction to the Practice of Statistics.* 7th ed. New York: W H Freeman and Company, pp. 533-574.

Natha, R., Jacksonb, W. M. & Jones, T. W., 1992. A comparison of the classical and the linear programming approaches to the classification problem in discriminant analysis. *Journal of Statistical Computation and Simulation,* 41(1-2), pp. 73-93.

Nguengang, G., 2011. *Unified Management Framework: Processes and tools for establishing autonomics in the management of networks and services.* Sophia-Antipolis, France, www.univerself-project.eu.

North, B. & Blake, A., 1998. *Learning dynamical models using expectation-maximisation.* Bombay, IEEE.

NTT Network Technology Laboratories, 2015. *Video quality assessment methods.* [Online] Available at: http://www.ntt.co.jp/qos/qoe/eng/technology/visual/02_4_2.html [Accessed 10 April 2015].

Penga, C.-Y. J., Leea, K. L. & Ingersolla, G. M., 2002. An Introduction to Logistic Regression Analysis and Reporting. *Journal of Educational Research,* 96(1), pp. 3-14.

Qiao, Y. et al., 2015. SVDN: packetization and layer synchronization for scalable video delivery over peer-to-peer networks. *Multimedia Systems,* 21(4), pp. 327-343.

Quinlan, J. R., 1986. Induction of decision trees. *Machine Learning,* 1(1), pp. 81-106.

Reidmiller, M., 1994. Advanced supervised learning in multi-layer perceptrons — From backpropagation to adaptive learning algorithms. *Computer standards and interfaces,* 16(3), pp. 265-278.

Rejaie, R., Handley, M. & Estrin, D., 2000. Layered Quality Adaptation for Internet. *IEEE Journal on Selected Areas in Communications,* 18(12), pp. 2530-2543.

Rosenblatt, F., 1958. The perceptron: A probabilistic model for information storage and organization in the brain.. *Psychological Review,* 56(6), pp. 386-408.

Ruckert, J. et al., 2012. *Quality Adaptation in P2P Video streaming Based on Objective QoE Metrics.* Prague, Springer.

Russell, S. & Norvig, P., 2010. *Artificial Intelligence: A Modern Aproach.* 3rd ed. New Jersey: Pearson Education.

Russel, S. & Norvig, P., 2010. Learning from Examples. In: M. Hirsch, ed. *Artificial Intelligence.* New Jersey: Prentice Hall, pp. 693-757.

Russel, S. & Norvig, P., 2010. Learning Probabilistic Models. In: M. Hirsch, ed. *Artificial Intelligence.* New Jersey: Prentice Hall, pp. 802-830.

Russel, S. & Norvig, P., 2010. Reinforcement Learning. In: M. Hirsch, ed. *Artificial Intelligence.* New Jersey: Prentice Hall, pp. 830-853.

Salakhutdinov, R., Mnih, A. & Hinton, G., 2007. *Restricted Boltzmann machines for collaborative filtering.* New York, ACM.

Sammon, J., 1969. A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers,* 18(5), pp. 401-409.

Sandvine, 2015. *Measuring Internet Video Quality of Experience.* [Online]
Available at: https://www.sandvine.com/downloads/general/whitepapers/measuring-internet-video-quality-of-experience.pdf
[Accessed 30 March 2015].

Schierl, T. et al., 2007. *Using H.264/AVC-based Scalable Video Coding (SVC) for Real Time Streaming in Wireless IP Networks.* New Orleans, IEEE, pp. 3455-3458.

Schwarz, H., Marpe, D. & Wiegand, T., 2007. Overview of the Scalable Video Coding. *IEEE Transactions On Circuits And Systems For Video Technology,* 17(9), pp. 1103-1120.

Singh, K., Ksentini, A. & Marienval, B., 2011. *Quality of Experience Measurement Tool for SVC Video Coding.* Kyoto, IEEE, pp. 1-5.

Sodagar, I. & Giladi, A., 2015. MPEG-DASH State of Affairs.. *Streaming Media.*, September, pp. 67-70.

Stankiewicz, R. & Jajszczyk, A., 2011. A survey of QoE assurance in converged networks. *The International Journal of Computer and Telecommunications Networking,* 55(7), pp. 1459-1473.

Steinmetz, R. & Wehrle, K., 2005. *P2P Systems and Applications.* 1st ed. Berlin: Springer-Verlag.

Stockhammer, T., 2011. *Dynamic adaptive streaming over HTTP--: standards and design principles..* s.l., s.n., pp. 133-144.

Strassner, J., 2003. *Policy based network management: Solutions for the next generation.* 1st ed. San Francisco: Morgan Kaufmann.

Sutton, R. & Barto, A., 1998. *Reinforcement Learning: An Introduction.* 1st ed. London: MIT.

Szepesvari, C., 2009. *Algorithms for Reinforcement Learning,* s.l.: Morgan & Claypool.

Telecommunications Industry Association, 2015. *Telecom Glossary.* [Online]
Available at: http://www.tiaonline.org/resources/telecom-glossary
[Accessed 15 July 2015].

Thang, T. C., Ho, Q.-D., Kang, J. W. & Pham, A., 2012. Adaptive streaming of audiovisual content using MPEG DASH. *IEEE Transsactions on Consumer Electronics,* 58(1), pp. 78-85.

The Economist, 2014. *Streaming Media: Video on demand.* [Online]
Available at: http://www.economist.com/news/technology-quarterly/21635325-online-video-continues-boom-publishers-are-exploring-new-ways-deliver
[Accessed 17th April 2015].

Tibshirani, R., 1996. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B,* 58(1), pp. 267-288.

Tucker, W. & Westereveld, R., 2015. ICT4D and Local Access. In: R. M. P. H. A. C. S. S. v. d. G. P. B. A. K. J. D. I. S. B. D. K. D. J. G. Robin Mansell, ed. *International Encyclopedia of Digital Communication and Society.* s.l.:Wiley and Sons, pp. 365-372.

Univerself, 2015. *Univerself.* [Online]
Available at: http://www.univerself-project.eu/
[Accessed 24 june 2015].

UPC, 2015. *Broadband.* [Online]
Available at: http://www.upc.ie/broadband/
[Accessed 15 June 2015].

Venkataraman, M. & Chatterjee, M., 2011. Inferring Video QoE in Real Time. *IEEE Network,* 25(1), pp. 4-13.

Watkins, C. J. C. H. & Dayan, P., 1992. Q-learning. *Machine Learning,* 8(3-4), pp. 279-292.

Wiegand, T., Sullivan, G. J., Bjøntegaard, G. & Luthra, A., 2003. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions On Circuits And Systems For Video Technology,* 13(7), pp. 560-576.

Woergoetter, F. a. P. B., 2008. Reinforcement learning. *Scholarpedia,* 3(3), p. 1448.

Wong, J. A. H. a. M. A., 1979. Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics),* 28(1), pp. 100-108.

Xiao, X., 1999. Internet QoS: a big picture. *IEEE Network,* 13(2), pp. 8-18.

Zambelli, A., 2013. *A history of media streaming and the future of connected TV.* [Online] Available at: http://www.theguardian.com/media-network/media-network-blog/2013/mar/01/history-streaming-future-connected-tv
[Accessed 15th January 2014].

Zhang, X. & Hassanein, H., 2012. A survey of peer-to-peer live video streaming schemes – An algorithmic perspective. *Computer Networks,* 56(15), pp. 3548-3579.

Zhou, Y., Chiu, D.-M. & Lui, J. C. S., 2011. A Simple Model for Chunk-Scheduling Strategies in P2P Streaming.. *IEEE/ACM Transactions on Networking,* 19(1), pp. 42-54.

Zou, H. & Hastie, T., 2005. Regularization and variable selection via the elastic net. *Journal of The Royal Statistical Society - Series B (Methodology),* 67(2), pp. 301-320.

Appendix

The accompanying resource DVD contains

- The Excel models used in the proof of concept
- The source code for the simulator
- The simulator output files
- The simulation results tables