Software Defined Intelligent Networks for Fairness and Quality of Experience Prediction and Evaluation

Ahmed Osama Basil Al-Mashhadani

A thesis submitted to the

University of Northampton

for the degree of

Doctor of Philosophy

May 2023

Abstract

This thesis addresses the challenges posed by the increasing demand for online distribution of high-quality and high-throughput content. The growing number of media applications competing for network resources results in a significant impact on network efficiency and the Quality of Experience (QoE). In a multi-user multi-device environment, measuring and maintaining perceivable user feedback and fairness becomes as critical as achieving the QoE on individual user applications.

To overcome these challenges, this thesis proposes novel framework designs using programmable networks such as Software-Defined Networks (SDN). The framework automates the measuring and maintenance of perceivable user feedback and fairness, leading to improved QoE and network efficiency. The proposed framework involves the use of Machine Learning (ML) techniques to build a prediction model based on subjective user experiments. This eliminates the need for physical experiments and automates the process of predicting QoE.

Adaptive streaming over a software-defined network environment is also being examined in the framework, evaluating and studying the media streams, aspects affecting the stream, and the network. By analyzing the network's features and their direct relationship with the perceived QoE, the framework improves the network's optimization and reduces the discrepancy of QoE across user devices. Finally, this thesis discusses application- and human-level fairness over networked multimedia applications and how such fairness can be managed through novel network designs using programmable networks such as SDN. By achieving both fairness and QoE, the proposed framework can provide a comprehensive solution to the challenges posed by the increasing demand for online distribution of high-quality and high-throughput content.

List of Publications

The research for this thesis was conducted at The University of Northampton, specifically within The Faculty of Arts, Science and Technology. The primary focus of this thesis is highlighted in Chapters 3-5, where the significant findings and contributions are discussed. These chapters are based on the publications listed below.

1. **Basil, Ahmed** Sharman, James Goldsney, Jacob. (2019). P4-Assisted Network Security for Future Smart Homes.
2. **Basil, Ahmed.** (2019). A Consistency Based Research: P4 Versus OpenFlow and the Future of Software Defined Networks. University of Northampton Annual Research Conference.
3. **A. Basil**, M. Mu and M. Agyeman, "A Multi-modal Framework for Future Emergency Systems," 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC /ATC/CBDCom/IOP/SCI), 2019, pp. 17-20.
4. **Ahmed Osama Basil**, Mu Mu, and Ali Al-Sherbaz. 2019. A Software Defined Network Based Research on Fairness in Multimedia. In Proceedings of the 1st International Workshop on Fairness, Accountability, and Transparency in MultiMedia (FAT/MM '19). Association for Computing Machinery, New York, NY, USA, 11–18.
5. **A. O. Basil**, M. Mu and A. Al-Sherbaz, "Novel Quality of Experience Experimentation Framework Through Programmable Network Management," 2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC), 2022, pp. 485-486.

6. **Al-Mashhadani, A.O.B.**; Mu, M.; Al-Sharbaz, A. Quality of Experience Experimentation Prediction Framework through Programmable Network Management. Network 2022, 2, 500-518.
7. **Al-Mashhadani, A.O.B** and Al-Khafajiy Mohammed, "Chapter4 - Big Data Analytics for IoT: Technologies, Importance and Algorithms". Elsevier 2023, 4.

Declaration

The research presented in this thesis was conducted at The University of Northampton, specifically within the Faculty of Arts, Science, and Technology. Unless specified otherwise, the content of this thesis is the original work of the author. The author, while registered as a candidate for the Doctor of Philosophy degree, has not pursued any other awards concurrently. Furthermore, this thesis has not been submitted, either in its entirety or partially, for any other degree.

Ahmed Osama Basil Al-Mashhadani
Faculty of Arts, Science and Technology
The University of Northampton
Waterside Campus,
University Dr
Northampton
NN1 5PH
UK

February 6, 2024

Declaration of Authorship

I, Ahmed Osama Basil Al-Mashhadani, declare that this thesis titled, 'Software Defined Intelligent Networks for Fairness and Quality of Experience Prediction and Evaluation' and the work presented in it are my own. I confirm that:

- This work was primarily undertaken during the author's candidature for a research degree at this University.
- Any instances where portions of this thesis have been previously submitted for a degree or qualification at this University or any other institution are explicitly indicated.
- Whenever the author has referred to published works of others, proper attribution has been consistently provided.
- In cases where direct quotations from the works of others are included, proper source citations are always provided. Except for these specific quotations, the entirety of this thesis represents the author's original work.
- The author acknowledges all significant sources of assistance received during the course of this research.
- In instances where the thesis incorporates collaborative work involving the author and others, a clear distinction is made between the contributions made by the respective parties.

Signed: Ahmed Osama Basil Al-Mashhadani

Date: February 6, 2024

Acknowledgements

Completing a PhD is an arduous yet rewarding journey that one may experience only once in a lifetime. I am truly elated to have overcome the challenges and approached the final stages of this remarkable expedition. At this juncture, I would like to express my heartfelt gratitude and thanks to the Almighty God for bestowing blessings upon me. I am deeply aware that this PhD journey would not have been possible without the unwavering support of the people around me.

I would like to express my gratitude to my two supervisors, **Dr. Mu Mu** and **Dr. Ali Al-Sharbaz**, for their invaluable aid, guidance, and unwavering supervision throughout my PhD journey. Their expertise, wisdom, and dedication have been instrumental in shaping the trajectory of my research and fostering my academic growth. Their profound knowledge in their respective fields has provided me with invaluable insights and perspectives that have significantly enriched the quality of my work.

I want to convey my deepest gratitude and heartfelt appreciation to my parents, **Osama Basil Abdulrazzaq** and **Adian Tawfiq Taha**, for their unwavering emotional, intellectual and financial support throughout my PhD journey. Their selfless sacrifices, boundless love, and unwavering belief in my abilities have been the driving force behind my perseverance and success. Their constant encouragement, understanding, and unwavering presence have provided me with the strength and motivation to overcome challenges and pursue my academic goals. I am forever indebted to them for their unconditional support and for instilling in me the values of resilience, determination, and the pursuit of knowledge. Their unwavering support has been the cornerstone of my achievements. I also wish to seize this opportunity to express my sincere appreciation to my sisters **Israa** and **Aaya**, for their encouragement throughout this journey. Your support has been a source of inspiration and has meant the world to me.

I wish to extend my sincere gratitude to the Faculty of Arts, Science, and Technology at the **University of Northampton** for awarding me the scholarship that made it possible for me to pursue this Ph.D. Your support and generosity have played a crucial role in enabling me to undertake this research journey. I am truly grateful for this incredible opportunity. Finally, I extend my deepest gratitude to all my friends and colleagues who have contributed to my academic and personal growth. Your support and camaraderie have made a significant impact on my life and have played a vital role in shaping me into the person I am today.

Abbreviations

List of commonly used acronyms;

Acronyms

AI Artificial Intelligence

DASH Dynamic adaptive streaming over HTTP

DPI Deep Packet Inspection

FFM Fairness Flow Model

FRA Flexible Resource Allocation

FTP File Transfer Protocol

HTTP Hypertext Transfer Protocol

IP Internet Protocol

ITU International Telecommunication Union

MAP Media Access Control Layer and Application Layer Performance Assessment

ML Machine Learning

MOS Mean Opinion Score

MPD Media Presentation Description

MPEG Moving Pictures Expert Group

PI Packet Inspection

QoE Quality of Experience

QoS Quality of Service

SDN Software-Defined Networks

SLA Service Level Agreement

TCP Transmission Control Protocol

UDP User Datagram Protocol

Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

"Fairness is a fundamental principle in the development and deployment of technology. As society becomes increasingly reliant on technology, it is important to ensure that access and benefits are distributed equitably. Without fairness, innovation and progress risk leaving some features and qualities behind."

Ahmed Al-Mashhadani

1.1 Introduction

Recent years have seen a dramatic increase in the general quality of experience expected by the users [6] over media delivery platforms such as online video streaming. As the number of online user applications and use devices grows, the capability of end devices varies significantly and resources such as network bandwidth are often shared between multiple user devices. Hence the quality that can be achieved on a single device is often limited by its share of resources. Conventional best-effort networking infrastructures allocate resources based on client requests and high-level Service Level Agreement (SLA) without the awareness of application or user-level requirements. This often leads to unfairness perceivable by the end user. There is a considerable amount of literature on the concept of fairness (some of which are discussed in Chapter 2). However, most studies have their scopes of vision, such as identifying the fairness issue that needs adjusting and conducting the research needed to rectify a single issue. However a major challenge with this kind of topic is the perspective, as mentioned before, fairness can be looked at from different levels, and aspects. Fairness can be considered from a hardware level or a software level, it can be shown in QoE and Quality of Service (QoS). Fairness is defined by the Cambridge Dictionary as considering everything that affects a situation so that a fair judgment can be made. The issue of fairness has been a controversial and much-disputed subject within the field of multimedia.

There is a lack of holistic research on fairness on online multimedia. Networking companies and operators started taking a vast interest in the SDN industry for its effectiveness and usefulness in terms of the creation of the right network architecture with the needed aspects, furthermore, it is necessary to test these architectures with the right tools such as API programming data plane switches such as OpenFlow switches and their protocols. Due to the increased demand in the flexibility of the network control plane, there's been an interesting evolution in the SDN market in recent years. A survey [7] was done in 2017 on 294 networking professionals stating that 49% are either considering or actively using an SDN implementation; 18% have one installed already. International Data Corporation (IDC), a global provider of market intelligence, has identified many use cases for SDN today.

1.2 Motivation: *fairness & its scope*

Software-defined networking, with its architectural approach, is thought of as the separation of the management of the control plane of devices from the underlying data plane that forwards network traffic [8]. SDNs mainly introduce an abstraction layer separating network configuration from the physical communication resources. This is highly useful since a network operating system running inside a control layer which is sandwiched between the application and infrastructure layers allows more room for applications to re-configure dynamically to adapt to their security, scalability and manageability needs [9]. Throughout this thesis, the term level will refer to a stage and a modal layer.

Figure 1.1: Fairness Research Aspects Diagram

SDN is being used to manage networking traffic and/or split up network connections between end-users and data centers. A network could be split into multi-segments, conceivably a public and a private segment where each segment has its own security applications installed such as firewalls and encryption policies with access control lists. Investigation and research will cover fairness in aspects that exceed resource allocation and QoE. Fairness of efficiency and the role of network management to make a network fair and efficient will also be discussed in this thesis. This work provides an important opportunity to advance the understanding of fairness and how it can be organized and achieved. This section of the report further defines SDN and its impact on businesses and networks in general which was the main motivation behind the Fairness research on multimedia technologies. Figure 1.1 shows the traditional fairness research scopes.

1.3 Research Problem and Questions

The purpose of this study is to review recent research into the definitions of fairness along with QoE and how researchers attempt to tackle the related challenges. This thesis also investigates how programmable networks such as SDN can be used to improve the fairness of online multimedia applications. Most current literature on fairness pays particular attention to partial issues or single levels, and by solving them, partial fairness is gained. For example, Y. Wang [10] discusses the wide scope of objective video quality measurements, similarly, U. Engelke [11, 12] and their discussion over the perceptual-based quality metrics for image and video-related services. These scholars defined the problem domain in their surveys and proposed a partial argument for creating better service feedback. This thesis aims to use the mentioned research to understand the wider issue in multimedia and how to improve the QoE from the user end. More research was conducted that included methods of classification in the realm of video quality assessment [13, 14], which is also a field of research that the thesis will cover to improve machine-learning techniques for solving and predicting QoE. Thus this thesis's research questions must target a wide array of fairness and QoE experimentation to provide a conceptual framework supported by SDN-based monitoring and management for online multimedia streaming. The thesis aims to answer three major questions which are:

- **Question I** What is Fairness in media technology and the relationship between Fairness and the Quality of Experience?
- **Question II** How to measure/classify Fairness and relate it to QoE?
- **Question III** How is QoE preserved from SDN fairness and how can it be predicted most efficiently?

1.4 Thesis Aim and Objectives

Research Aim: The project aims to develop a software-defined networking experimentation methodology that employs SDN-assisted and QoE resource monitoring to investigate the user experience, user-level fairness, and network efficiency of online adaptive media in a household environment. The project seeks to address the non-cooperative competition for network resources among a growing number of media applications, which negatively impacts network efficiency and user experience quality, as well as results in QoE discrepancies across user devices. The thesis aims to optimize network quality by predicting QoE preservation and maintaining fairness in a multi-user, multi-device environment. The project will create multiple plug-and-play testbeds using a P4-focused networking back end to evaluate the effectiveness of the developed software-defined cognitive networking.

Research Objectives: to tackle the research questions and achieve the research aims, this thesis identified objectives that are essential to developing the desired solution. The objectives are applied to guide the research process, and hence the following research objectives have been identified:

- **Define** All Technological Aspects of Fairness
- **Understand** User/Human-level Fairness
- **Develop** Network Inspection Tool that can Record Network-level Data and produce Training Ready Features
- **Design** a QoE prediction framework for assessing positive user experience in experiments.
- **Develop** Normalised network and QoE features database for future feature dependant research.

1.5 Research Methodology

1. Problem identification: a survey on the state-of-the-art research studies has been conducted to acquire a full knowledge about Fairness in Multimedia, its challenges and definitions. Naturally, this aids the research in the identification of the research landscape while analysing the gap in research. Problem identification took some critical thinking of the current solutions that are out there for streaming media content at the best user experience with an acceptable network efficiency model, while understanding previous strategies developed, there is a gap in the understanding of fairness and trying to achieve it on a large scale in the network. To understand where fairness occurs and its conditions, a fairness definition model is required, after defining the correct scope, proceeding with the technical approaches is the next step.

2. Objectives of the solution: identifying the objectives was driven by the identified problem, hence this required accurate knowledge about the state of the problem, its current solutions, and their efficacy. The proposed solution aims to provide a complete design and implementation to help deploy intelligent software-defined networking algorithms to achieve network and user-level fairness and efficiency. Thus, the objective of the solution is to provide; (*i*) a fairness flow model, and (*ii*) resources allocation that involves two main technical algorithms:

   - A network inspection tool that can extract all the needed network-level features from the experiment. Moreover, this data will help the resource allocation algorithm to divide the network in an efficient manner.
   - A Quality of Experience Model that can predict the QoE of a certain host computer in the experiment. This will aid in the improvement of the QoE and resource allocation algorithms' thresholds improvements and analysis.

3. Design and development: during this phase, agile methodology is adopted, thus the proposed solution was designed and implemented in parts so that intensive testing and evaluation could be carried out on each part, hence meeting the desired objective. Chapter 3 discusses the thesis's testbeds, other measurement tools compared to this thesis's approach and Load Balancing (the connection between fairness and QoE). Chapter 4 discusses the QoE experimentation framework along with all experiment details and routing rules used. Chapter 5 discusses the classification and prediction techniques used along with a state-of-art comparison to the analysis.

4. Demonstration: in this thesis, multiple experimentation-related scenarios are adopted to demonstrate the usefulness of the research and highlight the advantages of using a reliable fairness system in Multimedia Streaming Applications. Also, a technical simulation is provided to demonstrate the network resource management efficiency of the algorithms.

5. Evaluation: the evaluation stage of the proposed solutions includes multiple test-bed scenarios and simulations that have been implemented simply to test the solution and its effectiveness. It is

reflected in the QoE program that was created to show the predicted experience for actual users. The efficiency of the network must also be evaluated as it is a part of the fairness stages that will be explained in chapter 2.

1.6 Contributions to Knowledge

The Contributions to Knowledge section of this thesis outlines the thesis's findings and insights that have been generated through the research conducted for this study. These contributions represent a unique insight into the field of multi-media and computer networking, providing a deeper understanding of the complex issues surrounding the Quality of user Experience. The research has not only contributed new knowledge to the existing literature but has also generated practical applications and recommendations for future research, policy, and practice. The following section summarizes the key contributions of this research, figure 1.2 shows the contribution by order of appearance in the thesis.

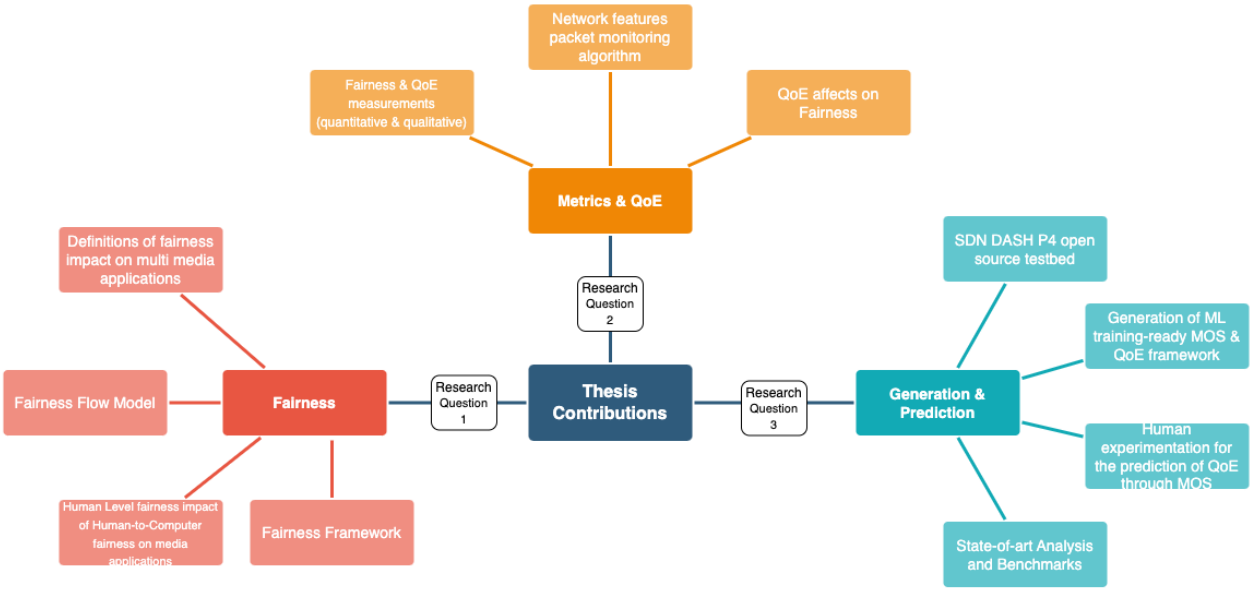


Figure 1.2: Contributions to knowledge summary

In the figure above, each research question has multiple parts, these parts fulfil the stated research question.

The main contributions of this thesis include:

- **Contribution 1:** The following contribution was made to advance the knowledge and understanding of fairness in online multimedia applications. Firstly, this study reviewed existing definitions of fairness and analyzed their applicability in this context, resulting in valuable insights into how fairness can be defined and measured in online multimedia. Secondly, a fairness flow model was developed to evaluate fairness in online multimedia applications at different levels, providing a comprehensive framework for evaluating fairness. Thirdly, this study proposed a new human-level fairness consideration that highlights the significance of considering the human perspective in designing fair multimedia systems. Finally, the study examined the impact of human-to-computer fairness on media applications, providing valuable insights into how fairness can be enhanced by considering the user's needs and preferences. This contribution was published in "software-defined network based research on fairness in multimedia" [15].

- **Contribution 2:** To facilitate the evaluation of fairness in multimedia systems, several tools and resources were developed in this study. First, a Network Inspection Tool and a QoE Measurement Tool were created to monitor network performance and user experience accurately. These tools provide scholars with reliable and comprehensive means of assessing fairness in online multimedia applications. Second, a segmented content database was developed, which contains a wide range of encoding configurations and resolutions. This database is a valuable resource for evaluating multimedia systems and conducting experiments. Third, a hybrid simulation environment was created using P4 [16] and OpenFlow [17], which provides a flexible and customizable platform for evaluating the effectiveness of proposed solutions. Finally, the study proposed a human-level fairness consideration that highlights the importance of considering the user's needs and preferences in designing fair multimedia systems. Together, these contributions provide a comprehensive framework for evaluating fairness in online multimedia applications and offer valuable tools and resources for conducting research in this area. The thesis's open-source test bed configuration and database are shown here [18]. This contribution was published in "Novel quality of experience experimentation framework through programmable network management" [19].

- **Contribution 3:** The third contribution to this research is focused on improving the process of the accuracy of QoE predictions in multimedia systems through the use of ML. First, an experimentation framework structure was proposed through programmable network management, enabling the generation of ML training-ready data and prediction of MOS/QoE. This practical solution allows for more accurate QoE predictions, improving the overall user experience. To evaluate the effectiveness of this proposed solution, human experiments were conducted with QoE MOS-based feedback. This benchmarking process provided a reliable and accurate method for measuring the accuracy of predicted QoE and network features. The results obtained from these experiments were used to validate the proposed experimentation framework and confirm its effectiveness in improving QoE predictions. Additionally, an analysis of state-of-the-art machine learning algorithms was conducted, and an experimentation framework for feature evaluation in network experiments was developed. This framework provides valuable insights into how machine learning can be used to improve fairness in multimedia systems. By identifying the most effective ML algorithms and features for predicting QoE, this research contributes to the development of more efficient and accurate multimedia systems. This contribution was published in "Quality of experience experimentation prediction framework through programmable network management" [20].

1.7 Thesis Organisation

The structure of the thesis and its chapters are depicted in Figure 1.3. It is important to note that this thesis and its contents have been built upon a culmination of research publications produced during the PhD journey.

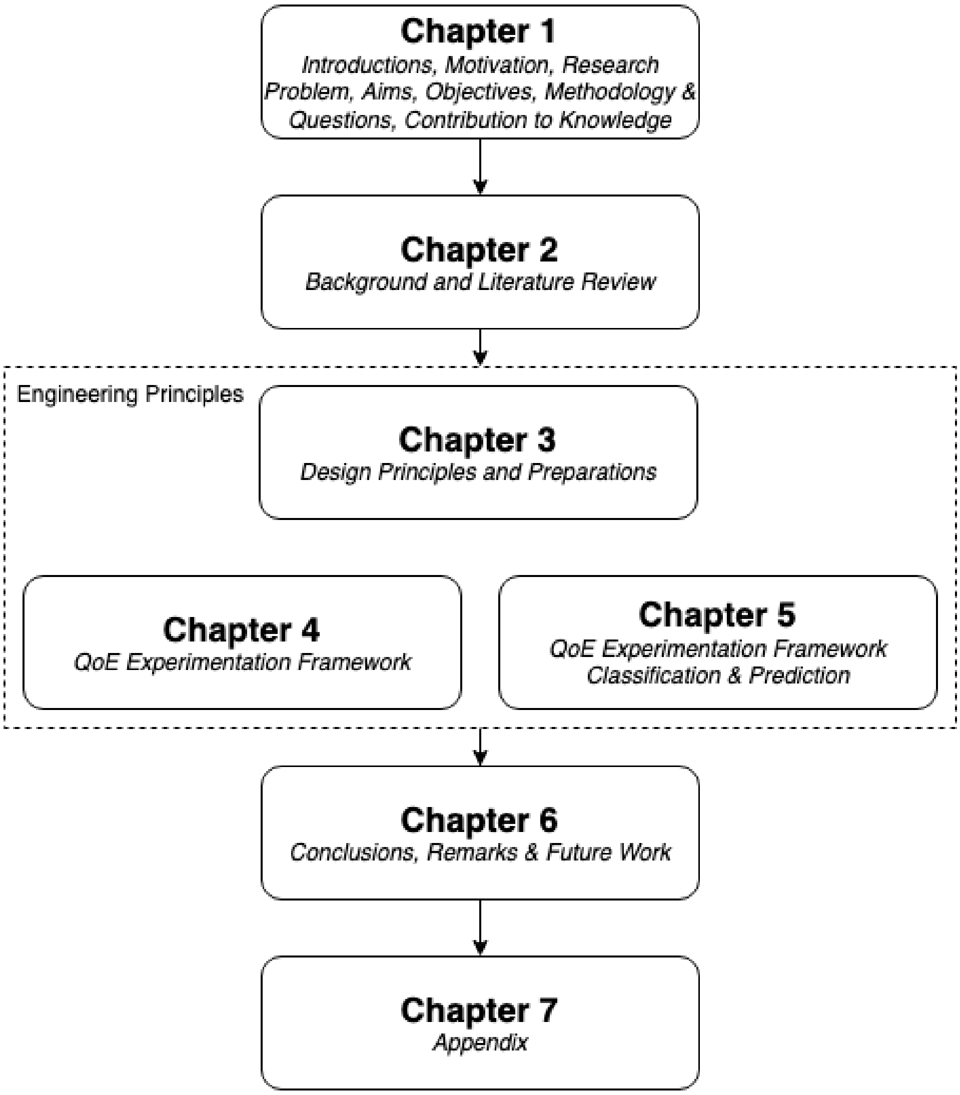


Figure 1.3: Thesis Structure

The overall structure of the study takes the form of six Chapters.

Chapter 1: This chapter provides an overview of the motivation for conducting the study, as well as a discussion of fairness and its scope. The research problem and questions are identified and related to the overall objectives of the study, which focus on fairness and QoE. Additionally, separate sub-chapters are dedicated to research methodologies and contributions to knowledge, providing clarity on the scope of the thesis and where to locate specific contributions.

Chapter 2: This chapter provides an overview of the background research study, which includes a comprehensive survey of related literature and a thorough definition of fairness using a structured classification approach known as the Fairness Flow Model. Additionally, the chapter presents a critical evaluation of the research gap, supplemented by an extensive list of quantitative and qualitative measurement tools.

Chapter 3: The chapter on Design Principles and Preparation presents a comprehensive list of design requirements, objective metrics, and subjective evaluation of DASH. Additionally, it provides a detailed case study and testbed setup, which includes a packet monitoring algorithm, experimental configuration, and quantitative techniques to measure the quality of the testbed. The chapter also includes subsections on expectations and performance evaluation of the testbed.

Chapter 4: The Experimentation chapter presents the results of user experiments, which were conducted with strict rules to control the experimental setup. The thesis presents the data analysis of the test-bed output, along with the experimental plan and data description. Additionally, the thesis explores further configurations of the test bed, specifically regarding dynamically changing networks using a reinforcement learning congestion control approach.

Chapter 5: In this chapter, machine learning classification is applied to data generated from a user experiment conducted in previous chapters. A detailed discussion on state-of-the-art comparison is also presented.

Chapter 6: Concludes and summarises the contributions and discusses the future work and remarks.

CHAPTER 2

Background and Literature Review

Think of it as a general language or an instruction set that lets me write a control program for the network rather than having to rewrite all of code on each individual router

Scott Shenker, a Berkley

2.1 Introduction: Background Research

Software-defined networking is a novel approach to networking that separates the control and data planes, enabling more flexible and programmable network management. SDN has emerged as a promising technology for supporting multimedia applications, such as video streaming, which require high QoE. QoE is a measure of user satisfaction with the quality of a multimedia service, and it is influenced by various factors, including network performance, content characteristics, and user preferences. This background research aims to explore the concepts of fairness and Quality of Experience (QoE) in the context of Software-Defined Networking (SDN) and multimedia technology. Additionally, it aims to investigate the use of training-ready databases for predicting QoE.

Fairness is a critical issue in SDN, as it involves ensuring that all users receive an equitable share of network resources. In the context of multimedia applications, fairness is particularly important, as unequal allocation of resources can lead to poor QoE, such as video buffering or freezing (this is discussed in chapter 3). SDN can provide a more fair allocation of resources by enabling dynamic and fine-grained control of network resources, based on the needs of individual users and applications. For example, SDN can prioritise traffic based on QoE requirements or dynamically adjust network paths to avoid congestion.

QoE is a multifaceted concept that encompasses various subjective and objective factors. In multimedia technology, QoE is often measured in terms of video and audio quality, as well as user engagement and satisfaction. Video quality can be affected by factors such as video resolution, bit rate, and encoding format, while audio quality can be influenced by factors such as sampling rate and compression. User engagement and satisfaction can be measured through various metrics, such as viewing time, clicks, and ratings. QoE can be affected by various network parameters, such as delay, packet loss, and bandwidth [21].

Predicting QoE is a complex task that involves analyzing various parameters and their interactions. Machine learning techniques are used to predict QoE, based on a training dataset that captures the relationships between QoE and various parameters. Training-ready databases are datasets that have been preprocessed and curated for use in machine-learning models. These databases can include features such as network performance data, content characteristics, and user behaviour data. They can be used to train and validate machine learning models that can predict QoE, based on real-time network conditions.

Thus, fairness and QoE are important considerations in the context of SDN and multimedia technology. SDN can provide a more fair allocation of network resources, based on QoE requirements. QoE is a complex concept that involves various subjective and objective factors and can be predicted using machine-learning techniques and training-ready databases. Predicting QoE accurately is crucial for providing high-quality multimedia services and enhancing user satisfaction.

Fairness can be addressed and researched from numerous points of view. In some cases, researchers dismiss the idea that fairness can be targeted, because the methodology of equal opportunity to the individuals in resource sharing may not always mean equal allocation of resources, however, other researchers discussed a different perspective to the one presented in this thesis; where a fair allocation can always be the outcome of a process where individuals do not have equal opportunity [21]. Thus the objective and target of any fairness must be explained. Not to forget that fairness on different networking levels will be discussed, from a single user within the network and the network as a whole.

The past twenty years have seen increasingly rapid advances in the field of media streaming, it has evolved greatly from the ability to download videos to streaming them [22]. Due to the high demand for media streaming, media companies and internet providers take QoE in high priority and seriousness [23]. The high demand naturally resulted in a high number of requests, real-time requirement of the video feedback, and high bit rates of the media content [24]. On a user level, the users' expectations include the ability to stream with ease in their environment while engaging in other network activities. Thus media fairness can be improved by ameliorating the initial delay, and interruptions of the playback which is known as stalling [25]. These aspects are improved by creating new algorithms that aim to make a fair distribution of resources for all users to maximize users' experience and satisfaction.

2.2 Related Works & Defining Fairness

Fairness may have different definitions and interpretations developed in various contexts. It is necessary here to explore the linguistic definition of "fairness". As mentioned in the first chapter, the Cambridge Dictionary defined fairness as "considering everything that affects a situation, so that a fair judgment can be made", it is also defined as "the quality of treating people equally or in a way that is right or reasonable" [26]. Oxford Dictionaries defined fairness as "Impartial and just treatment or behaviour without favouritism or discrimination" [27]. In "Vocabulary" online dictionary, fairness is defined as "The quality of making judgments that are free from discrimination" [28].

Fairness is a term frequently used in literature, to date there is a clear consensus about its meaning, as shown above, the definitions share a common factor which is the distribution of the fairness target object in an equal and organized manner between all contributed parties. There is a degree of uncertainty around the terminology in the field of technology. Moreover, these definitions are limited to the verbal sense of the sentence which is dependent on a scenario, however, they all focus on equality, justice, quality of distribution, and the fairness target audience's satisfaction. This section aims for the understanding of these definitions in the field of technology in media streaming.

Fairness in networking is viewed from different perspectives. Allocation of resources is one of the most used strategies to achieve network fairness. To achieve a better QoS, Flexible Resource Allocation (FRA) strategies are often used. FRA's main objective is to assign a unique variable identification number of resources to the active calls [1]. In Figure 2.1, the wireless network shows all nodes with the same priority. If each node in this example is assigned the same bandwidth and priority, achieving the fairness required would be theoretically possible, however if the priority of connection C5 is lower than that of

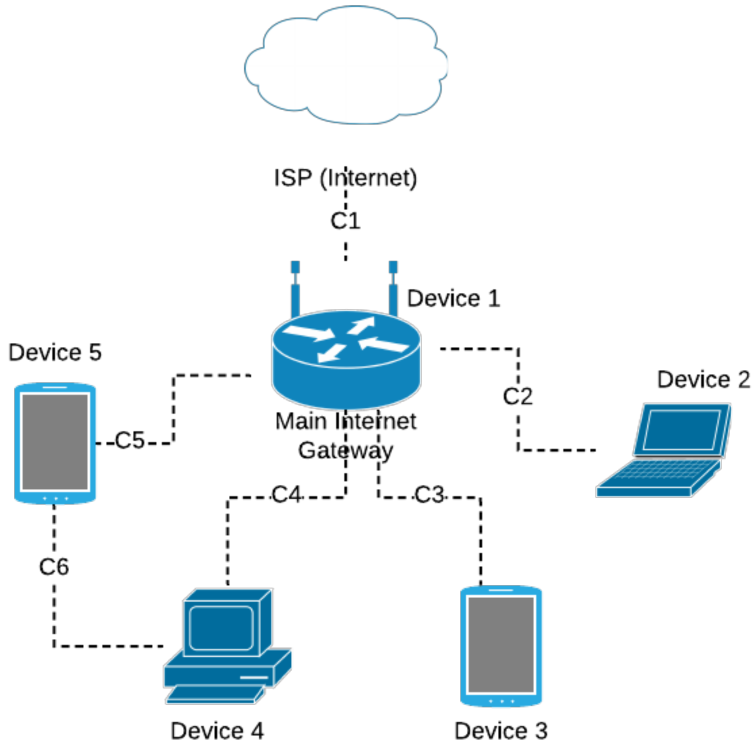


Figure 2.1: Wireless network diagram that shows a set of devices with five nodes and six links where they all access the Internet from one gateway node [1].

connection C4, then it will take a longer time for device number 5 to access the network than device number C4 because the requested network traffic will flow through the forth and the sixth connection to reach the fifth device, thus unfairness is detected in the network. With the right resource allocation strategy fairness can be achieved, but this is dependent on the user and network traffic, to be precise; if the requested traffic load is high, FRA offers the technique of a small number of resources to be assigned to offer high capacity feedback. This process occurs when providing a QoS lower than what is requested. When unfairness is detected, the researcher must find where the unfairness is located or the reason for its generation. Moreover, a target/objective must be chosen to apply a developed solution. Locating and targeting unfairness is essential so that the fairness model/solution works within the correct designated unfair location.

Element of Time in Fairness; in the case of time and duration of fairness, it is crucial to understand the importance of long-term and short-term network-level fairness. Recognising short-term fairness is somewhat of a higher priority for us, short-term fairness is a solution that can provide a fast and temporary adjustment to a network to overcome a certain difficulty. The researcher chose this to be more important because having a set of short-term solutions can be more accurate in achieving fairness than having long-term solutions. This is because short-term solutions can be applied to fix an 'instance' of unfairness and can be run automatically using machine learning Artificial Intelligence (AI). Whereas long-term solutions will be applied to a network without the ability to change the solution as frequently as the short solutions. This is important because not all networking topologies are similar, every network is designed to serve a purpose and achieve a goal, in this case, short-term solutions can be applied often and for multiple proposes. If an equal share of data is allocated over a short instance, then as time progresses, there will be a large number of short instances, resulting in an equal share of data in the long-term fairness. It is also important to recognize that the opposite can be beneficial for static networks with a single purpose. Thus if short-term fairness is achieved, then it will aid in long-term fairness, a study was done in [29] to show differences between long-term and short-term network-level fairness. Looking at Figure 2.1, it can be assumed that every node gets 20% of the network's bandwidth allocation for a short instance, and casually repeated over time, it can be identified that the short instance to be around 20 minutes. In this 20 minutes, given that the correct resource allocation strategy was used and all nodes have the same priority level, fairness is achieved and bandwidth speed is limited to 20% per node. Researcher prioritized the short-term fairness (which was achieved in this example) because it can dynamically update every 20 minutes thus achieving long-term fairness in the long run. This is important to recognise because if one of the devices such as Device 2 achieves a high error rate, then the resulting outcome will be a large request of data, which will end up with a lower data received by the other devices which shows one of the problems that short-term fairness can solve.

In terms of network fairness, in a company, fairness must be considered from two main perspectives; the perspective of the user and the perspective of the company as a whole. Assuming that Figure 2.1 represents a small division within the company. User fairness will be achieved if every user is satisfied with their quality of experience and has full access to their 20% capacity, moreover having a smart network configured with machine-learning aid then more advantages can be achieved and fairness will have a new perspective. If Device 2 was requesting only 5% of the network's bandwidth capacity, and Device 4 is requesting 35% (more than the allocated 20% per node) of the network's bandwidth capacity, and the AI allows the fourth device to get the needed bandwidth and limit the second device to it's needs, while everyone else is also satisfied with their 20% then whole network fairness is achieved for that small company division [21].

Fairness can be achieved in several ways, examples include, new fairness algorithms using SDN [30], load balancing schemes [31], AI switch layer machine learning[32], and many others. This was a short categorization of the types of fairness that should be considered. It is crucial to consider the various viewpoints when discussing network-level fairness in media technology. Thus researcher must highlight the levels of fairness, the techniques for measuring fairness, the tools used and how to achieve fairness.

2.2.1 Fairness Flow Model

The Open Systems Interconnection Model (OSI) is one of the most known models that describe traffic flow from the Network Level, to the Application Level. This section will look at the perspective of fairness from each of the relevant levels, highlighting the Network, transport, and application levels for the software level. Along with a new Level titled the Human Level which will state the importance of fairness between human and computer [33]. Figure 2.2 shows the Fairness Flow Model (FFM) which details the flow of fairness on three different levels, the flow of fairness is directly associated with the number of the level. This section will explain those levels and previous research and techniques discussed on them.

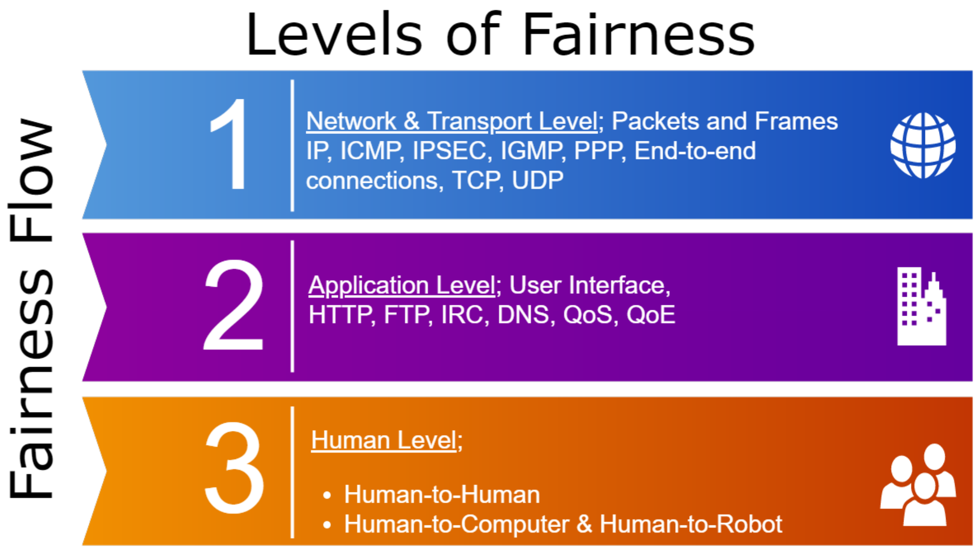


Figure 2.2: Fairness Flow Model (FFM)

2.2.1.1 Network and Transport Level Fairness

Generally for networking and media distribution, the network and transport levels are two of the most important levels to be discussed and defined while noticing their effect on the concept of fairness. The network level is important because its operation, equity, equality, and integrity affect all other levels of fairness. It considers transmission priority, the level of network congestion, QoS and occasionally, dynamic routing for the determination of the cost from one network node to another and identifying the best path for traffic flow [34]. Often in the network-level routing addressing schemes, models and guidelines are re-visited and adjusted to best suit the network's accessibility for data transmission resulting in a better QoE for the end-user. On the other hand, the transport level provides reliable data transmission. If the network level were defined as the beginning of a point-to-point connection process, the transport level could be defined as the concluding stage of the source-to-destination procedure. The transport level is responsible for virtual circuit management, error tracking, correction and recovery, and more importantly flow control and multiplexing. These responsibilities are highly important and should always be considered when working with the term "fairness".

There are multiple network and transport level protocols ranging from Internet Protocol (IP), Internetwork Packet Exchange (IPX), Routing Information Protocol (RIP), and Open Shortest Path First (OSPF) in the network level. Where the transport level includes; Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Sequenced Packet Exchange (SPX) with abundant others. The following subsection will discuss TCP fairness and the algorithms proposed to show network-level fairness and limit steady-packet drop rate.

There are two primary types of Internet Protocol traffic: 1. Transmission Control Protocol and 2. User Datagram Protocol. According to a study mentioned in [35] and [36], after a clear observation of UDP and TCP-based protocols, it was deduced that UDP provides no fairness at all since it is an un-ordered lightweight datagram-based service. Analysis shows that UDP and TCP usage ratios in media-service companies lie between 5% and 20% depending on the popularity of the media server and the characteristics of the end-user's settings and choices. Even though UDP is preferred for applications that are based on media streaming or VoIP and most types of tunnelling applications, it will be difficult to replace the value of TCP due to UDP's unreliability and lack of fairness mechanisms.

TCP in Network-level Fairness TCP treats lost segments as an indicator of congestion. Congestion control is defined as a network-level obstacle that presents itself when an unreasonable amount of data is being sent in the network with minimal lost packets [37]. Moreover, TCP treats packet receive variations as a congestion indicator thus TCP is favoured for the ability to prevent congestion collapse. High-speed TCP congestion control algorithm is an algorithm developed by S. Floyd [38]. A study in [39] shows that fairness is detected in the high-speed congestion algorithm however it has minor restrictions. Regardless of the restrictions, S. Floyd proposed to modify the TCP congestion control mechanism due to the issue that in a steady-state environment, with a packet loss rate p, the average congestion window is around $\frac{1.2}{\sqrt{p}}$ segments. This shows a strict limitation and constraint on the congestion windows that are produced by TCP in realistic environments. Floyd proposed to change the algorithm to gain high throughput for a steady-state packet drop rate. This is quite useful for High-Speed TCP because High-speed TCP share fairly available bandwidth but takes a longer time to converge than standard TCP. Floyd's proposal focuses on modifying the increased perimeter and decreased perimeter of additive increase and multiplicative decrease, resulting in finally increasing the fairness with high-speed congestion and limiting packet drop rate, see example implementations for more on High-speed TCP ([38, 40, 41]). The network level comprises multiple protocols, which can be utilized to enhance fairness and improve user experience. Above was a short example of the idea of manipulating protocols and algorithms to better increase the functionalities and limit the high loss of data packets.

UDP Fairness; UDP is a communication protocol that aims to establish low-latency and loss-tolerating connections between servers on the web, in contrast to TCP. UDP is a best-effort protocol that sends datagrams, unlike TCP which can break large sets of data into packets and re-send any lost packets to assemble a correct sequence for the packets, UDP simply sends packets. Packets may take random paths to reach their destination which often results in non-retrievable lost packets or received packets out of order due to route delays. UDP is favoured by some media applications such as gaming or IPTV streaming, where the loss of a few packets will unlikely affect the resulting outcome in a notable user-end perspective [42]. Applications that require loss-less data transmission may favour UDP as well. Using application-level error control, those applications can re-transmit lost packets, resulting in this protocol being a better choice for large files' data transfer rate compared to TCP.

Since UDP applications have increased, copious attempts were targeted at achieving fairness while using protocols to assist UDP-based High-speed transport protocols. In a performance evaluation [43], RUBDP, Tsunami, UDT, and PA-UDP high-speed UDP transport protocols were used to gather protocol data, the researchers looked at inter-protocol fairness and intra-protocol fairness.

Given that e-science (distributed network collaboration environment) applications are in more demand, communication patterns started preferring point-to-multipoint or multipoint-to-multipoint rather than point-to-point [44]. Intra-protocol fairness shows multiple flows that are using the same protocol and their ability to share bandwidth fairly and/or use the same service with no connectivity obstacles. Since network applications vary in the use of protocols, the inter-fairness term is used to explain an entire flow of data based on different protocols and distribute the link bandwidth fairly between each other. If fairness is achieved on both inter and intra-protocol then the data flow will be fair, giving a better QoE to the user.

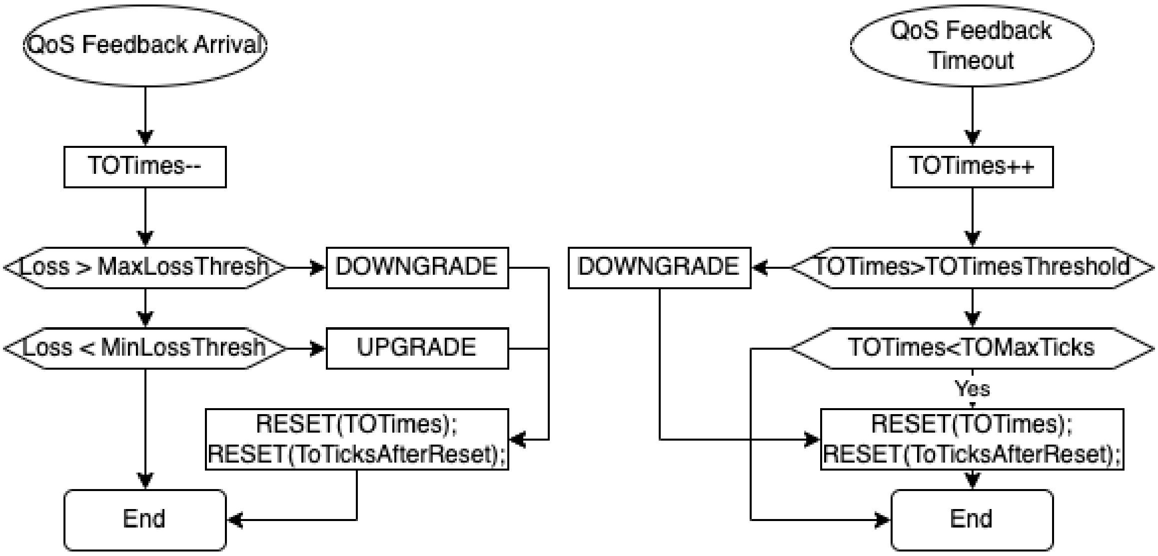


Figure 2.3: Adaptation Mechanism shown in [2]

2.2.1.2 Application Level Fairness

The application level is defined as a user-oriented level. This level is responsible for providing an interface to the embedded network services for the end-user. It is also responsible for making services such as file transfer, file management and information processing. The protocols that are involved and used at the application level include: Telnet, File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), and Simple Network Management Protocol (SNMP) [33]. This level also highlights the importance of QoS and QoE and how the application level makes the best effort to deliver the required features and expectations. Some researchers have taken the opportunity to study application-aware network management and try to enhance it in many ways. The research tried to achieve fairness of adaptive audio applications for multi-hop wireless networks. The authors highlight the idea that a redundant speech captioning scheme is necessary and when connections are adaptive in multi-hop networks perception, efficient bandwidth usage, and fairness are improved [2, 45]. An adaptation mechanism was introduced in [2], where a server takes the loss rate information and tries to adapt it. This was introduced because at high congestion feedback packets might be lost and become unable to reach their destination thus the timeout mechanism is useful to increase the probability of packets reaching the server. After testing the model it proved highly useful, it was found that in multi-hop mobile networks over WiFi, IEEE 802.11 successfully showed fairly distributed rates of transmission. Moreover, there were some scenarios where adaptive and non-adaptive connections were competing, thus the model was used and showed the end-to-end feedback effectiveness.

Some other research explored application-level QoS fairness in wireless video scheduling. It was argued that the initial delay for pre-fetching video at the client buffer should be shorter due to buffer limitations and application level user's QoE. A cross-layer optimized multi-user video adaptation and scheduling scheme for wireless video communication was proposed in [46, 47]. This research focuses on the video content delivered to each user and targets the quality of user satisfaction, video throughput was attempted to be maximized. Let the instructions that are sent through the available channel data rate at time k be $\lambda_{i(k)}$. Let the average video throughput be *t(k)* and Z as the total number of video requests from video streaming users. The scheduler decides on the target for the channel throughput at any given time with the source encoding rate shown by Equation 2.1 [46].

$$t(k) = \frac{k-1}{k} \cdot t(k-1) + \frac{1}{k} \sum_{i=1}^{Z} a_i(k) \cdot t_{i,l}(k) \tag{2.1}$$

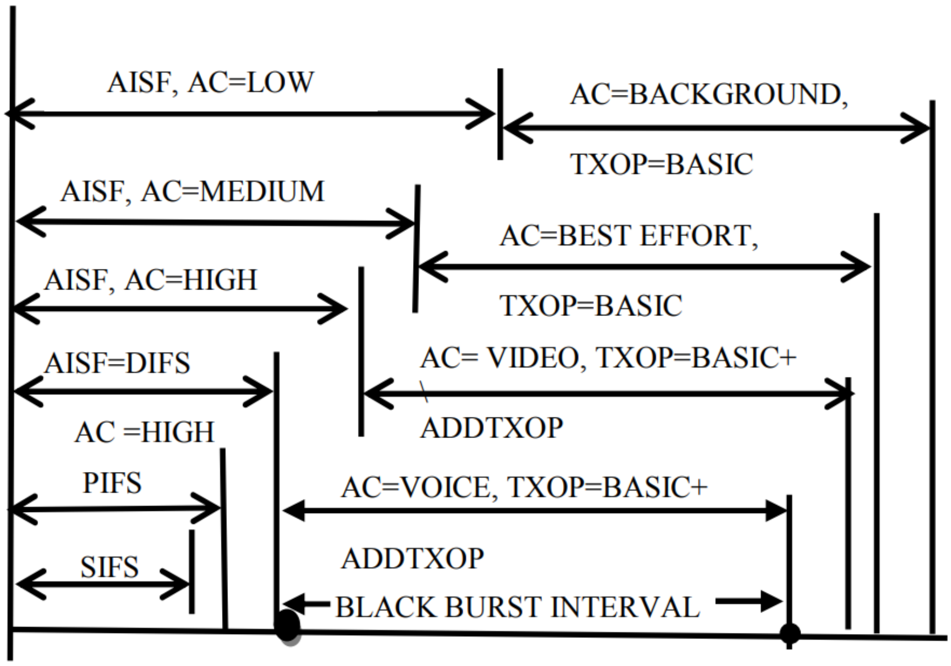


Figure 2.4: Improved Black Burst Access Mechanism shown in [3]

This scheme was used for video communication over packetised networks. It was tested in multiple environments to evaluate the QoE for users while playing continuous high-definition videos and monitor fairness. Using the scheme it was noted that the cross-layer shows a decrease in the undesired pauses in playback which increased the QoE for users. The authors also investigated the WLAN's user QoS for multimedia applications using MAC and Application layers, which they refer to as Media Access Control Layer and Application Layer Performance Assessment (MAP) approach. This is a hybrid approach to target the user's experience and achieve a multilayered approach. This investigation highlights research on two levels, starting with the MAC layer, improving the black-burst protocol [48] along with the prioritization of services.

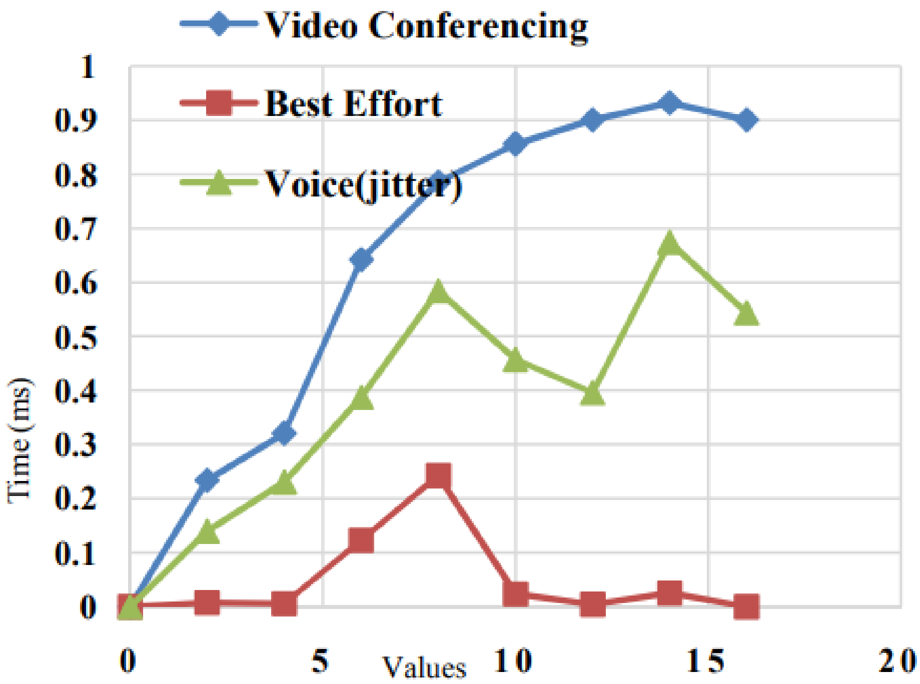


Figure 2.5: Fair resource allocation for Different Class [3]

Moreover Session Description Protocol (SDP) [49] and Real Time Streaming Protocol (RTSP) [50] are used to achieve QoS. Admission Control was used as well as it restricted excess service assignment over the network [51]. Maximum and minimum congestion windows had to be modified with a transmission approach for the first prioritized frame shown in Figure 2.4 [3]. After the technical adjustments, priority levels are also discussed. The authors recommended that after every priority assignment, the bandwidth will automatically updated to suit the situation of the end-user [52]. The graph shown in Figure 2.5 shows the fair allocation of resources based on the developed MAP approach. Moreover, the authors also investigated fair load share in a multi-source multimedia overlay in application-level multi-cast. They discussed the efficiency and scalability of IP multicasting and its ability to deliver high-quality video and audio streams for group communication. Since the spread in IP multicast is limited in the near future unless substantial infrastructure modifications take place, the authors proposed an Application Layer Multicast (ALM) scheme with its ability to support group communication independent of network-level support. The target was to achieve fair load sharing and overall communication quality among the participants in multi-source multimedia communication [53, 54, 55, 56, 57].

2.2.1.3 Human Level fairness

Human factors are essential to understand and maximise fairness. Studies have shown that fairness can be observed from multiple points of view. In this work's scope, it is essential to discuss the fairness perceived between humans and machines, and how it affects fairness from human to human, leading to a wider scope of vision and understanding of robot-to-robot fairness. Plentiful researchers discussed robot-to-robot fairness but this type of fairness can be viewed in countless ways, it could be simple machine data distribution fairness or allocation fairness of resources between devices equipped with Artificial Intelligence. Figure 2.6 shows the aspects that will be discussed in this subsection.

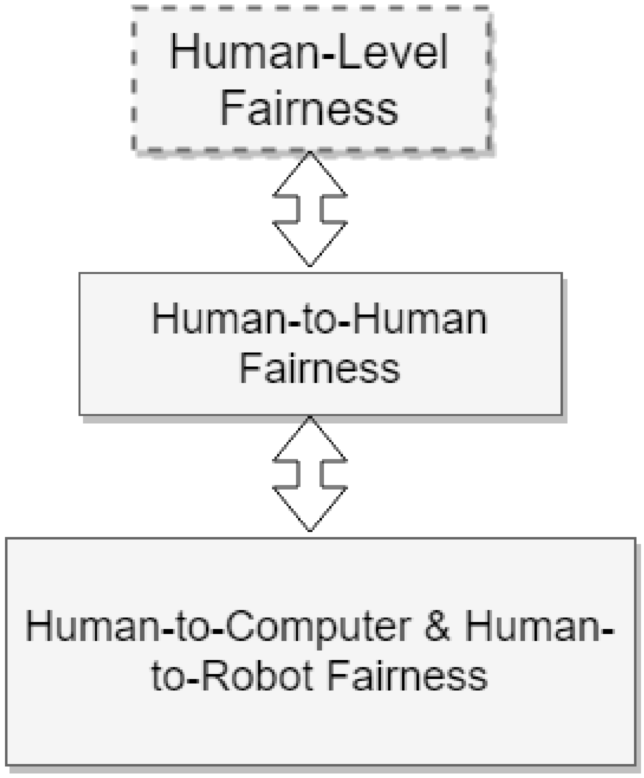


Figure 2.6: Human Level Affects, Relations and Research on Fairness

Human to Human Fairness; A multitude of scholars and researchers have discussed the importance of fairness, and how it affects social life. Fairness is quite complex because it can be referred to in countless scenarios. The central understanding and mature concept of fairness in humans is shown in the principles governing the allocation of resources, so-called distributive justice [58, 59, 60]. Psychological understanding of human behaviour of how people react to unequal distributions such as the absence of richer contextual information like neediness and deservedness is the key to unlocking human fairness [61, 62]. The concept of human fairness has dominated the research field of exploration of studies in disciplines, such as psychology, economy and anthropology. When examining technology, a clear resemblance of fairness and its attributes becomes apparent. Simply on networking and source allocation; host devices compete to get more resources, timeliness, and access to the Internet over a shared connection. The conclusion can be drawn that the fairness rules implemented in technology are founded upon or akin to fairness and justice in real-life social contexts. To be fair between humans and what they get in QoE along with QoS provided for them is to get equal resources from (for example) a governing body, and exponential growth in success with effort made by the human, in what they desire to achieve in life. Naturally, it can be observed that there is a clear link between fairness in humans and technology.

The ability to predict the provider's QoS and the user's QoE is highly important to this thesis's scope. With the correct additions and interactions, fairness can lead to the best possible QoE on a human-user level. Studies have shown that QoE management mechanisms are found usually in Application and/or Network levels where a process of mapping functions between QoS and QoE when optimizing the system occurs. Some researchers have used illustrative numerical examples to show that the choice of different types of metrics for quality estimation corresponds to optimizing the system for different types of users, leading to different QoE management outcomes such as optimal and fair QoE resource allocation [63]. There are different types of users and thus service providers must address each of their target audiences in a specific way to their needs and expected services to gain the maximum QoE from a user's point of view. A plethora of authors have journeyed to achieve fairness in the field of media technology, an interesting approach to this (in addition to what was mentioned previously in other levels) can be the adaptation of algorithmic logic for increasing QoE fairness for HTTP Adaptive Video streaming [64], where a simulative performance environment evaluation is conducted to compare the QoS and QoE fairness to achieve fairness between users or service providers. Previous studies have proven that QoE directly affects QoS provided by the service providers [65] and since that is true, then the most desired approach for human-to-human fairness is to maximize QoE while ensuring fairness among users.

Human to Computer & Human to Robot Fairness; To understand and view the fairness between humans and machines, HCI and HRI must be explained. Human-to-computer interactions (HCI) have emerged as the centre of computing research, moreover, it is highly useful for understanding and improving interactions with computer-based technologies. As time progressed with computer technologies, breakthroughs led to robotic technology, which naturally represents huge implications for the Human to Robot interactions (HRI). HCI can have a variety of modes including, (I) data; which is the communication stage of humans and computers, providing signs such as figures, colours and graphs. (II) Image; the ability of a computer to recognise images which is shown in image processing, recognition and perception. (III) Voice; is the combination of audio frequency and stored data to communicate. (IV) Intelligent reactions; computer's prediction of human actions based on the data given from their behaviors and needs Artificial Intelligence (AI). These interactions are examples of how HCI can be described [66]. Previous studies suggested that a significant aspect of collaborative environments and interactions is fairness [67]. In this scenario, there must be some form of an understanding and given properties that outline fairness to humans, computers and robots. A task can be given to any of these entities, however, with a fair point of view where each has its fair sources and tools, even the highest difficulty tasks can be accepted if all the collaborators are convinced that they are being treated fairly. Authors have also pointed out that in

HRI, some issues arise in the human's point of view. For instance, it is important to consider the fact that human perception of robots is different from other computer technologies, due to the idea that people humanise robots. Also, robots are able to learn about themselves and their surroundings and make their own decisions depending on the situation, this is a clear difference between HRI and HCI [68, 69]. HCI is a static and restrictive environment where it can only be controlled by a human entity, HRI is shown to be more dynamic with their interactions [70]. Internet of Things (IoT) devices can be described here as a form of computing environment. Resource sharing between human and IoT devices can be a combination of a static and a dynamic environment. Fairness between them must be considered as IoT devices can have machine learning algorithms that sort and distribute resources to humans and other connected devices. QoS and QoE must be achieved through techniques such as Max-min fairness for resource sharing to be achieved between those two entities.

2.2.2 Extended Fairness Flow Model

Figure 2.7: Extended FFM

2.2.2.1 Physical Level

The Physical Level of the FFM, as shown in Figure 2.7 is a hardware layer. It is described to be the domain of all physical media. Any type of hardware, cables, BNC (Bayonet Neill–Concelman) connectors, Hubs, and additional network adapters. Physical level protocols and detection of port signals are present to send and receive data from defined ports [33]. Researcher Xi'an studied physical-level fairness and it can be shown in his mathematical equation which first appears in [4]; devices such as relays, and eavesdropper devices were used to test the fairness of the system in a wireless communication environments. Fairness was viewed and analysed by investigating the expression of relay selection probability and numerical results.

Figure 2.8 shows the diagram that was tested which has a single source and a single destination with randomly distributed eavesdroppers. Assuming the relays are half-duplex, then it is also fair to assume that the transmission needs two slots to complete. Moreover, device Z starts the transmission to all relays,

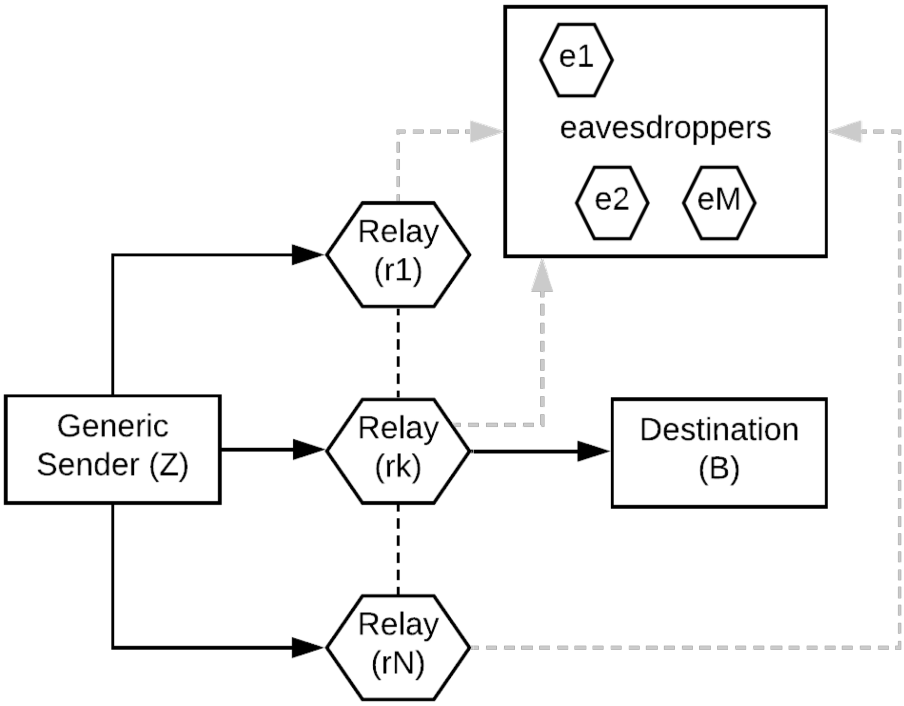


Figure 2.8: Relay Network with eavesdroppers [4]

a natural broadcast protocol of wireless communication. In the second slot, only the shown algorithm in Equation 2.2 which is the optimal relay can amplify the source information and transfer it to destination B. During that process, the eavesdroppers have live access to the transmission from all relay nodes and attempt to monitor the information. Following the work given in [71] and [4], Equation 2.2 shows $A_{(i,j)}$ which is defined for notation convenience being maximized, and this is due to minimizing the intercept probability.

Let's break down the parameters of equations 2.2 and 2.3 shown:

- k and $\tilde{r}$: These represent the optimal relay (in Equation 2.2) and the sub-optimal relay (in Equation 2.3), respectively. They are the solutions determined by the expressions.
- i and j: These are indices used in the expressions. i iterates from 1 to N, and j iterates from 1 to M. They are used to select specific elements from the matrices or arrays involved in the calculations.
- $A_{i,j}$ and $F_{A_{i,j}}(a_{i,j})$: These are functions that involve the elements of matrices or arrays.

  In Equation 2.2, $A_{i,j}$ represents an element in a matrix A at row i and column j. The outer function finds the maximum among the minimum elements in each row of matrix A.

  In Equation 2.3, $F_{A_{i,j}}(a_{i,j})$ denotes a function F applied to the element $A_{i,j}$, with $a_{i,j}$ as a parameter. The outer function identifies the maximum value obtained from the minimum value after applying the function F to each element $A_{i,j}$.

- arg max: This operation finds the argument (in this case, the index k or $\tilde{r}$) that maximizes the expression inside the subsequent brackets.
- min: This operation computes the minimum value among the elements specified in the brackets.
- max: This operation computes the maximum value among the elements specified in the brackets.

These equations are utilizing optimization techniques where they seek the best relay by evaluating specific functions and matrices to find either the maximum of minimum values or the maximum of transformed minimum values from a given set of elements over specified ranges. The actual meaning of these equations depends on the context and the specific interpretation of the functions and matrices involved.

$$k = arg \underset{i=1,...,N}{max} \left\{ \underset{j\epsilon \left\{1,2,...,M\right\}}{min} \quad A_{i,j} \right\} \tag{2.2}$$

[4]
(Optimal Relay)

$$\tilde{r} = arg \underset{i\epsilon \left\{1,2,...,N\right\}}{max} \left\{ \underset{j\epsilon \left\{1,2,...,M\right\}}{min} \quad F_{A_{i,j}}(a_{i,j}) \right\} \tag{2.3}$$

[4]
(Sub-Optimal Relay)

The Optimal Relay does not achieve fairness, even though it has a fair concept, there is a huge issue in the fairness of data, if $a_{(i,j)}$ (which denotes the instantaneous value of $A_{(i,j)}$) at one relay is always larger than that of the other relay which means in every case the array that has the smaller $a_{(i,j)}$ cannot be chosen, thus not a fair concept. To make it fair the same algorithm is used, and the cumulative distribution function (CDF) of $a_{(i,j)}$ is recorded. The final result would be the concluded algorithm in Equation 2.3 where $F_{A_{i,j}}(a_{i,j})$ denotes the CDF of $A_{i,j}$. This was a small example of fairness being applied to a relay network where fairness was created in a sub-optimal relay selection based on [4].

2.2.2.2 Session Level

The session level is responsible for the established connection, it explains the stages of a connection including; its establishment, control, and termination. Some fundamental session-level functions include (but are not limited to) communication link establishment, and maintaining a smooth flow of communication links during the session. Another important function states the ability of simultaneous communication leading to identifying interruptions in communications and working on solving them by re-sending the needed requests. Fairness at this level is dependent on the processes that occur within a session. Prior investigations have been implemented to show fairness within the Session level; One author in [72], studied a session-level resource control over 802.11 DCF Wireless LAN and implemented TCP-AV; which is an enhancement for TCP congestion control algorithm first introduced in [73], on a wireless network. It was suggested that the bandwidth reservation was 10% more than the fair-share value. Hence QoS can be achieved in a non-stable wireless environment. That is to say, unfairness was detected in the throughput of the TCP flow, thus it was deduced that the destruction of ACK (acknowledgement) by the buffer overflow affected the network directly making it mainly an issue of fairness in the session process, considering the Three-Way handshake process, taking into account the perspective of a researcher (Figure 2.9); a process that is usually associated with TCP handshake process where it is used to establish and tear down socket reliable connections over the network [74], meant here as an example of the session's establishment process [72].

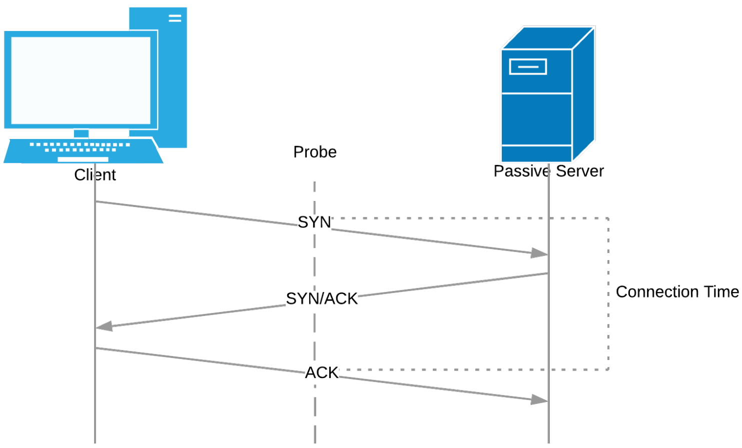


Figure 2.9: TCP Three-Way Handshake example [5]

Similarly, an author thrived to achieve Inter-Session Fairness for Layered Video Multicast. This study [75] shows a solution for the inability to provide fair bandwidth sharing between competing video sessions. Two main schemes were proposed; layered video multicast with congestion sensitivity and adaptive join-timer, along with layered video multicast with priority dropping to achieve inter-session fairness for layered video multicast. This was a very useful proposition due to the fact that there was still unfairness within the schemes, the authors realised that if users request a rate which is far more than the bandwidth available, the network would be consistently congested. Therefore many packets will be dropped before reaching the destination, and thus the two proposed schemes solve inter-session fairness when needed.

More studies also attempted session fairness, in a recent research, the study of lexicographically fair bandwidth allocation in multi-session layered media multicasting using network coding was looked at. Integer linear programming (ILP) was formulated around the fairness issue and the author proposed a polynomial time approximation algorithm to approximate the optimal solution of the ILP. Simulations for the algorithms were successful however ILP is not time efficient in large networks [76, 77]. Numerous multi-session and multi-layer fairness techniques can be improved upon but the session level is important from the fairness point of view since it is responsible for the entire communication that is needed for the session's establishment till its termination.

2.2.2.3 MPEG, QoE & QoS

Moving Pictures Expert Group (MPEG) Dynamic adaptive streaming over HTTP (DASH) is an adaptive bit-rate streaming methodology which has several abilities providing the best quality streaming of multimedia-related applications across the internet progression from traditional HTTP web servers. It works on the criteria of breaking down video content into small sequential data segments, which further worked over HTTP. Every segment has a small time duration of playback time that consists of multiple characteristics; like a short movie clip or the time broadcast of an event-like show or sports program. The availability of the segments is at multiple bit rates i.e.; a substitute encoded segments at a variety of bit rates surrounded by a small period of playback by an MPEG-DASH client. The automatically chosen highest bit rate segment, which can perform downloading on time for playback, will be the re-buffering occurrence of the playback. The MPEG-DASH can adapt to alternating network fluctuations and enable the best quality playback with a minimal number of re-buffering occasions. One of the main problems that the researcher is trying to solve is figuring out the correct classification methods for noisy data such as the output of a DASH media stream.

There is a considerable rise in the general quality of experience QoE anticipated by the users across multimedia distribution strategies like live streaming of video. Since the number of online client implementations and the equipment usage is increased, there is a significant change in the capability of the end equipment and facilities like network bandwidth are usually distributed among various end-user equipment. Therefore, single-device features can be gained; which are usually restricted due to the assets shared. Traditionally best-effort network construction assigns assets depending upon the request of the client and advance-level SLA without putting application and user-level necessities into consideration. The end-users are usually unsatisfied due to the perceivable unfairness. The concept of fairness within a network or between network resources can be interrupted in multiple ways and thus fairness can also be achieved in abundant different ways depending on the scenario [15].

QoS refers to the performance level of a telecommunications or computer network and its ability to deliver different types of traffic with varying requirements. In the context of MPEG standards, QoS plays a significant role in ensuring the delivery of multimedia content (such as video) over networks. QoE on the other hand, focuses on the end user's subjective experience while using a service or application. It encompasses a user's perception of various factors like video quality, smoothness, interactivity, and overall satisfaction.

In the context of MPEG, QoS is crucial for maintaining the quality of multimedia content being transmitted over networks, ensuring that video data is delivered with the requirements such as:

- 1. Bandwidth: Providing sufficient data rate to support high-quality video without interruptions.
- 2. Latency: Minimizing delays in video transmission to maintain real-time or near-real-time interaction.
- 3. Packet Loss: Reducing the loss of data packets to maintain video quality.

Improving network-level efficiency directly impacts QoS and subsequently aids QoE in various ways:

- 1. Resource Allocation: Efficient network management allows for better allocation of resources, ensuring that sufficient bandwidth and priority are given to multimedia traffic, which directly affects QoS parameters.
- 2. Traffic Prioritization: Effective traffic prioritization mechanisms ensure that time-sensitive video data is given precedence, resulting in smoother video playback and reduced latency, thereby improving QoE.
- 3. Reduced Congestion: Efficient network management helps prevent network congestion by optimizing traffic flow and minimizing packet loss or delay, thus enhancing the overall QoS for multimedia content.
- 4. Consistent Performance: A well-managed network with stable and predictable performance contributes to a consistent QoS, leading to a more consistent and satisfying user experience (QoE).

Improving network-level efficiency, by managing QoS parameters effectively, directly influences the end users' Quality of Experience by ensuring a smoother, higher quality, and more reliable delivery of multimedia content. It helps in meeting the specific demands of multimedia applications, especially those based on MPEG standards, leading to a more satisfactory user experience.

2.2.2.4 Research Gap

This thesis will discuss, a new quality analysis; which is used as a distant broadcast technique for measuring the appropriateness of the work situation for contributing to multimedia video evaluation. In a laboratory experiment, contributors (past researchers) achieved this quality analysis with various listening devices in various listening surroundings, involving a silent room allowing an imitation of circumstantial noise situation. Their results show important observations of the situation and the attending device on the quality inception, and their physical settings had a direct relation to user perception. Thus, the arrangement of this work's video trials will be free of sound and only target one aspect, the quality of the video perceived, in a virtual environment guided by International Telecommunication Union (ITU) Recommendations. The researcher aims to tackle the issues of subjective evaluation of objective QoE models and Adaptive Bitrate Algorithms (ABR). The researcher proposed an experimentation framework structure through programmable network management for the generation of ML Training-Ready Data and MOS/QoE Prediction. The thesis's testbed's data-generated analysis with real user experimentation and ML for training and predicting QoE based on the generated monitoring data was used. This way with the generated prediction, limited user testing is needed in the future, simply place the generated monitored data from the monitoring tools into the prediction model and it will generate a predicted MOS for faster and more efficient network-level tests and experiments. This model is unique because the data was tested using state-of-the-art ML algorithms by researchers, resulting in promising outcomes as shown in Chapter 5.

The thesis outlines several contributions that have been made in the field of SDN and multimedia technology. Firstly, a Virtual-Box environment has been developed that includes all the necessary libraries and applications required to run P4, Openflow, Python 2 and 3 instances, DASH, and Mininet. This environment has been designed to be error-free, providing a reliable testing environment. Secondly, a segmented content database has been created that includes six source videos, 120 test videos, H.264 encoding configuration at six levels, and is resolution-adaptive with fully configurable options. Thirdly, a P4 SDN testbed has been developed over Mininet, which enables the control of DASH initial buffering, stalling, switching, monitoring, bitrate adaptation, and bandwidth limitation over selected ports. The testbed also provides data insights into congestion for congestion-related experiments and is fully reconfigurable over the data plane and other mentioned features. Fourthly, a proposed experimentation framework structure has been created through programmable network management, which generates ML training-ready data and MOS/QoE prediction. Finally, a human experiment has been conducted with MOS-based feedback to benchmark the accuracy of predicted QoE and network features. Together, these contributions provide valuable insights and tools for researchers and practitioners working in the areas of SDN, multimedia technology, and quality of experience prediction. These contributions should aid in bridging the research gap of running simulations and with training-ready datasets that generate high accuracy for the network features aiding researchers in testing their experiments with this environment and predicting the QoE before running a physical experiment for a researcher to evaluate and conclude the necessity of building a physical experiment. This will then eliminate running unsuccessful human trials to save time and research efficiency.

2.3 Tools and Measurements

Network fairness can be measured in copious different ways, this section will explain the aspects and techniques used to measure fairness quantitatively to ensure maximum understanding of the nature of fairness and its properties. Quantitative research is a numerical-based research. It relies on numbers as the main unit of analysis, this type of research is highly used in countless scientific research fields. Quantitative results usually give accurate data based on measured algorithms, which is why it is a key category of techniques that must be discussed in the process of Fairness measurements [78].

2.3.1 Quantitative & Qualitative measurements

The researcher referred to objective metrics used to evaluate and analyze the performance of SDN systems. These metrics typically involve numerical measurements that can be analyzed statistically to draw meaningful conclusions about the behaviour and effectiveness of an SDN system. Some common quantitative and qualitative measurements in SDN include:

- Network performance metrics: These include measurements of network delay, packet loss, throughput, and jitter. These metrics can be used to evaluate the effectiveness of SDN-based congestion control and flow management techniques. [79]
- Traffic engineering metrics: These metrics include measurements of network utilization, link capacity, and load balancing. They can be used to evaluate the effectiveness of SDN-based traffic engineering techniques, such as dynamic routing and load balancing. [80]
- QoE metrics: These include measurements of the user's perception of the quality of a service or application. They can be used to evaluate the effectiveness of SDN-based QoE management techniques, such as adaptive bitrate streaming and congestion control.

Overall, quantitative measurements provide valuable insights into the performance and effectiveness of SDN systems, helping researchers and practitioners to optimize network performance and enhance user experience.

2.3.1.1 Jain's Fairness Index

Jain's fairness index [81] is one of the most recognized and used indexes of fairness measurements.

$$J(x_1, x_2, \ldots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \cdot \sum_{i=1}^{n} {x_i}^2} = \frac{\overline{x}^2}{\overline{x^2}} = \frac{1}{1 + \widehat{c_v}^2} \tag{2.4}$$

n represents the users and this equation rates the fairness of a set of values where x_i is the throughput for the ith connection and $\widehat{c_v}^2$ is the sample coefficient of variation. This index gives the result of 1 for the best case allocation and a $\frac{1}{n}$ worst-case representation [82]. This index aims to achieve an equal share of bottleneck; optimal allocation or an equal fraction of optimal allocation to users. It is scale-independent (applies to any number of users n), bounded between 0 and 1 where variance, standard deviation and relative distance are not bounded. Jain's fairness also shows a direct relationship of more generated fairness in a higher index and vice-versa [83].

2.3.1.2 Max-min Fairness

Max-min fairness is the process of applying a flexible sort of allocation. Max-min fairness can be useful in developing an intelligent fairness mechanism, this is since the Max-min allocation technique allows an increase in data flow and balances by creating a decrease in data flow in another allocation to create an equally balanced flow of data. To expand on this, multimedia applications can achieve max-min fairness in machine learning on a network level. To have the ability to recognize what the user is using the network for is essential to achieving multimedia fairness. Max-min fairness modal aims to distribute network bandwidth equally with infinitesimal increments to all flows until one is satisfied. With the right understanding of Max-min fairness and a machine learning model, multimedia applications can achieve fairness, which increases the QoE on an application level [21, 84, 85, 86].

2.3.2 Qualitative measurements

Qualitative measurement techniques are as important to this research as quantitative techniques. These type of techniques pays more attention to non-numerical aspects of measurements. A major qualitative technique is QoE fairness measurements. This technique focuses on considering the QoE as perceived by the end-user [87, 88, 89, 90]. This is important in cooperation where operators want to keep their users sufficiently satisfied in network management. Plentiful techniques were proposed especially for the field of adaptive video streaming [91, 92]. The most obvious technique is a questionnaire survey, distributed among end-users with their suggestions for increasing QoE fairness. QoE cannot be measured on ratio scales, thus previous quantitative measurements will not be applicable. One example to measure QoE can be an interval scale such as a simple 5-point mean Mean Opinion Score (MOS). Users with can rate the experience from 1 being the lowest to 5 being the highest quality on the MOS scale.

2.3.3 Alternative Metrics vs. Thesis's Measurement Techniques

HTTP adaptive streaming (HAS), such as MPEG-DASH splits a broadcast file into a multitude of segments. Every segment is encrypted into several bitrates to attain access for customers with changing stream specifications. The video segments are consecutively requested by the client, having maximum bitrate which is approximated by the network capacity to be sustainable. The process of ABR is through which a client suggests the ideal bitrate of the part to download. ABR is a traditional science technology. This rule contains default conditions, but a variety of them do not reflect the difference between the multiple scenarios that may occur in a production environment and often work poorly when the organization makes changes in the work environment. A recent attempt is to link the increase in ABR to the capabilities of the interpolation method [93], which is mainly based on a machine learning model. The ML-based ABR strategy is divided into two parts. In the first part, you can adjust the current ABR parameters. A calculation plan based on ABR variables is proposed. Support systems where ABR can change variables depending on the order in which the network conditions are changed. In deep learning, this ABR variable is based on an adaptation policy, where changes are presented in the context of the flow. In deep learning, depending on the strategy used in which tuning of the parameters of ABRs, would directly affect the streaming content.

Previous research states that bitrate depends upon ABR models having trained predictive collection of decisions (SMASH [94]). Where a 'combine grouping' scheme was used to make a map network-related properties of bitrate. A supervised ML-based ABR was implemented with features related only to the bitrate status. Therefore, there are some restrictions with those predictions as features are focused on limited network factors. Both the conception of engineering and the ML algorithm selections will not be executed methodically, moreover, the trained model has planned to support, rather than replace, the existing fixed rules that depend upon modifying the algorithm. Moreover, the recently introduced model-free strategy is said to be Pensieve [95], which uses a reinforcement learning strategy to introduce a neural network based upon ABR. This strategy makes no explicit assumption related to effective data. Multiple papers have reported issues like Pensieve, therefore it is being accompanied by implementing detailed experimental evaluation of Pensieve, keeping various sets of video content and network go over under observation [96]. In [97] within the training process, the results change significantly when using a web account, the deficit has a high value and does not tend to coincide. They have a high bitrate presented in their video setup. For example, when running UHD and 4K data packets, authors choose reasonably bright screens based on past learning success, indicating that the data is fragmented. A bitrate equal to the brightness percentage is reduced by 50%, the experimental model only learns to achieve maximum results, and the bitrate leads to an inaccessible video level. The heterogeneity of wireless networks means that larger variants continue to be known, and as the value of video resolution continues to grow, so does the popularity of data. This work was therefore encouraged. For research and experimentation, researchers spend an enormous amount of time in the creation of their test environment, most of these environments are quite specific to the target of their research. There is a plethora of downloadable virtual environments such as P4 or SDN-based virtual testbeds however, these environments are very specific to their purpose. Configuring a suitable testbed that can run multiple solutions from multiple different testbeds is very time-consuming. Thus researcher created a Virtual-Box Environment with all the necessary libraries and applications needed to run P4, Openflow, Python 2 and 3 instances, DASH and Mininet. This provides ease for researchers to use the thesis's virtual machine setup to dive straight into testing and data generation without wasting their time and effort building the virtual setup.

Table 2.1 shows a list of previous HTTP Adaptive Video Streaming Databases that are widely used in research within QoE. The thesis's generated Database of encoded videos contains 6 source segment division videos and 120 different resolution videos. This database comes with a pre-configured P4 Software Defined Network with a server and multiple clients for testing and evaluating QoE with DASH Reference Player.

Its main contribution is that it monitors the network and all its ports for recording DASHIF Reference Server packets and evaluating client's data. It then generates training-ready clean data for ML usage purposes. This was done to ease the experimentation with big data and data mining for use in ML and the understanding of multimedia network environments.

Furthermore QoE is widely discussed and predicted in multiple aspects within the use of different features and multiple joint classifiers [98, 99, 100, 101, 102]. This thesis's approach is unique in its data generation. The generation of network data required for our predictions relies on this work's testbed. Using this network data, machine learning algorithms are employed to predict the generation of MOS and other features. Additionally, an analysis of the network data is conducted to identify the most influential feature directly impacting MOS. Subsequently, this feature is prioritized in the thesis's testbed. Thus the data generated from the test bed is always training-ready to test on more algorithms.

Database	Source Videos	Test Videos	Encoding Configurations	Test case Formation	HAS-related Impairments	Resolution Adaption
LIVEMVQA [103]	10	200	H.264 at 4 levels	hand-crafted	switching or stalling	No
LIVEQHVS [104]	3	15	H.264 at 21 levels	hand-crafted	switching	No
LIVEMSV [105]	24	176	no compression	hand-crafted	stalling	No
Waterloo SQoE-I [106]	20	180	H.264 at 7 levels	hand-crafted	switching	Yes
LIVE-Netflix Video QoE Database [107]	14	112	H.264 at 6 levels	hand-crafted	initial buffering & Stalling & Switching	No
Waterloo SQoE-III [108]	20	450	H.264 at 11 levels	simulated	initial buffering & stalling & Switching	Yes
ITEC DASH [109]	7	131	H.264 at 6 levels	hand-crafted	initial buffering & stalling & switching	Yes
The researcher's Dataset	6	120	H.264 at 6 levels	simulated	initial buffering & Stalling & switching & Monitoring	Yes

Table 2.1: Comparison of publicly available QoE Dataset for HTTP-Based Adaptive Video Streaming

Using QoE, the researcher will talk about the three primary approaches that can be taken to determine the best way to adapt: techniques that are network-based, techniques that are client-based but include assistance from network elements, and client-based approaches that include assistance from network elements. A client is responsible for doing quality adaptation in the purely client-based technique. This adaptation is based on the client's local factors, such as network throughput, occupied buffer size, and so on. The second method proposes that these clients can be helped by the information provided by elements of the network such as proxy servers. Within the context of the third way, the researcher makes use of a central network component to carry out rate modification on the customers' behalf.

A. Client-based adaptation

As illustrated in [110] and [111], DASH video players in clients are responsible for quality adaptation to optimise the QoE objectives. These objectives include, among others, the minimising of initial buffering time, the minimization of stalling, and the maximising of quality. The authors of [112], [113] provide a broad framework for bit rate adaptation that is comprised of a collection of approaches with the goal of achieving a balance between video stability, fairness, and efficiency. This framework can be found in [112], [113]. To establish a biassed interaction between the bit rate and estimated bandwidth, a stateful bit rate selection heuristic technique is utilised. This algorithm is used to achieve the goal. Nevertheless, this client-based quality adaptation results in unequal competition for bandwidth, which becomes more severe when a large number of clients use the shared network resources [114]. In addition, quality of experience fairness is not produced by resource fairness, particularly in environments with a heterogeneous composition [115]. Because of this unfair resource utilisation and unanticipated network traffic spikes, clients may experience frequent variations in the perceived video quality, which is referred to as "client quality oscillations." This has the effect of drastically lowering the quality of their overall experience. As a result, an optimization model needs to be built to strike a balance between the various QoE criteria, such as the maximum of video quality and the minimising of variations in bit rate, while simultaneously satisfying network restrictions. To accomplish this objective, a number of distinct algorithms have been devised through the careful examination of a variety of criteria including things like predicted network bandwidth, network throughput, and received network feedback signals (such as the incidence of congestion) [116], [117]. However, it is abundantly evident that, in the absence of a central controller for clients and through the use of entirely client-side sub-optimal adaptation decisions, there will be frequent shifts in the video quality, which will ultimately result in a decrease in QoE [110], [118]. Additionally, with a central controller, it can implement multiple management policies, such as different subscription policies (for example, Gold, Silver, and Bronze) [119]. This is made feasible by the fact that it is possible to differentiate between different levels of subscriptions.

B. Client-based adaptation assisted by Network Elements

As was noted earlier, certain aspects of the network can assist customers in the process of client-based rate modification. A client can improve its quality adaption method by making use of the information offered by the elements of the network [120]. Help client-side quality adaption in an AVC DASH streaming technique by utilising an SDN controller with a comprehensive view of the network. In addition, the authors of [121] suggest an SDN-based video streaming strategy as a means of equitably maximising the quality of experience for several competing clients operating within the context of a shared network. Similarly, the SDN is the foundation for the suggested method in [122]. The authors of [122] use two primary strategies to optimise the quality of experience in terms of the number of quality changes as well as fairness. In the first method, the video quality of the clients is decided by the SDN controller. The second one involves the creation of a queue for each client in order to allow dynamic rate adjustments. Even if these studies make use of certain components of the network to enhance the quality of experience, consumers determine on their own which adjustments are most appropriate. As a result, the use of these solutions does not result in an effective sharing of network resources. Bentaleb et al., in their paper [114], address scalability difficulties with HAS, such as video instability, inequity in quality of experience, and underutilization of network resources. They do this by providing the highest possible quality of experience to each individual customer. In addition, [16] eliminates the most significant shortcomings of SDNDASH [114], which include scalability, communication overhead, and the inability to accommodate heterogeneous client configurations.

However, there have been certain works that have primarily focused on QoS-aware video traffic routing [123], [124], and [125]. This research has explored the QoS-aware video flow routing in networks that are equipped with OpenFlow and SDN. After that, taking into account the QoS priorities, the routing algorithm determines the shortest path that may be taken between the media server and the client. In addition, Egilmez and coworkers [124] create a QoS-aware controller that is capable of delivering multimedia flows in networks that are equipped with OpenFlow. In this method, the incoming traffic is categorised according to the different types of data flows. After that, multimedia flows are delivered to the destination over QoS-guaranteed paths, while other flows are sent there by the shortest possible paths. These studies don't go into detail on the different quality adaptation approaches. Although these strategies have the potential to improve client-side decisions, the decisions that result from using them are still not ideal because no one core element is responsible for decision-making that is independent. In addition, customers may not be capable of carrying out the judgments that were made by the controller [122]. As a result, quality adaption based on the core location might be suggested.

C. Network-Based Adaptation

[126], [127], and [128] are references to publications that make use of proxies or controllers as the central decision-making authority for rate adaptation. The authors of [126] offer a method that makes use of a proxy server to monitor users' requests for a particular video quality. In this method, a proxy server will, at regular intervals, work through an optimization challenge to ascertain the highest possible quality levels of segments that the clients will be able to download. This is accomplished by taking into account the present state of the network in addition to a predetermined objective function. In point of fact, whenever a client-based adaptation results in a reduction in the overall level of fairness, the proxy server can replace its choice of the video quality with that of the client. As a result, a particular quality of customer experience can be assured for some or all customers. In addition, Mok et al. [127] create a proxy architecture with

the goals of preventing frequent and extreme oscillations and enabling a predetermined amount of change in video quality at each phase. [128] This method, in contrast to this work, is not scalable since all of the network traffic destined for the video server must first be routed through a proxy server. In addition, strategy types such as reactive and proactive QoE improvement are discussed further in [128]. Utilizing information about the network, client-based quality modification is accomplished through the use of the reactive technique. The second method provides clients with a video quality that is both superior and more consistent across the board since it takes into account client buffer consumption and quality adaption in the controller.

The thesis's open-source database for machine learning prediction that contains 6 source videos, 120 test videos, H.264 encoding configurations at 6 levels, simulated test case formation, HAS-related impairments such as initial buffering, stalling, switching, and monitoring, along with resolution adaptation offers a range of benefits. Firstly, an open-source database is readily available to the research community, allowing researchers to replicate and build on existing work and share their own contributions. Secondly, the inclusion of source videos and test videos with different encoding configurations and resolutions provides a comprehensive dataset for researchers to test and evaluate their machine-learning models. Thirdly, the simulated test case formation allows researchers to create controlled experiments and analyze the performance of their models under different conditions. Fourthly, the inclusion of HAS-related impairments, such as initial buffering, stalling, switching, and monitoring, allows researchers to evaluate the performance of their models in a real-world context. Finally, the resolution adaptation feature allows researchers to test the adaptability of their models under changing network conditions, which is an essential aspect of machine learning-based QoE prediction. Overall, an open-source database with these features provides a valuable resource for researchers and practitioners working in the field of machine learning-based QoE prediction.

2.3.4 Chapter Summary

SDN is a promising technology for supporting multimedia applications, such as video streaming, which require high-quality experience. Fairness and QoE are important issues in SDN, as unequal allocation of resources can lead to poor QoE. Training-ready databases is used to predict QoE. Fairness and QoE are important considerations in SDN and multimedia technology and can be predicted using machine-learning techniques and databases. Media fairness can be improved by creating new algorithms to maximize user experience and satisfaction.

Fairness is a term used in literature, technology, and media streaming to achieve equality, justice, quality of distribution, and fairness in target audience satisfaction. Locating and targeting unfairness is essential for achieving fairness in a wireless network. Short-term fairness is more important than long-term fairness due to its ability to dynamically update every 20 minutes and achieve long-term fairness in the long run. It can also solve problems such as large requests for data and lower data received by other devices.

Inter-Session Fairness for Layered Video Multicast is achieved through two schemes: congestion sensitivity and adaptive join-timer, and lexicographically fair bandwidth allocation using network coding.

MPEG DASH is an adaptive bit-rate streaming methodology that provides the best quality streaming of multimedia-related applications. It can adapt to alternating network fluctuations and enable best-quality playback with minimal re-buffering occasions. This thesis proposes an experimentation framework structure to generate ML Training-Ready Data and MOS/QoE Prediction to tackle the subjective evaluation of objective QoE models and ABR. These contributions provide valuable insights and tools for researchers and practitioners working in SDN, multimedia technology, and quality of experience prediction.

Quantitative research is a key category of techniques used to measure fairness quantitatively. Quantitative measurements provide valuable insights into the performance and effectiveness of SDN systems, helping researchers and practitioners optimize network performance and enhance user experience. Jain's Fairness Index and Max-min Fairness can be used to achieve fairness in multimedia applications, increasing QoE. Both indexes are scale-independent and bounded between 0 and 1. Qualitative measurement techniques such as QoE fairness measurements and HAS are important to measure QoE fairness.

ML-based ABR strategies can be used to adjust the parameters of ABRs and directly affect the streaming content. Research has shown that bitrate depends on ABR models having trained predictive collection of decisions, and Pensieve is a model-free strategy that uses reinforcement learning to introduce a neural network based upon ABR. A Virtual-Box Environment has been created to provide ease for researchers to test and generate data.

CHAPTER 3

Design Principles & Testbed

"Good design principles are essential in the preparation of any technology testbed. Attention to detail, consideration of potential use cases, and a thorough understanding of the underlying technology are all critical factors in creating an effective and successful testbed environment."

Ahmed Al-Mashhadani

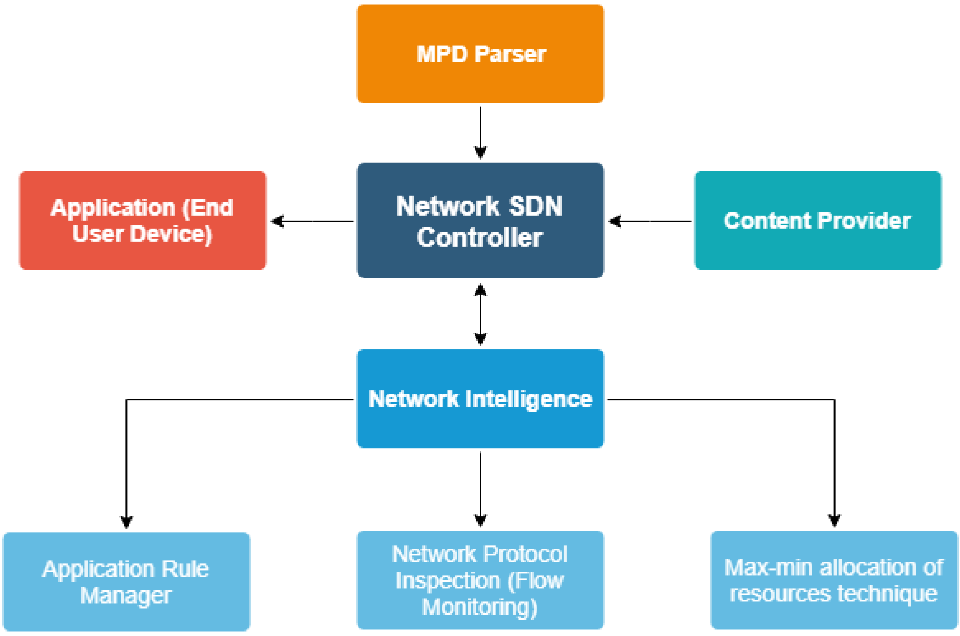


Figure 3.1: Proposed Fairness Theoretical Framework Structure

3.1 Introduction

The purpose of the current study was to determine the definition of fairness on different levels and points of view in multimedia and how it can be affected by and achieved through networking mechanisms. The most important finding to emerge from this study is the importance of human-level fairness as an addition to the resulting Fairness Flow Model in Figure 2.2. The Human aspect and point-of-view on multimedia fairness is quite important because it has a direct effect on all the other levels in the FFM. The generalisability of these levels is subject to certain limitations. For instance, future research must expand the network-level fairness within the FFM with an addition of hardware-based level fairness. Future research will include an investigation on Physical, and Session-level fairness and their perspective on SDN algorithms in multimedia. An additional proposition is to create a plug-and-play test bed and use SDN to program a network controller to understand users' needs and requests by applying Max-min fairness strategy after thorough research on the best language to base the test bed on, such as; OpenFlow or P4. This will allow a network controller to distinguish between users' needs, where if one user requires more media network resources, then the controller will grant the user its needed bandwidth while decreasing the bandwidth for another user who does not need its entire bandwidth. This will create the best QoE possible for both and all users.

Figure 3.1 shows a proposed framework structure for achieving fairness in the use of multimedia applications with SDN.

- Framework Inputs; in this framework, the network protocol inspection using the flow manager (an

SDN application) along with the Media Presentation Description (MPD) Parser help the controller identify the number of devices in the network that are requesting streaming bitrates and the MPD files requested by the users. This information is essential to the SDN controller as it will use the Application Rule Manager to filter the provider's content for security measures. This process will occur as the requested content provider reaches the SDN controller. The controller at this stage gains the user's media duration and the requested encoding bitrates.

- Framework Functions; in addition to the filtering that occurs above, the flow manager will ensure the accurate flow to the switch. This affects the decisions made by the framework's intelligence aspect because it makes sure that all users receive the requested video stream from the provider, through the controller.
- Framework Intelligence; using the max-min fairness allocation of resources technique, MPD parser, and application rule manager will not only identify the network traffic but will also distribute the traffic and bitrates fairly and securely to the user.

Understanding cross-device and cross-user fairness has become as crucial as the QoE on individual user devices. This section discusses different levels of fairness considerations in multimedia applications. The thesis also discussed how perceivable fairness is linked to resource allocation and traffic engineering at the network level and how emerging programmable networks such as SDN can be used as a tool to improve fairness. A cross-layer fairness framework is also proposed to harness the capabilities of new network designs and the growing availability of computing resources in future networks for fairness-aware content distribution.

The FFM showed fairness on three different levels to further the understanding of fairness's definition in multiple scopes. This thesis discusses a collection of SDN research focused on the fairness of network usage through multimedia applications in single/multi-home based in a smart router/switch environment. Monitoring and understanding network fairness are the main aspects of achieving network-level fairness on a human level thus quantitative techniques of measuring fairness were researched and investigated to aid in the development of a possible future plug-and-play multimedia networking test bed. Expansion of the FFM was proposed to be researched along with the creation of the SDN controller defined in the framework presented, to achieve a smart-source allocation network controller to develop a fair network structure that supports multimedia applications.

3.2 Design Principles and Requirements

Understating the FFM design principles and requirements is essential to gain the most efficient services aimed at user satisfaction. The following subsections will highlight the main design principles of the FFM and its requirements. Moreover, performance requirements will be provided for a better understanding of the functionality of the technical experiment.

3.2.1 Design Principles

The approach presented in this section makes use of different configurations of neural networks and classifications arranged to provide the best-fit feature classifier and MOS prediction suitable for most of the tasks characterizing modern media streaming which is the specific goal of this work. Moreover, the thesis focused on the design and implementation of the thesis's P4 testbed to host the DASH Reference player and the ability to monitor it, whereas another thesis can easily extract the data and use the ML techniques that this thesis shows in the next chapter to test and improve upon this work.

Multimedia data file is segregated into multiple parts or segments and then conveyed to the user using HTTP. An MPD explains the particulars of the segment, the particulars of the segments include aspects like time, website, and multimedia properties such as video resolution and bit rates. These segments can be arranged in several methods like Segment Base, Segment TimeLine, segment Template and segment List, relying on the use-case. Segment can be a media file of any type or format, such as the "ISO base media file format" and MPEG-2 Transport Stream; there are two major kinds of container format. DASH can be considered as a Video/Audio codec sceptics. Media files are usually provided as a multiple number of illustrations and the concerned choice of data is mainly related to the network status, equipment potential and client preferences that are responsible for allowing the Adaptive Bitrate Steaming and impartiality of the Quality of the Experience.

The adaptive bitrate streaming logic is not defined by the MPEG DASH standard. Therefore, DASH can be implemented on any type of protocol. PCC expressively decreases the storage quantity at the cost of multifaceted pre-processing and execution at the client. HAS deals with dynamic setup circumstances while endeavouring to transport at the maximum quality possible in the given circumstances.

3.2.2 Objective Metrics

The basic remodelling of the automation is found in the media streaming that includes best quality on-demand data and live media content. In the present scenarios, the main interest is to attain the pre-eminent quality of service and experience because of the ever-aggregating network consumption and user demand. The traditional type of streaming methodologies confronts multiple trials in distributing multimedia content to the end user without lowering the quality of the service. Adaptive HTTP streaming is the ever-increasing content-providing tactic; which delivers real-time content without negotiating quality in decision-making and guarantees an excellent quality of experience. The selection of bit rate must be

Codec	Bandwidth of Activation	Resolution
avc1.64001f	3134488 bps	1024x576
avc1.64001f	4952892 bps	1280x720
avc1.640028	9914554 bps	1920x1080
avc1.64000d	507246 bps	320x180
avc1.640015	759798 bps	480x270
avc1.64001e	1013310 bps	640x360
avc1.64001e	1883700 bps	768x432
avc1.640033	14931538 bps	3840x2160

Table 3.1: Video Database Encoding Information

effective and durable, it depends upon the nature of the network and thus thesis argues that it must always be dynamic. The client's key potential point-outs towards gaining the outstanding quality of service towards the end user, is to achieve an increase in the range of standards and procedures introduced in the Adaptive HTTP streaming field. Researching and contrasting is compulsory for executing numerous techniques depending upon predefined merits. The HTTP Live Streaming, Microsoft Smooth Streaming and MPEG DASH are arising model techniques of adaptive HTTP streaming. To calculate the transporting execution, the experiment moves forward using G-streamer adaptive HTTP streaming. To calculate the transporting execution for achievement; is based upon changing multiple networks and situations for on-demand streaming and live streaming content. The adaptive HTTP streaming method's resultant is calculated and examined using pre-explained performance indices. The processed data depicts that each entertained method of delivery and best conductance is achieved by the predefined advantages. In short, DASH provides appreciable balanced performance throughout multiple network arrangements as compared to other streaming approaches.

The experiment is staged based on the encoding information in Table 3.1. Users will watch a video based on the video segmentation and resolution properties mentioned in Table 3.1, all videos are fixed on 30 frames per second on all video streaming qualities, with a minimum buffer time of 2 seconds of loaded content. Users will watch five 40-second trials of video content on a limited bandwidth of 0.5 Mbps, 1 Mbps, 3 Mbps, 5 Mbps, and unlimited settings. With every trial, the user will input a Video MOS (vMOS) rating based on their video experience of initial load delay, resolution change, and overall quality of experience.

3.2.3 Subjective Evaluation of DASH

In principle, the aim is to evaluate the relevant QoE parameters and variables that take into account the kinetic properties of the video. Common objective indicators of a subject's performance to regularly determine their relevance to human perception are essential for evaluation. There is no subjective evaluation for DASH Adaptive streaming, for justifying longer video patterns; which are enough to explain the bitrate switching for the data-set that acquires the longer segment videos to various network conditions sequences. Estimation of the end user's real-time Quality of Experience (QoE) online by exploring the apparent influence of delay, diverse packet loss rates, unstable bandwidth, and the apparent quality of using the altered size of DASH video stream segment over a video streaming assembly under multiple video arrangements is quite possible with this sections's approach. The performance and potential of the system and the prospects of the end-user depend upon the Mean Opinion Score. The subjective evaluation of DASH gives an overview of impairments with various networks and various video segments on different end-users. For the test setup and procedure, the thesis used the most recent ITU-T P.913 titled: "Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment." [129] as general guidelines. A screen with 4K resolution was used to passively stream the content and record its segmentation. Which was used to compile as a separate video and present to the user to eliminate Virtual Environment computational power limitations from affecting the video and its MOS rating, The thesis recommended sitting at a distance of approximately four times the height of the screen used by users. The applied test protocol was as follows: Firstly this work started with the Welcome text-based information; Briefing and informed consent. Then this work moved on to the explanation and recommendations of the Setup as recommended by the ITU-T; Screening and demographic information. Afterward, the content of this work was displayed; 5 video samples based on different networking runs. This work's Evaluation stage was next, which was a collection of 3 QoE-related questionnaires for each video sample. Ending with the Debriefing which is entitled the Feedback and remarks from the user-end. This way the MOSes from the users and the recorded network data of their experiment to used for the works's prediction mechanisms later. Equation 3.1 defines MOS where R are the individual ratings for a given stimulus by N subjects. The MOS ratings were defined into 5 different classes before uploading the rating data to the training process as shown in Table 3.2.

$$MOS = \frac{\sum_{n=1}^{N} R_n}{N} \tag{3.1}$$

MOS	Definition	Description	Class
1	Bad	Dissatisfactory Perceived Quality	1
2	Poor	Unsatisfactory Perceived Quality	2
3	Fair	Acceptable Perceived Quality	3
4	Good	Satisfactory Perceived Quality	4
5	Excellent	Highly Satisfactory Perceived Quality	5

Table 3.2: Mean Opinion Score Scale

3.3 Case Study and Testbed Setup

Software Defined Networks are the decade's most important networking advancement. Packet Inspection (PI) technology improves network security and administration, but when integrated with SDN, it may centralise network strategy control and speed up automation. The transition from traditional to SDN networks is a considerable difficulty. The study prioritized improved Deep Packet Inspection (DPI) that can monitor the SDN controller along with the network and its traffic flows. This allows SDN to view the network as a whole, not a collection of devices (e.g. switches, security and other Layer 4-7 elements). Connecting SDN with DPI will help network pros control and automate the full network, not just parts. This is a technique used to analyze network traffic at the packet level to extract detailed information about the content and structure of the data packets. DPI goes beyond basic packet header analysis and examines the payload or content of the packets to gain insights into the applications, protocols, and services being used on the network. Using a central DPI capability will supply knowledge and intelligence to every critical function (security, controller, policy, etc.) So, inspecting SDN network-related feedback for central control became necessary.

3.3.1 Packet Monitoring Algorithm

The study begins by determining the network path or segments for media data transport. We'll employ SDN-specific approaches to test and validate the network path used to transfer data. Before transmitting user data, the work can use invasive techniques like Pathlet Routing [130], Pathlet Tracer [131], and ICING Path Verification Mechanism [132] to ensure that the SDN controller's configured network path will be used. The use case will dictate which approaches can be utilised before, during, and after transmission. During transmission, the study uses alternative approaches like SDN TraceRoute [133] that have a minimum impact on bandwidth and processing and use the switch's flow rules to confirm that we're using the indicated route. The study utilises the above tools to analyze segment data: 1. Data Flow Analysis -

Data path flow assessment compares pre-flow, actual-flow, and flow switch-by-switch and the entire route. 2. Packet/Flow Size - Data flows from switch to switch, comparing average packet size. 3. Packet/Flow Duration - Measured from the first packet entering the switch plane to the last packet exiting. 4. Average Packets per Flow - study computes a median value for this [134]. 5. Switch Latency - The time a flow spends in the switch and switch plane, from switch to switch, helps identify if several segments use the same flow table entry (i.e. many flows, but few packets).6. Switch/Device Details - Knowing the type of switch, topology level, location, and other device characteristics may increase network usage. 7. Median Bytes per Flow - This often small number indicates congestion (for example TCP flooding). The work can estimate path dependability and congestion using a median value [135]. All of this helped in the construction of the network inspection algorithm tool. This utility created training network implementation data.

Algorithm 1, "Monitor Network Packet," is designed to extract network traffic from packets and monitor network hosts for potential bandwidth change that can affect user experience. The algorithm takes input parameters, including source and destination addresses and ports, protocol, and packet information, and outputs the extracted network traffic. One of the primary benefits of this algorithm is its ability to detect resolution changes by monitoring network hosts and extracting network traffic from packets. Additionally, the algorithm can be configured to set thresholds for requests per minute and packet count, which can trigger alerts or actions if exceeded, providing an additional layer of control for the network. The algorithm's ability to write data to a file also allows for the collection of data over time, which can be used for network analysis and optimization. Overall, the "Monitor Network Packet" algorithm is a valuable tool for network administrators looking to monitor and control their networks for changes in network features that could affect QoE.

Algorithm 1: MONITOR NETWORK PACKET

Input: SourceAddress (S_a); SourcePort (S_p); DestinationPort (D_p); DestinationAddress (D_a); Protocol (T_i); Time (T_t); Packet IP Destination (P_d); Packet IP Source (P_s); Packet Transport Layer (P_t)

Parameters RequestsPerMin (R_m); PacketCount (P_c);

:

Result: Extract Network Traffic from packets.

Procedure 1. *Monitor Host 1, 2,...etc* **by**
  if $(P_d) = trueh1$ *or* $(P_s) = trueh1$ *or* $(P_d) = trueh2$ *or* $(P_s) = trueh2$ *or...* **then**
    $T_i = P_t$; $S_a = P_s$; $S_p = P_tP_s$; $D_a = P_d$; $D_p = P_tP_d$; $T_t = ConstantStartTime$
    if $(P_s) = trueh1$ **then**
      $Writein$*”data.txt”*
    end
    if $(P_s) = trueh2$ **then**
      $Writein$*”data.txt”*
    end
    if $(T_t) = 0$ **then**
      if $readFromFile() = 2$ **then**
        check D_s (get R_m ()); print S_d S_p D_a D_p T_i T_t; *set* R_m = ”1”; $writeToFile$(”3”)
      end
      if $readFromFile() = 21$ **then**
        print S_d S_p D_a D_p T_i T_t;*set* R_m = ”1”; $writeToFile$(”3”)
      else
        print S_d S_p D_a D_p T_i T_t; $writeToFile$(”0”);*set* $R_m = string$ *set* R_m +1
      end
    else
      if $readFromFile() = 0$ **then**
        check D_s (*get* R_m); print S_d S_p D_a D_p T_i T_t; *set* R_m (”1”)
      else
        print S_d S_p D_a D_p T_i T_t; $writeToFile$(”2”); *set* $R_m = string$ *set* R_m +1
      end
    end
  end
End

3.3.2 Case Study - Network Monitoring & Prediction

Achieving a positive quality of experience in a network is quite difficult. Network efficiency plays a crucial part in network and user fairness. Efficiency can affect the network and the user in both positive and negative aspects. First and foremost thesis must identify what is important to achieve from this testbed. Network Fairness can be achieved by monitoring the network data and acting according to the network's functionalities.

1. Network Level Monitoring: It is fundamental for the thesis's project to be able to know what is going on on a network level. Thus a network monitoring tool was created for the sole purpose of monitoring media video streams.

2. Quality of Experience Prediction: It is important for the experiment to predict the network and user's quality of experience. Based on the network-level data feedback provided by the monitoring algorithm, the prediction tool must make a decision based on tested thresholds to show if the network has a better or worse quality of experience prediction. This is important as the thesis's target is to improve the network efficiency and the user's quality, thus the tools that will affect the network resources and their allocation will be based on the QoE Prediction algorithm and the Network Level Monitoring Algorithm tools.

3.3.3 Testbed and Experiment Configuration

Test-bed in Figure 3.2 was used to run different but related TCP and UDP tests to investigate the problem domain of this topic. The figure shows a red-coloured link between the three switches, this is to show the nature of the first test taken. Using this test bed, the thesis ran 57 test runs, which included in total; of 189 generated data sets with 54,828 rows/records including interval, transfer, bandwidth, IP of sender, IP of receiver, protocol, and port as features/columns of the test runs.

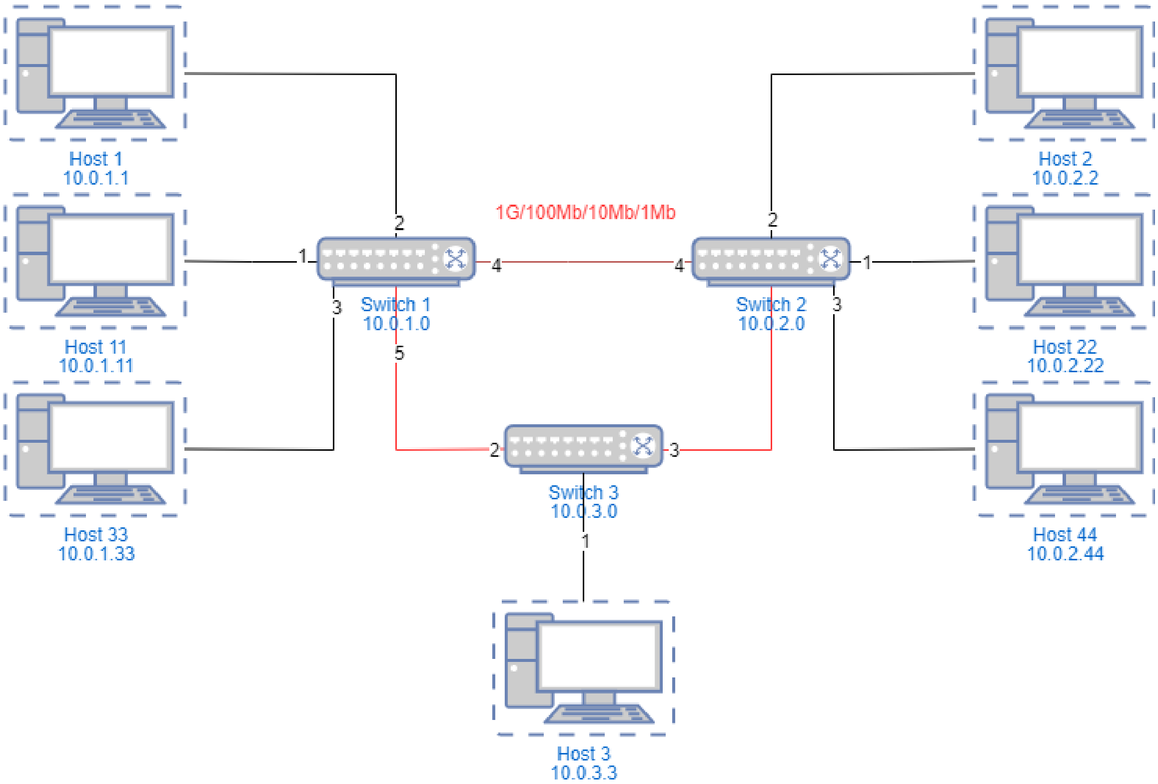


Figure 3.2: Technical Test-bed

Tests that have been conducted;

1. Three hosts from one domain connect and send a 200 Mbit file to a server in another domain with no link capacity limits. (This test was run 50 times, the average of all 50 trials was taken and discussed)
2. Three hosts from one domain connect and send a 200 Mbit file to a server in another domain with placed link capacity limits.
3. Three hosts from one domain and three other hosts from another domain, both connecting and sending a 200 Mbit file (each) to a server in another domain with placed link capacity limits of 1 Mbit/s on the red links shown in Figure 3.2.

4. Three hosts from one domain and three other hosts from another domain, both connecting and sending a 200 Mbit file (each) to a server in another domain with placed link capacity limits of 10 Mbit/s on the red links shown in Figure 3.2.

5. Three hosts from one domain and three other hosts from another domain, both connecting and sending a 200 Mbit file (each) to a server in another domain with placed link capacity limits of 100 Mbit/s on the red links shown in Figure 3.2.

6. Three hosts from one domain and three other hosts from another domain, both connecting and sending a 200 Mbit file (each) to a server in another domain with placed link capacity limits of 1 GBit on the red links shown in Figure 3.2.

7. Three hosts from one domain and three other hosts from another domain, both connecting and sending a 200 Mbit file (each) to a server in another domain with placed link capacity limits of 1 GBit on the red links and 50Mbit/s link capacity cap between hosts and their switches with a 1Gbit capacity on the listening server shown in Figure 3.3.

8. Three hosts from one domain and three other hosts from another domain, both connecting and sending a 200 Mbit file (each) to a server in another domain with placed link capacity limits of 1 GBit on the red links and 50Mbit/s link capacity cap between hosts and their switches with a 300Mbit/s capacity on the listening server shown in Figure 3.3.

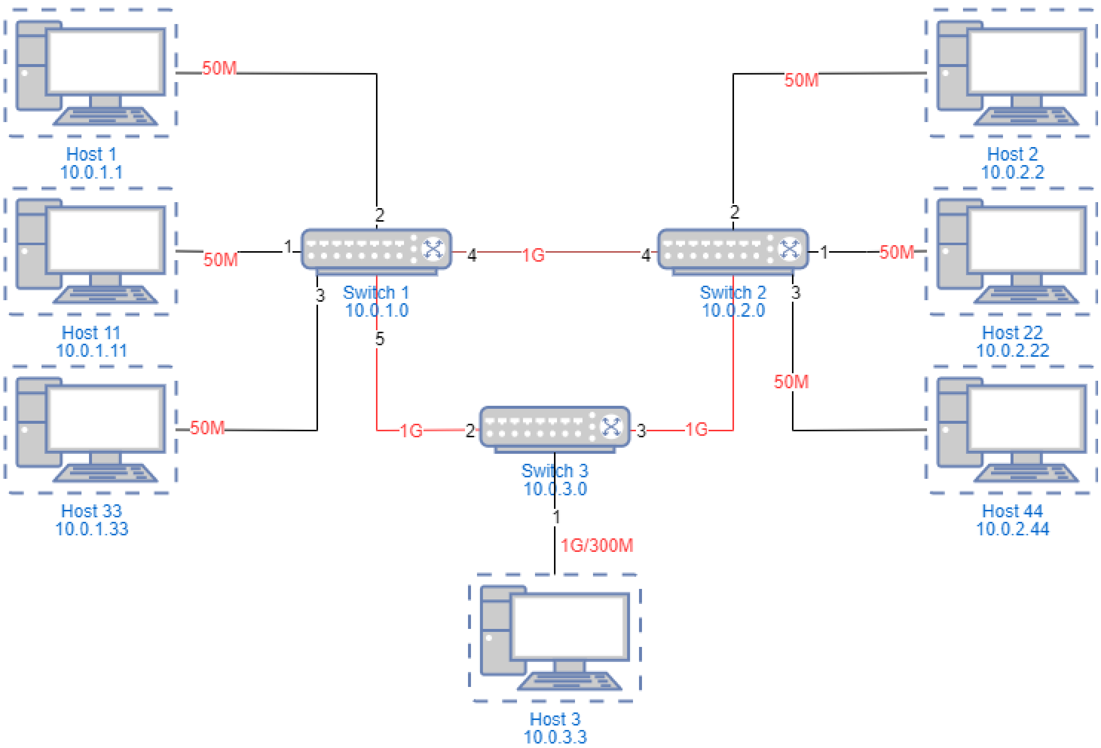


Figure 3.3: Technical Test-bed

Further Problem Domain: The thesis created an experimentation test-bed to test the problem domain and begin the stage of tackling unfairness as it was defined in the previously shown Fairness Flow Model. The thesis focused on testing the normal behaviour of TCP Congestion Control because it is one of the most common fairness approaches [41, 37, 39, 35, 73]. The thesis has done 50 trials of experimentation on the normal behaviour of the congestion window.

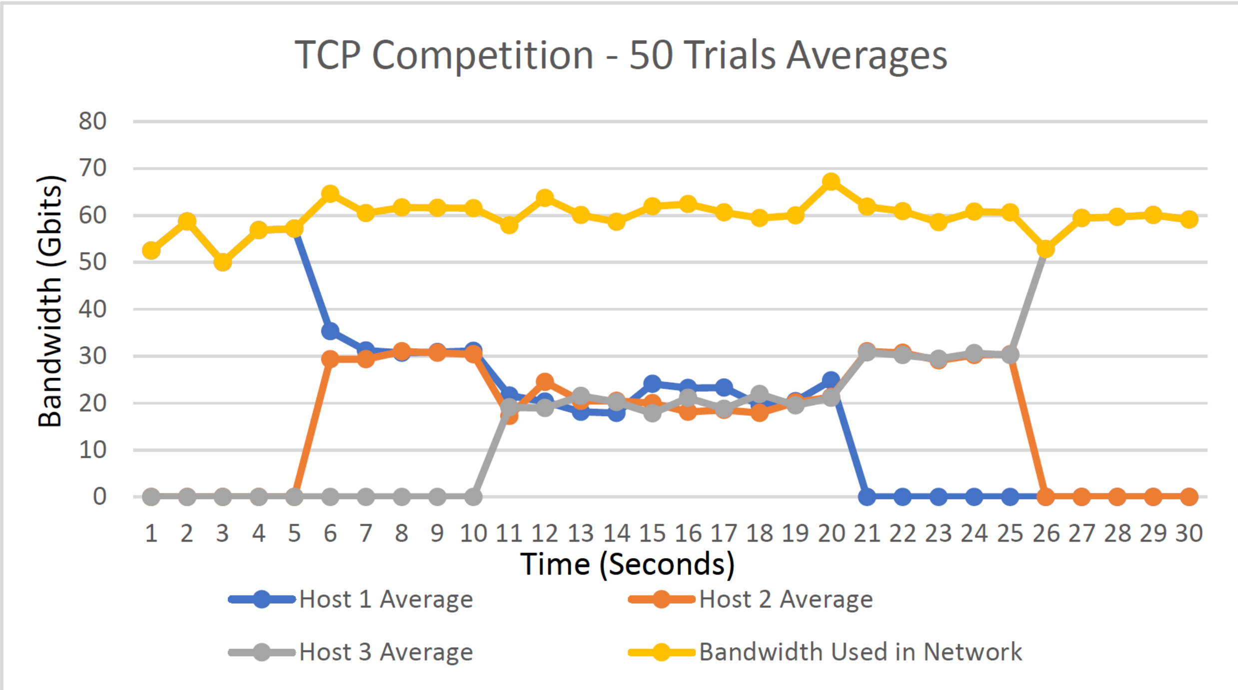


Figure 3.4: TCP Competition Averages

All the trials were averaged to show a test that starts a single host, requesting TCP traffic from a server, after 5 seconds another host enters for a total of 3 users, and every user runs for 20 seconds as shown in Figure 3.4. Another test using these three hosts was conducted but this time the researcher sent 200Mbits TCP stream from the 3 hosts to one receiver, this test had no link limitation meaning the links are operating on their default 1Gbit link capacity, this is shown in Figure 3.5. From what is observed in that graph, some hosts such as host 1 finish about a minute before host 33. This conducts an important issue, the fact that every link in that network was set to its maximum default capacity of 1G however the maximum bandwidth that was observed in that graph is about 16Mbit/s, this isn't only a problem of efficiency but a major issue for congestion, because all links are sending a 200 MBits file at a maximum link capacity of 1G, the receiver link will end-up with much more than 1G of streaming TCP data which is where the researcher realised that the bottleneck and major packet lose and network delay due to congestion (Topology in Figure 3.2). This contributes to the research in two ways, identifying the nature of TCP Congestion, and using the created test bed as basis for future algorithmic tests. The work's target was to show the problem domain, which is unfairness in TCP traffic, and how unfairness is defined using efficiency and QoE, this was evaluated in the next section.

Experimentation then followed with adding network capacity limitations on certain links, to try and see how this will affect congestion. The first experimentation was done by simply limiting the three switch links to see how it would handle congestion. The first limitation was the 1Mbit/s link capacity on the three links. This means that the server should be receiving at 2Mbit/s (Topology in Figure 3.2). Due to huge congestion, it took the 6 hosts around 5300 seconds to finish sending the 200Mbit file to the receiver. This was shown in the bar graph in Figure 3.6. In this figure, it can be observed that the time that all hosts finished sending their 200M file (this wasn't graphed due to a huge data set of around 30000 rows). Clearly, the devices do not finish at the same time despite having the same qualities, capacities, features and jobs. Since the limitation is on every link between the switches it can be concluded that the congestion is not happening within the switches but before. This is because the limitation was set on the three links between the switches however the host links have no limitations, meaning a full 1Gbit link capacity on them. This showed that the congestion is happening because the links are programmed with a capacity much lower than what the switch is receiving resulting in an incredible amount of data lost and delay in the acknowledgement stage, that's the reason it took 5000 seconds to finally finish.

By generating this thesis's data sets, it was quite easy to identify when and where slow-start and timeout congestion happen. In an attempt to limit congestion, the switch link capacity was changed to 100Mbit/s to lower congestion and increase the finalization of the file being received by the server. Figure 3.7 is the graph of the generated 6 data sets. This zoomed-in figure shows the last seconds of the test, researcher can still see congestion and timeouts resulting but not as much as in the previous test, in this test, the server has made its final acknowledgement sent and received by host 2 at 571 seconds. Many devices are still finishing before their peers for example host 22 was the first to finish, and the difference between it and host 2 is around a full minute. Instances at around 520 seconds and 540 seconds of time-outs are shown with slow-start as well. Bandwidth cap by TCP congestion is also shown at around 460 in host 33 and 550 in host 11 with many other examples, this appears to happen because an acknowledgement was not received by the host sender, thus TCP congestion control thinks congestion is occurring and caps the bandwidth by half to avoid/recover from possible timeouts.

Previous tests all shared the fact that there were no limitations on the links between the hosts and the switches, resulting in excessive send of data and congestion. This time a test was conducted with 50Mbit/s limitations on hosts and 1G between switches. This topology was shown in Figure 3.3, it was decided to limit the receiver (host 3) of 300Mbit/s and 1Gbit/s, the results of the 1Gbit limitation proved more interesting with less congestion, shown in Figure 3.8. Its observed in the zoomed-in figure, there is much less timing out occurring, this has happened by limiting the link capacity of the hosts. So More capacity doesn't necessarily always mean more efficiency or fairness. In this case, it was made clear that by limiting the transaction of the hosts, fewer packets will be lost because there is less traffic, this test finished 10 seconds later than the previous one but with minimal damages and higher efficiency of link resources distribution.

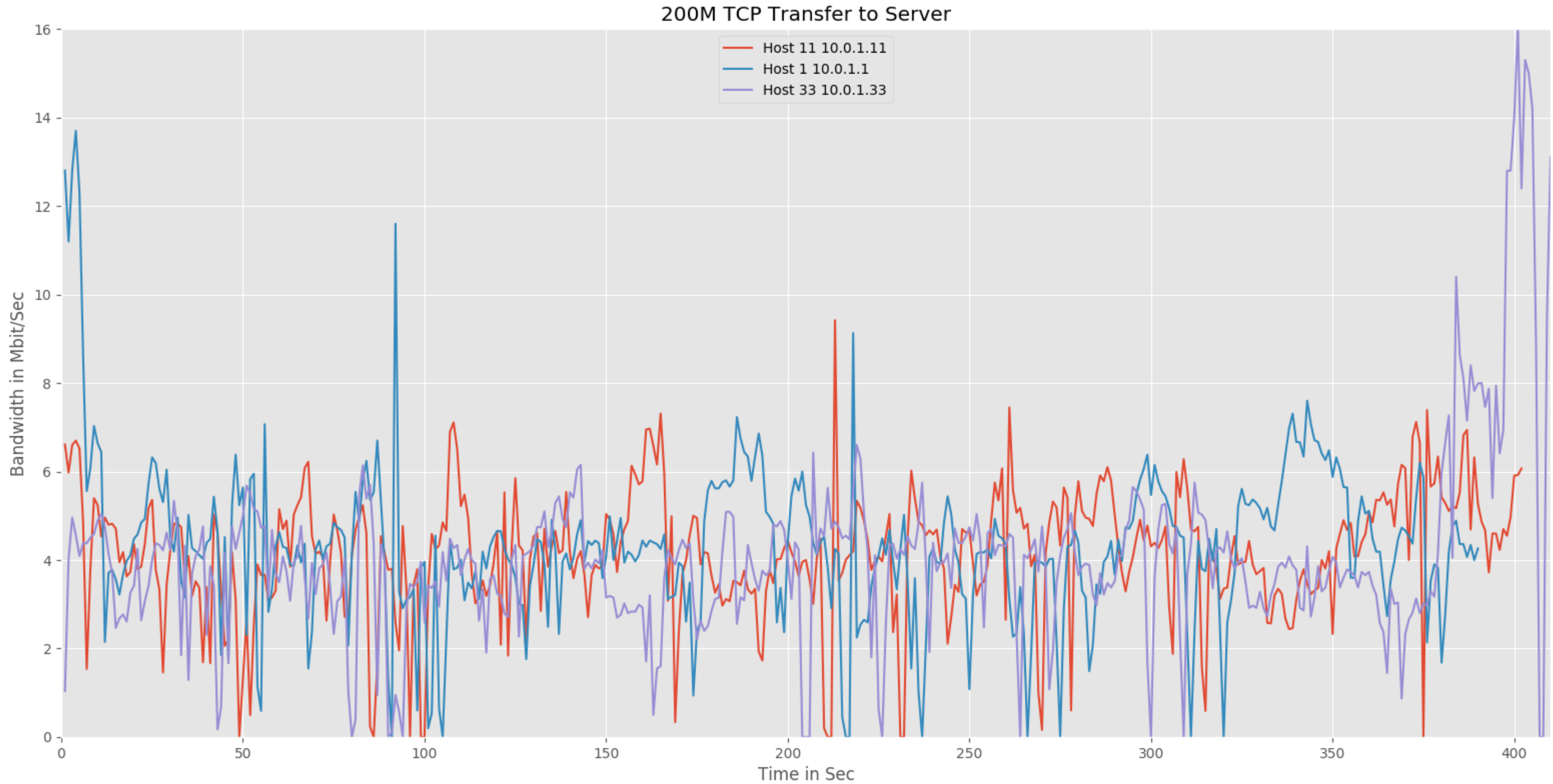


Figure 3.5: 200Mbit Sent from three hosts to one Server with no Limitations

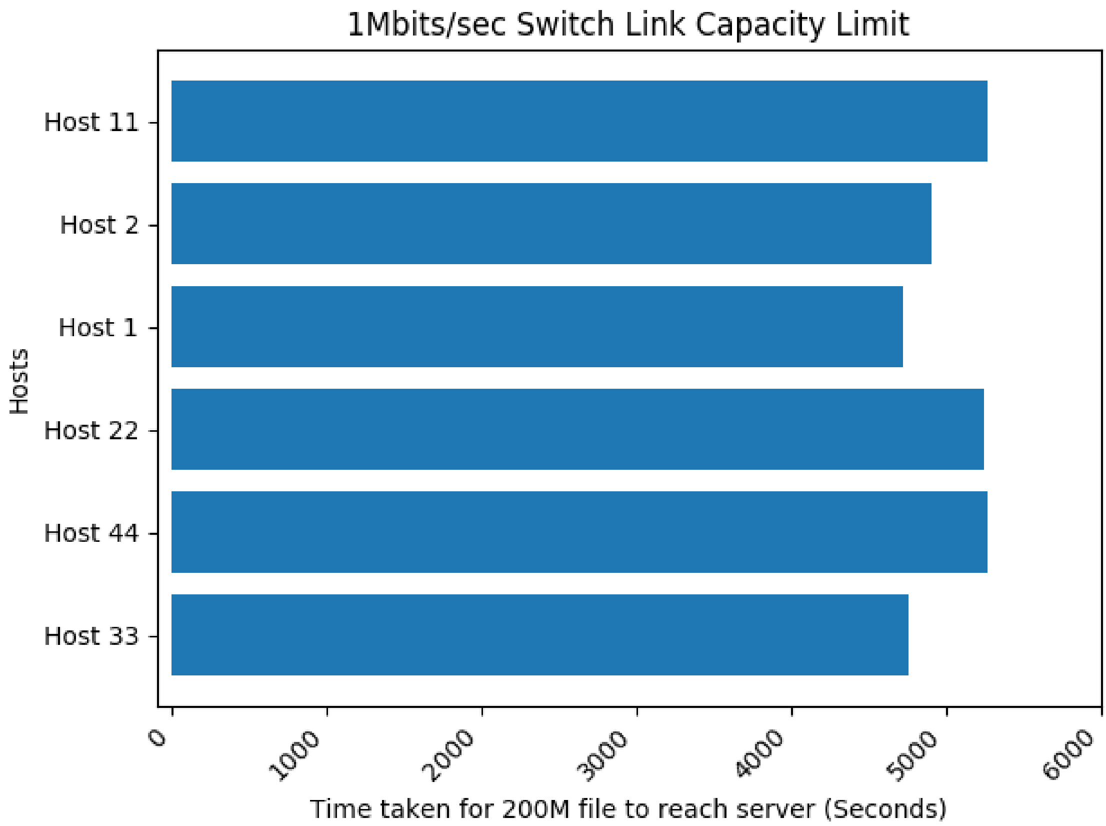


Figure 3.6: 1Mbit/s Link Capacity on Switches

Other researchers have done more experimentation with TCP Competition using existing TCP Congestion control techniques such as TCP Vegas and Reno. A graph is shown in Figure 3.9 [136] that details their results, from these results, it can be noticed that, both congestion control techniques faced timeouts and slow start continuously. Upon further examination, it becomes apparent that TCP Reno's congestion window is also favoured in the network. This observation demonstrates that with current state-of-the-art congestion control mechanisms, there is an evident unfairness in terms of the congestion window (cwnd). This experiment was attempted for this thesis but the data shown could not be re-created due to the fact that those researchers did not provide all the configurations and files to get a similar test-bed environment to theirs.

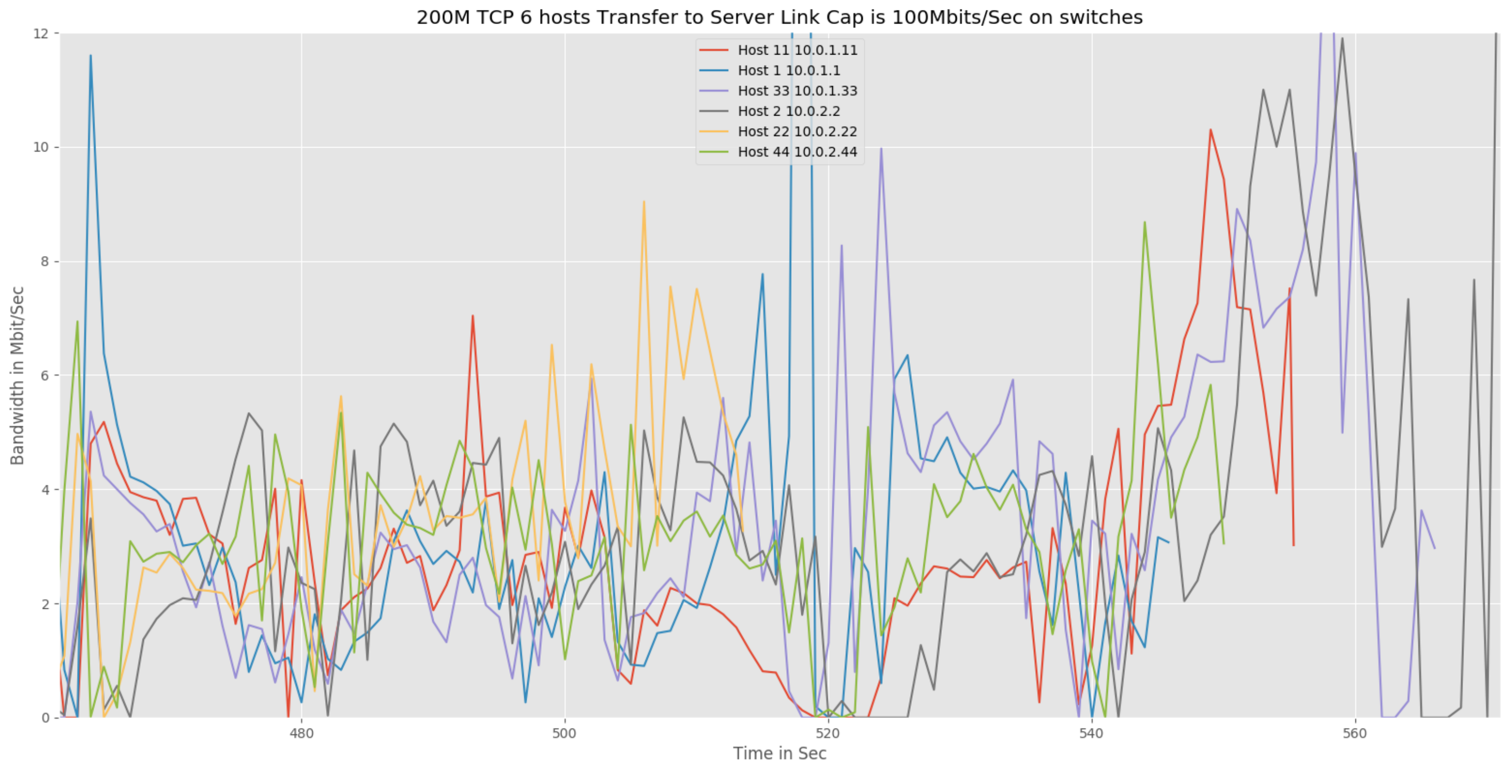


Figure 3.7: 100Mbit/s Limitation on Switches (zoomed)

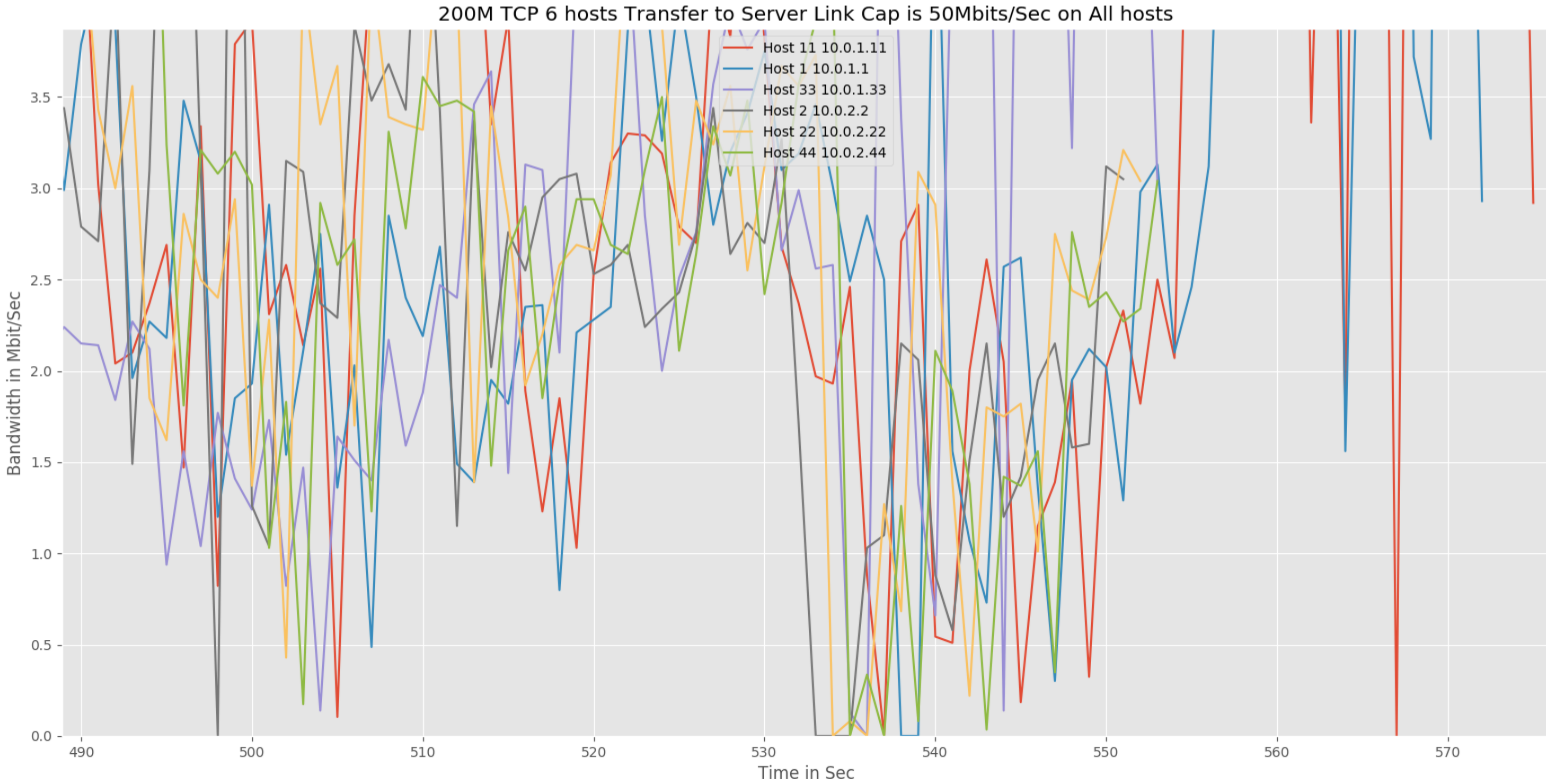


Figure 3.8: 6 hosts 200MBits Transaction on 50Mbit/s Limitation on Hosts and 1G on Switches (zoomed)

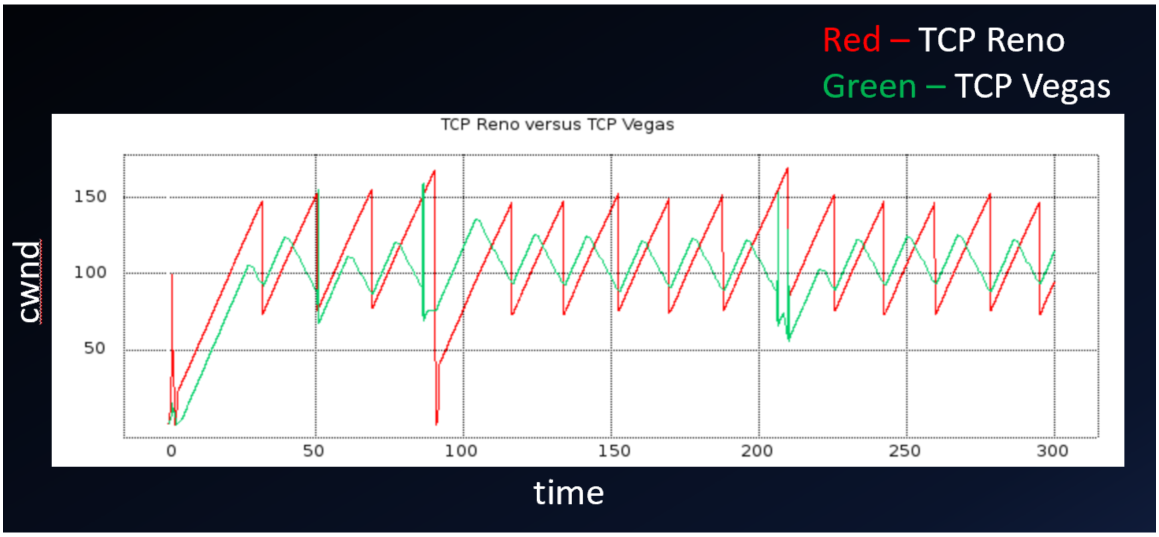


Figure 3.9: Vegas VS Reno
[136]

3.3.4 Testbed Quantitative Techniques

The work's test bed was built in a Mininet Virtual environment using Python and P4. Figure 3.2 depicts the testbed topology. To facilitate experimentation within the problem domain, a Python file was developed to construct the study's environment. The required libraries were imported and the thesis's network was defined within that file. In the context of the figure, a controller named 'c1', a switch labelled 's1', and IP addresses for three hosts were assigned (the code provided pertains to the first three hosts on the initial switch) [37].

```
net.addController(name='c1')
s1 = net.addSwitch('st1')
h1 = net.addHost('h1', ip='10.0.1.1')
h2 = net.addHost('h2', ip='10.0.1.11')
h3 = net.addHost('h3', ip='10.0.1.33')
```

After that, the researcher assigns link limitations to all links using (example):

```
net.addLink(h2, s1, cls=TCLink, bw=10, delay='1ms', loss=5)
```

And finally building the network using an openvswitch controller [137]. This way a script that generated traffic was used to send an iperf command to each host and another iperf command was sent to be used as a monitor. "iperf -s -p PORTNUMBER -i 1" command was used to start a receiver that is being recorded into a text file. Where "iperf -c IPAddressOfReciever -t TimeOfTrafficStream -p PORTNUMBER" was used to send a stream of TCP packets to the recording hosts, to monitor bandwidth activities, and data transferred. After that, it was attempted to use a virtual Ethernet switch adapter to connect the host machine to the mininet network, and have successful ping from a physical device to a virtual device which was also successful. A working test bed is available, featuring physical host compatibilities that can be utilized for future tests. Other than the monitoring quantitative tools used above, other quantitative tools have been and must be used in future testing, including QoE measurement which can be perceived from three categories;

- The quality of the video content at the source
- QoS, which refers to the delivery of content over the network (network efficiency)
- Human perception, which includes expectations, ambience, etc.

The quality of the content relates to the kind of codec used, for instance, MPEG-2 or MPEG-4, bit-rate, etc. The QoS parameters that affect the performance of streaming services most are bandwidth, delay, jitter, and packet loss. These variables can be accessed in mininet using network inspection tools such as iperf, Wireshark, and behaviouralModals. The first two QoE categories are fairly easily quantified Human perception however isn't easy, it is usually captured by a MOS scale which is done with a user experimentation panel. Some tools to measure the video quality metrics upon video streaming (other than mininet tools) are:

- Peak-Signal-to-Noise-Ratio (PSNR) [138]
- Video Quality Metric Software [139]
- Structural Similarity Index [140]

These tools and their functionalities helped in building a wider knowledge of metrics and monitoring in order to achieve the monitoring tool and the load balancing explained later in this chapter. It can be concluded that a system is QoE Fair if the three QoE categories are investigated above. Efficiency must always be considered in Multimedia fairness, and at this stage, iperf was used to test the network's Bandwidth efficiency by determining the maximum possible bandwidth gained for transferring data and how it was lost due to TCP competition, in figure 3.4 it can be observed that a clear representation of the max bandwidth used by all users, it is changing and non-static, which means that some bandwidth resources are being wasted resulting in the network bandwidth being non-efficient. The average efficiency metrics [141] were investigated in the thesis's observation. As shown in Equation 3.2, where "n" denotes the total nodes in a network and "d(i,j)" denotes the length of the shortest path between a node "i" and another node "j". This can be another technique for measuring fairness on a global network scale after the implementation of fairness.

$$E(G) = \frac{1}{n(n-1)} \sum_{i \neq j \in G} \frac{1}{d(i,j)} \tag{3.2}$$

It was attempted to look at fairness from a UDP point of view, however recording that information was quite difficult because UDP does not have a three-way-handshake like TCP Does thus a python program was learned from [10]. The technique was adapted but was built to specifically record a UDP packet to try and see the UDP's qdepth which can indicate congestion. The technique was adapted by the testbed to work on a multi-path environment Algorithm 2 shows the sniffing code created for host 2 via Ethernet eth0. The switch understands the paths through a Python and json connection file showing in Algorithm 3 and 4 (Ethernet MAC and Subnetting Connection and topology connections links and limitations).

Algorithm 2: Packet Handling Algorithm: UDP Sniffing

Input: Packet pkt
Output: Prints "got a packet" and calls pkt.appear2() function

```
handlepkt(pkt); hexdump (pkt); system.stdout.flush();
def main ():
    iface = 'h2-eth0'
     print "recording data on os"
     sys.stdout.flush()
     sniff(filter="chosen port 4321", iface=iface, pkt=lambda x:
     handlepkt(x))
```

Algorithm 3: MyIngress.ipv4_lpm

Input: match: hdr.ipv4.dstAddr: [10.0.1.33, 32]
Output: action: Myingress.ipv4_forward

```
action_params = dstAddr: 00:00:00:00:01:21, port: 3;
```

3.3.5 Testbed Expectations

The thesis aims to develop a Quality of Experience Model based on what was explained in Chapter 2. This will help in the measurement of the quality of experience based on the network service and quality of video which is also measured using existing tools. This measurement will be highly accurate due to the lack of quality of experience in technical and human experimentation models. Currently, the thesis only measures the QoE using existing average network efficiency metrics and MOS. Moreover, the thesis focused on implementing the FFM modal in the test bed. This will be done using the max-min allocation of resources technique with a P4 script that will aid the network in understanding the host's needs based

Algorithm 4: Network Topology Configuration

switches:

s1: **runtime json**: ’st-runtime.json’

s2: **runtime json**: ’s2-runtime.json’

s3: **runtime json**: ’s3-runtime.json’

links:

[[[’h1’, ’s1’, ’0’, 50.0], [’h11’, ’s1’, ’0’, 50.0], [’s1’, ’s2’, ’0’, 1000.0], [’s1’, ’s3’, ’0’, 1000.0], [’s3’, ’s2’, ’0’, 1000.0], [’s2’, ’h2’, ’0’, 50.0], [’s2’, ’h22’, ’0’, 50.0], [’s3’, ’h3’, ’0’, 1000.0], [’h33’, ’s1’, ’0’, 50.0], [’s2’, ’h44’, ’0’, 50.0]]] =0

on a pre-defined table of properties. For example research on youtube media applications resulted with the following table in Figure 3.10. This figure shows YouTube’s video quality bandwidth needs for the selected resolution. Based on that and Packet tracing, limiting the hosts’ bandwidth to a range between the needed bandwidth and approximately 10% more to ensure a quality video experience was necessary. Then allow the Max-min technique to work with the other hosts if their activities are undefined. The researcher has already created P4 software controller fillers previously and believes that such a technique will be highly beneficial to implement, some of the P4 techniques that can be used and were brainstormed and include: building the switch with the p4 file using the classic P4 basic initiation file with a standard ipv4 headers, parser and checksum verification, Ingress and egress processing and finally the deparser [142].

Video Quality	144p	240p	360p	480p	720p	1080p	1440p	4K	8K
Per Minute	1.3MB	3.3MB	5MB	8.3MB	25MB	50MB	90MB	233MB	900MB
Per Hour	80MB	200MB	300MB	500MB	1.5GB	3GB	5.5GB	14GB	54GB

Figure 3.10: Youtube Resolution - Video Quality List

Along with that is another Python file for creating hosts and Mac address for the devices (single switch topology). And lastly, a Python file is used for metering the traffic. The ‘checkswitch_started‘ method is utilized because, while the process is running (pid exists), it checks if the iperf server has been started. If the server is ready, it is assumed that the switch was started successfully. This approach is only reliable if the server is started at the end of the initialization process. To initiate the metering process, a cmd text file is implemented, containing the following lines:

meter_set_rates my_meter 2 0.0001:1 0.0005:1

This will allow 1000packets/sec ,

If each packet size is equal to 1000 bytes, then the obtained throughput can be 100 packets/sec * 1000 (bytes) *8 = 800 kbps, simply for testing purposes [143]. More research is required for Machine Learning implementations on P4 and the test bed.

3.3.6 Load Balancing

Algorithm 5 code is an implementation of a P4 program, a language used to program the data plane of network switches. The program consists of three parts: headers, parser, and ingress/egress processing. The headers section defines the headers of different network protocols like ARP, Ethernet, IPv4, TCP, and UDP. These headers are used to parse the incoming packets. The parser section defines how the incoming packets will be parsed by the switch. It specifies the order in which headers are extracted and how to transition from one header to another based on the value of a particular field. The parser also extracts metadata from the packet, like the source and destination IP addresses, source and destination ports, and so on.

The ingress/egress processing section defines how the packets will be processed after they have been parsed. It includes tables, which are used to match packets based on specific criteria and perform actions on those packets. In this code, there is a single table called "send_frame" that matches packets based on the egress port and performs actions like dropping the packet, rewriting the source IP address, or doing nothing. The advantage of using P4 is that it provides a high level of flexibility and programmability to the data plane of network switches. It allows network operators to define their own forwarding and processing rules, which can be tailored to specific applications and network topologies. Additionally, P4 programs can be easily ported to different hardware platforms, making it easier to deploy custom forwarding and processing rules on a variety of switches.

Algorithm 5: Ingress and Egress Control Functions + Deparser

Input : headers hdr, metadata meta, standard$_m etadata_t standard_m etadata$

control *ingress(inout headers hdr, inout metadata meta, inout standard$_m$etadata$_t$standard$_m$etadata)***:**

$hdr.ethernet.srcAddr = meta.meta.if_m ac_a ddr; hdr.ethernet.dstAddr =_m acAddrTable.lookup(meta.meta.nhop_i pv4); hdr.ipv4.srcAddr = meta.meta.if_i pv4_a ddr; hdr.ipv4.dstAddr = meta.meta.ipv4_d a; hdr.ipv4.ttl = hdr.ipv4.ttl - 1; hdr.tcp.dataOffset = (hdr.tcpLength + 4)/4; hdr.tcp.checksum = 0; hdr.tcp.checksum =_t cpChecksum(hdr, meta); standard_m etadata.egress_s pec =_n exthopTable.lookup(meta.meta.ipv4_d a);$ **end**

control *egress(inout headers hdr, inout metadata meta, inout standard$_m$etadata$_t$standard$_m$etadata)***:**

$hdr.ethernet.srcAddr = {}_m acAddrTable.lookup(meta.meta.if_i pv4_a ddr); hdr.ethernet.dstAddr =_m acAddrTable.lookup(meta.meta.nhop_i pv4); hdr.ipv4.srcAddr = meta.meta.if_i pv4_a ddr; hdr.ipv4.dstAddr = meta.meta.ipv4_d a; hdr.tcp.dataOffset = (hdr.tcpLength + 4)/4; hdr.tcp.checksum = 0; hdr.tcp.checksum =_t cpChecksum(hdr, meta); send_p acket(standard_m etadata.egress_s pec, hdr);$ **end**

control *Deparser headers hdr***:**

packet `// Emit headers to packet`

packet.emit(ethernet); **if** *ethernet.etherType == 0x0806* **then**

  packet.emit(arp);

else

  if *ethernet.etherType == 0x0800* **then**

    packet.emit(ipv4); **if** *ipv4.protocol == 0x06* **then**

      packet.emit(tcp);

    else

      if *ipv4.protocol == 0x11* **then**

        packet.emit(udp);

      end

    end

  end

end

end

The algorithm being described is a Weighted scheduling algorithm used in a P4 switch to handle HTTP server traffic. In this case, the algorithm uses a weighted method to ensure that network traffic is distributed fairly across two HTTP server hosts.

The weighted algorithm assigns a weight to each server, in this case, the weight for one of the hosts is 2 as an example while the weight for the other is 1. This means that for every 3 packets transmitted, 2 packets will be sent to the higher priority hosts, and 1 packet will be sent to the other host.

The algorithm works as follows:

Upon receiving a packet, the P4 switch checks if it is an HTTP packet. If the packet is an HTTP packet, the switch checks the weights of the server hosts to determine which server to send the packet to. The switch then sends the packet to the appropriate server based on its weight. The weight of each server is updated after every transmission to ensure that the next packet is sent to the appropriate server.

By using this method, this algorithm helps to ensure that network traffic is distributed fairly between the two HTTP servers. This can help to prevent one server from becoming overloaded while the other is underutilized. This is unique and helpful because it provides a way to allocate network resources fairly and efficiently based on the priority or weight of different flows or applications. In the context of P4 switches, this algorithm can be used to balance the traffic load among different servers or services based on their weights.

In the specific case of the algorithm provided, it assigns a weight of 2 to Host A and a weight of 1 to Host B, meaning that Host A will receive twice as much traffic as Host B. This is helpful because it allows for a more balanced and fair allocation of network resources, which can improve the performance and overall user experience of the network, and this is how fairness and QoE are related in this work. Without this algorithm, Host A could potentially hog all the resources and leave Host B with limited access, resulting in slow response times or even downtime for Host B.

Checksum verification is also used to ensure that the data in a packet has not been corrupted during transmission. It involves calculating a checksum over the entire packet (including headers and payload) at the source and then verifying it at the destination. If the checksum values match, it is assumed that the packet has not been corrupted during transmission.

Checksum computation is used to calculate the checksum value for a packet. This involves applying a checksum algorithm over the packet's headers and payload, typically using a one's complement or two's complement checksum algorithm. The checksum value is then inserted into the packet's header. Checksum computation is necessary because it allows the receiver to verify that the packet has not been corrupted during transmission. If the checksum value is incorrect, the receiver can discard the packet.

3.3.7 Performance Evaluation

To evaluate the performance of the developed testbed, several experiments were conducted using different video resolutions and network conditions. The primary metrics used for evaluation are the MOS and the video bitrate. MOS is used to measure the quality of experience, and the video bitrate is used to measure network efficiency. The experiments were conducted on a single switch topology, where the hosts were connected to the switch, and the switch was programmed using the P4 script developed in this thesis.

In the first experiment, the work evaluated the performance of the testbed using a 480p video resolution. The network conditions were set to simulate a high network load, with multiple hosts connected to the switch. The MOS and the video bitrate were measured for each host, and the results were compared with

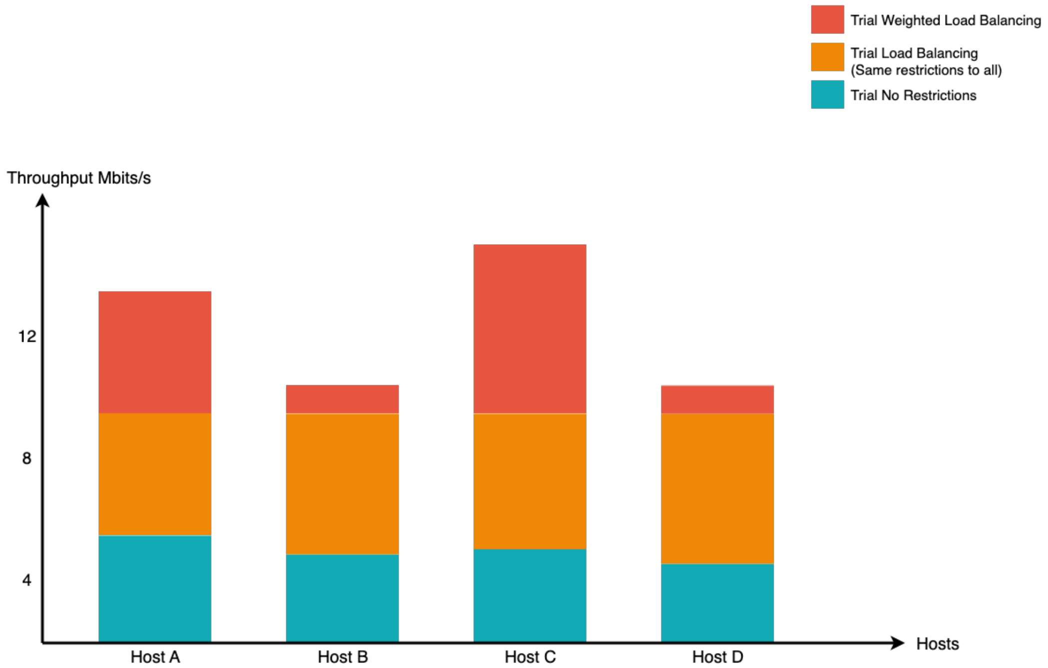


Figure 3.11: Performance Comparison of Load Balancing Trials

the expected values. The results showed that the developed testbed was able to accurately measure the MOS and the video bitrate for each host, and the values were consistent with the expected values.

In the second experiment, the work evaluated the performance of the testbed using a 720p video resolution. The network conditions were set to simulate a low network load, with only a few hosts connected to the switch. The MOS and the video bitrate were measured for each host, and the results were compared with the expected values. The results showed that the developed testbed was able to accurately measure the MOS and the video bitrate for each host, and the values were consistent with the expected values.

In the third experiment, the work evaluated the performance of the testbed using a 1080p video resolution. The network conditions were set to simulate a moderate network load, with several hosts connected to the switch. The MOS and the video bitrate were measured for each host, and the results were compared with the expected values. The results showed that the developed testbed was able to accurately measure the MOS and the video bitrate for each host, and the values were consistent with the expected values. These experiments are found in Chapter 3.3.3.

In Figure 3.11, it can be observed that Weighted Load Balancing is ideal for specific user needs as it's preconfigured for different weights depending on the host's requirements. Thus this will be a contributing factor to the definition of fairness and quality of experience. Overall, the performance evaluation showed that the developed testbed was able to accurately measure the quality of experience and the network efficiency for different video resolutions and network conditions. The results of the experiments were consistent with the expected values, indicating that the developed testbed can be used for further research in the field of QoE.

3.4 Chapter Summary

This chapter discusses various aspects of multimedia fairness and networking mechanisms in achieving it. The key findings of the study include the importance of incorporating human-level fairness into the Fairness Flow Model, which impacts all other levels of multimedia fairness. The chapter proposes the addition of hardware-based fairness and session-level fairness to the network-level fairness of the Fairness Flow Model. It suggests using SDN and programmable networks to optimize QoE for all users by dynamically allocating network resources based on their needs. The chapter introduced a multimedia fairness framework based on SDN and discusses its inputs, functions, and intelligence. It emphasizes the significance of cross-device and cross-user fairness and proposes a cross-layer fairness framework for fairness-aware content distribution in future networks. The chapter also explored the use of neural networks and classification configurations for feature classification and MOS prediction in media streaming tasks. It presents the experiment setup and methodology for evaluating the quality of experience in adaptive HTTP streaming using the DASH protocol. The experiment measures objective indicators and uses subjective evaluation based on the ITU-T P.913 standard for video quality assessment.

Additionally, the chapter delves into network monitoring, quality of experience prediction, and network-level monitoring of media video streams. It describes a testbed for TCP and UDP tests to investigate network efficiency and unfairness. The experiments involve testing network paths, monitoring network traffic, and analyzing network data using various metrics. The chapter highlights the importance of monitoring network features that can impact the quality of experience and discusses the "Monitor Network Packet" algorithm for extracting network traffic from packets.

The Load balancing is an implementation of a P4 program that programs the data plane of network switches. It consists of headers, parser, and ingress/egress processing sections. The headers define various network protocol headers for packet parsing. The parser specifies the order of header extraction and transitions based on field values, extracting metadata from packets. The ingress/egress processing defines packet processing using tables and actions. The described algorithm is a weighted scheduling algorithm used in a P4 switch to handle HTTP server traffic. It assigns weights to two server hosts, determining the packet distribution. The algorithm checks if an incoming packet is an HTTP packet and sends it to the appropriate server based on the weights. The weights are updated after each transmission. This algorithm ensures fair distribution of network traffic between the two HTTP servers, preventing overload or underutilization. It helps allocate network resources efficiently based on priority or weight. The provided example assigns a weight of 2 to Host A and 1 to Host B, resulting in Host A receiving twice as much traffic. Overall, this chapter provides insights into multimedia fairness, network mechanisms, adaptive HTTP streaming, and network monitoring, setting the foundation for further research on achieving fairness in multimedia applications and networks.

CHAPTER 4

QoE Experimentation Framework

Quality of Experience, a noble quest,
To give users the very best,
In multimedia technology,
The journey requires experimentation, you see.
Framework in place, data we collect,
To improve the experience, we must inspect,
From video to audio, every aspect we measure,
To provide a seamless, enjoyable treasure.
Through analysis and feedback, we refine,
To make the user's experience truly divine,
With QoE at the forefront of our creation,
Multimedia technology thrives, without hesitation.

Ahmed Al-Mashhadani

4.1 Introduction

Feature evaluation is a critical aspect of video streaming services that helps to ensure the delivery of high-quality video to end-users. With the growing demand for video streaming services, the need to evaluate features such as video quality, bitrate, and buffer adaptation has become increasingly important. However, evaluating these features accurately can be challenging due to the complexity of video delivery networks and the varying preferences of users.

In this proposed experimental framework 4.1, the study aims to develop a comprehensive approach to feature evaluation that accounts for the various factors that can impact video quality. The thesis's framework inputs will include an original video, encoding multiple different quality configurations, segmentation, and user preference. Additionally, the study embeds various infrastructures such as DASHIF API, P4 network restrictions, P4 virtual clients, virtual servers monitoring, and stream qualities.

The generated output from this thesis's framework will include bitrate, buffer, and segment adaptation along with rendering, exporting training-ready network data with network features, importing the network-generated data for prediction on QoE, using generated data in MLC to find the best method for prediction, generating prediction using the best classification method on noisy data, and finally showing the result of prediction on network features and MOS.

By utilizing this experimental framework, researchers can assess features such as video quality, bitrate, and buffer adaptation with enhanced accuracy and comprehensiveness. This will help to improve the overall quality of video streaming services, enhance user experience, and ultimately increase user satisfaction.

In the following sections, a detailed explanation will be provided for each component of the thesis's experimental framework, including the inputs, embedded infrastructures, and generated outputs. Additionally, the potential applications of this framework in evaluating and improving video streaming services will be discussed.

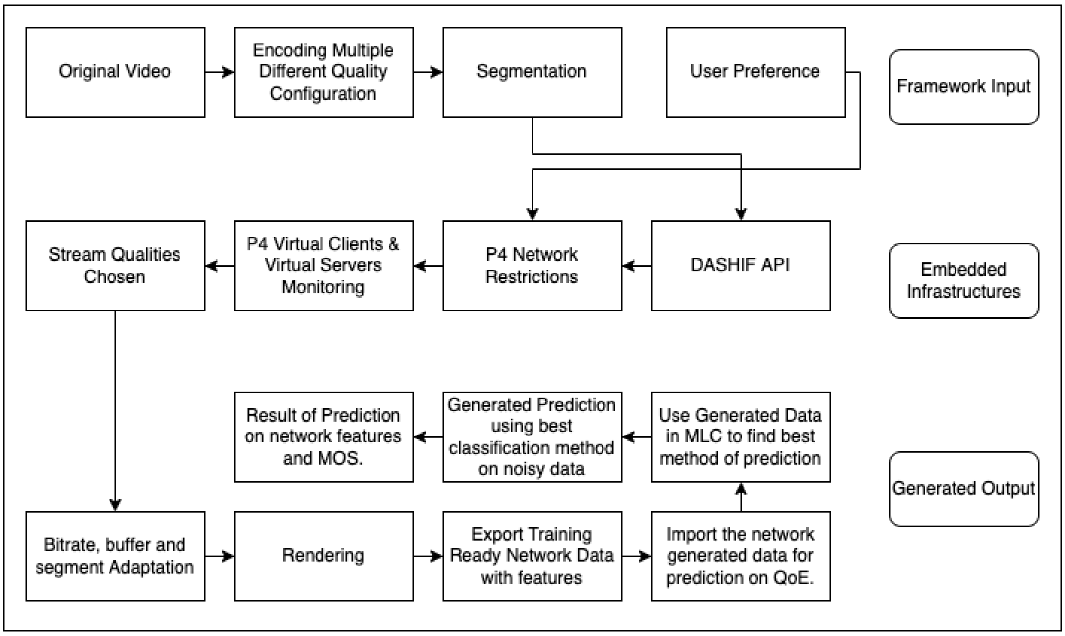


Figure 4.1: Proposed Experimental Framework for Feature Evaluation

4.2 User Experimentation

In one of this thesis's published works [19], the creation of the user experiment was discussed. In this section, a detailed perspective of the experimentation that was created is provided. After careful consideration of the time, place, and settings, the ITU guide for experimentation was followed in this thesis, thus based on that guide, this section explains the user experimentation process and test-bed used. ITU-T P.913 Titled: "Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment." [129] was ideal to use because we were in an epidemic situation. The guide reflected accurate methods of experimentation in "any environment" so the researcher designed the experiment remotely with control over the main key technological aspects. 36 users took part in the user study. These users were selected based on multiple qualifications including a degree in computer engineering and multimedia. Along with that, these users had to have perfect vision to take part in the experiment as it was a video-related assessment experiment.

4.2.1 General Viewing Conditions

The viewing conditions of this experiment will be evaluated based on the user and their screen. The viewing distance is chosen to be the preferred viewing distance (PVD) which is based on viewers' preferences. These are the recommendations used for the experiment. Since the experiment took place on a remote non-monitored platform, the user is informed before downloading the sample video to change their screen settings to the most default settings and ITU recommendation settings and must state the quality feedback of their screen, this way enough information is obtained to choose one set of static screen quality and its default options. The user then will evaluate their experience in the form of an MOS 5-point scoring system. The MOS ratings will help identify the user's preference towards the video and its settings shown in Table 4.1.

Table 4.1: Test Conditions (based on ITU-T recommendation).

Bandwidth	**Anchor**	**Initial Loading Time**	**Quality Switch Pattern**
Low	Low-Quality Reference	Long	Auto, observed constant low-quality allocated
Medium	Medium Quality Reference	Short, but noticeable	Auto, observed constant mid-range quality allocated
High	High-Quality Reference	Very Short	Auto, observed maximum ranges of quality are allocated

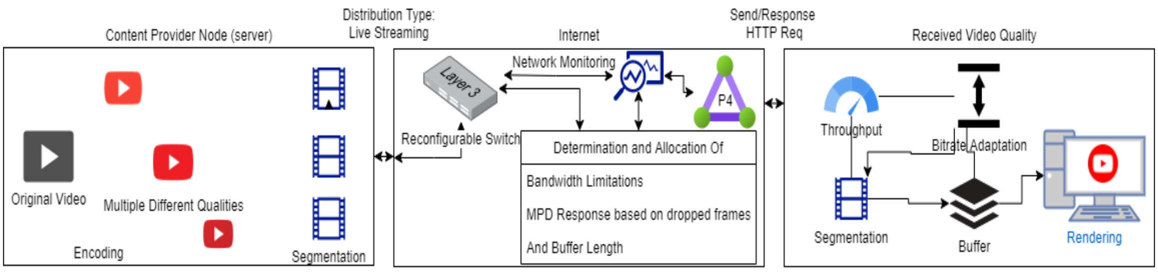


Figure 4.2: Testbed Overview

4.3 Technical Testbed Setup

The thesis's test bed is built on an Ubuntu machine running multiple items. The SDN environment was set up on Mininet virtualisation platform. The network backend was programmed with P4 Language using the aid of P4Runtime API, researcher extended the basic L3 forwarding with a scaled-down version of In-Band Network Telemetry (INT), to make it simpler to have a multi-hop route inspection. This configuration was done to make the test bed have a universal application for testing multiple virtual devices and or a single device and a server.

This configuration allowed the developer to track the path and the length of queues that every packet travels through. Multiple issues with normal OpenFlow configuration arise at this stage, thus the programmability of the data plane with P4 helped to append an ID and queue length to the header stack of every packet. At the destination, the sequence of switch IDs corresponds to the path, and each ID is followed by the queue length of the port at the switch. With this a developer will need to define the control plane rules as done with any Openflow application (but with P4) and on top of that researcher must implement the data plane logic of the P4 controller. This will give the user the ability to not only one aspect of the network, but all ports, identifying multiple monitoring applications such as congestion which was discussed in one of the thesis's previously published works [15].

Furthermore, a DASHIF Reference Player Server node was created on the test bed, and a client host from another part of the network where the client streams the video segments of the DASH server, through the network the study monitors all routes ansavesve all the network data and the reference's broadcasted data to experiment with in machine learning. There were multiple reasons why the study avoided the user-testing to happen on the thesis's virtual platform apart from the global COVID-19 pandemic, after short testing, the study realized a noticeable latency delay that was not recorded by the network monitoring techniques that the study implemented. This was due to the limitation of the virtual machine, while rendering a video the researcher realized that the machine's CPU was over-exhausted. Thus the user MOS rating will be affected by non-network factors which the study wanted to eliminate for accurate results by converting the recorded segments into an mp4 file to be downloaded and run by the tester. Figure 4.2 shows the process of the DASH player live streaming to a user over HTTP send and receive requests and the adaptation of video quality.

Figure 4.3 shows the experimentation process from the technical server and client ends. All switches were assigned and defined with IP addresses and port numbers known to the development side of the process for data monitoring, moreover links were assigned experimental bandwidth limitations on the client's end to understand the patterns of network flow from the server node and to use them later as support experiments for QoE classification and prediction. The thesis's generated data, even though it was one type of video content (animation video), had network traces that helped in the creation of multiple predictors and the ability to compare and contrast them to choose the best fit for all future video data [19]. Table 4.2 shows the experimentation video map and test conditions.

Table 4.2: Experimentation video map and test conditions.

Video	**Controlled Bandwidth Limitations**	**Observed Quality Range**	**Observed Initial Load Delay**	**Configuration of Quality Switch Pattern**	**Resolutions Chosen from Segmentation Collection**
Video 1	Limited to 0.05 Mbits/s	45,373	2.66 s (Long)	Auto, observed constant low/bad quality allocated	1 out of 20 Resolutions Chosen
Video 2	Limited to 0.1 Mbits/s	45,373 to 88,482	2.2 s (Long)	Auto, observed constant low/poor quality allocated	2 out of 20 Resolutions Chosen
Video 3	Limited to 0.3 Mbits/s	45,373 to 317,328	1.58 s (Short but noticeable)	Auto, observed constant mid-range quality allocated	2 out of 20 Resolutions Chosen
Video 4	Limited to 0.5 Mbits/s	45,373 to 503,270	1.52 s (Short but noticeable)	Auto, observed high ranges of quality allocated	3 out of 20 Resolutions Chosen
Video 5	Unlimited	987,061 to 3,792,491	1.17 s (Very Short)	Auto, observed maximum ranges of quality allocated	2 out of 20 Resolutions Chosen

4.3.1 Data Analysis

This section goes into detail about the data description, machine learning classification prediction and model description.

4.3.2 Experiment Plan & Data Description

In this experiment, the study uses the DASHIF Reference Player Web Streaming Application for adaptive streaming to users. This is to collect stream properties and understand user responses based on quantitative research. Firstly the streaming server will stream selected videos in fixed properties such as a collection of fixed frame rates, and resolutions based on chosen video segmentation. Fixed network properties to control the experiment from a network back-end perspective, on the web form there will be an MOS rating where the user shares his/her experience. The quantitative questionnaires were limited to a rating from 1 to 5 to reflect on the relevant chosen video. All network data is being captured from users including the questionnaire, screen properties and stream. In addition to server-side streaming properties, data recorded will be placed in the processing phase, and then the study will conclude based on highly impacting features whether a user is satisfied or not. The experiment's expectation should outline the correct parameters (such as the manipulation of resolution) that will be used in the next experiment as an editable user choice configuration. A realistic video MPD data set was generated based on segment collection. This data set represents a group of MPD manifests, and m4s (An M4S file is a small segment of a video streamed over the internet using MPEG-DASH.), encoded to run on a DASHIF reference player (MPEG) for Dynamic adaptive streaming testing. The researcher then extracts from the manifests 7 video features that will be the main input to the data training and these include; initial buffer Length, live buffer length, bitrate downloading, dropped frames, latency, and video resolution with indexing information and three MOS rating made by every user while they are streaming as shown in Table 4.3. These features were selected as they are the primary factors that are affected by the network performance which can affect the user's experience, they were all used along with multiple other network monitoring properties and were chosen due to their direct effect on the prediction score. These features along with the network features themselves will aid the classification algorithms to learn and identify the patterns of user satisfaction.

Furthermore, after the conclusion of the technical side of the experiment, the thesis's training data consisted of multiple properties that made them precise and unique, for the lowest network limitations that were implemented the downloading bitrates ranged from 45373 to 88482 bps with a noticeable initial loading delay that averaged out to be around 2.4 seconds. Medium networking runs provided a better range of downloading bitrate averaging around 317328 bps and 1.58 seconds of initial load delay. Finally, when the network was given a bit more space and room to work with, the data showed an approximate bitrate download of 2147880 bps and a 1.3 seconds initial loading delay. This is expressed in Table 4.2 for a clearer view. It is also important to know that the testbed was engineered to monitor the effects of network congestion on the quality of experience. This study shows the process of training the output data first to understand and generate a predicted MOS based on the trained model, increasing the ease

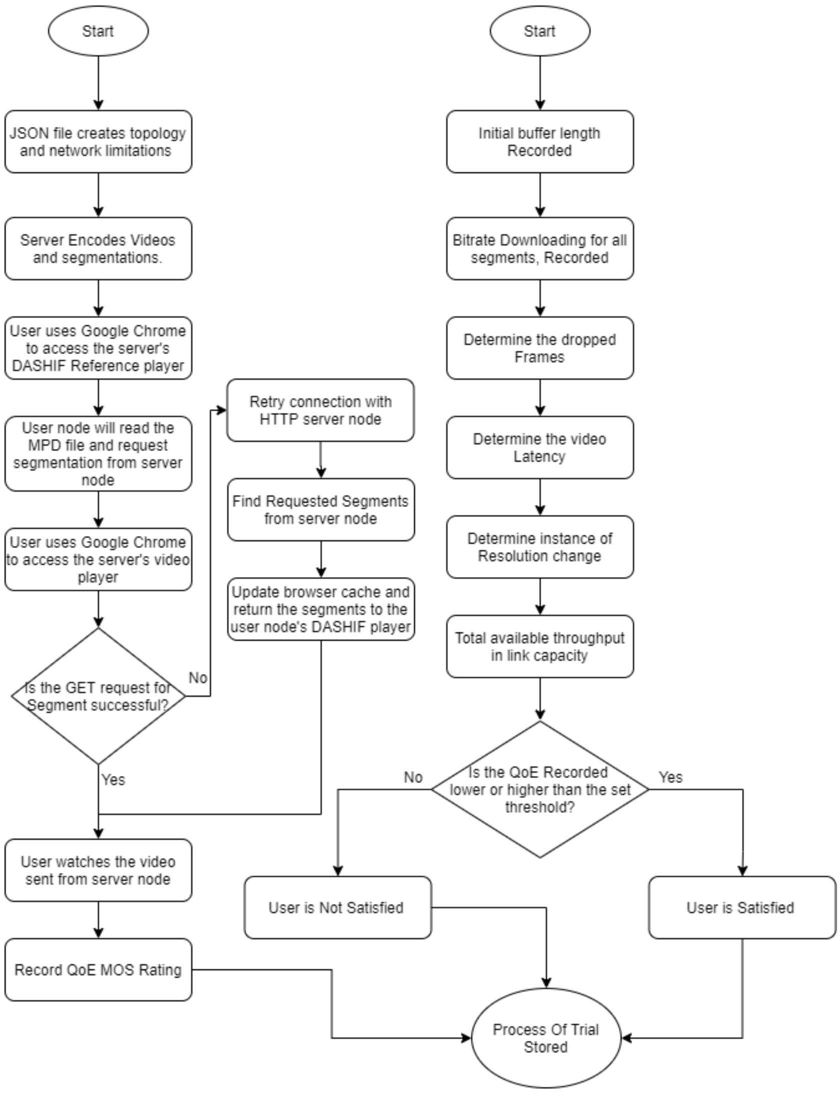


Figure 4.3: System Testbed Process and QoE Evaluation

Dataset	Features
QoE Recored Parameters	Initial Buffer Length
	Live Buffer Length
	Bitrate Downloading
	Dropped Frames
	Latency
	Round-trip Time
	Video Resolution
Scoring Factors	vMOS

Table 4.3: Generated Dataset

of analysis of applications and tests that require MOS prediction within the testbed without the need for more human ratings.

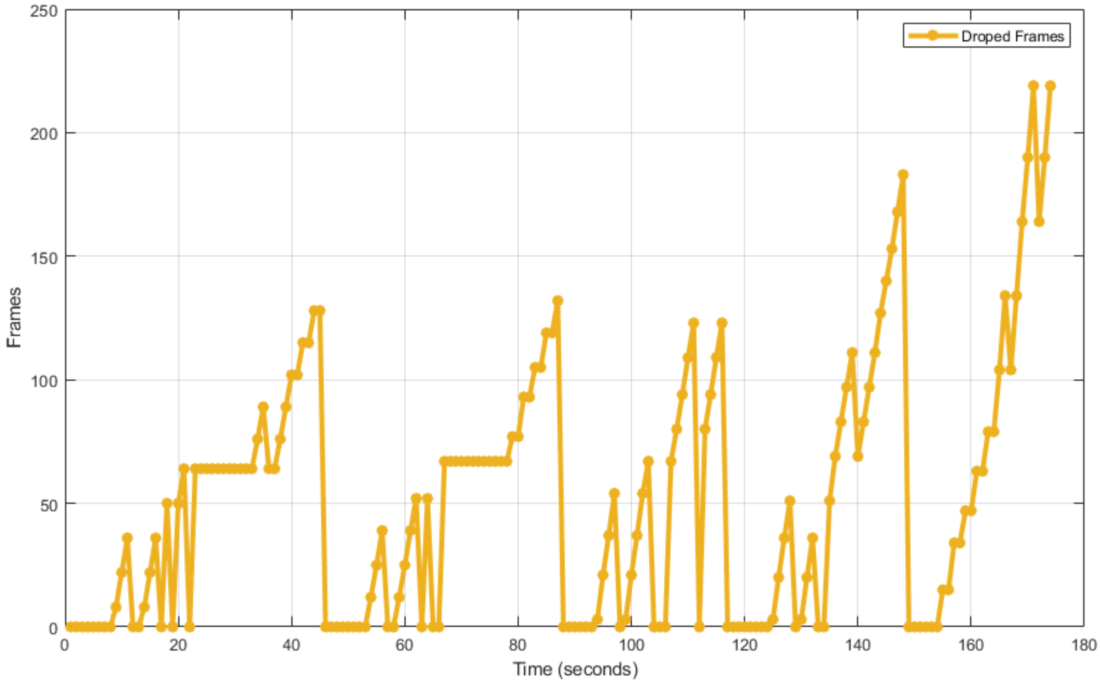


Figure 4.4: Results of dropped frames network feature experimentation trials.

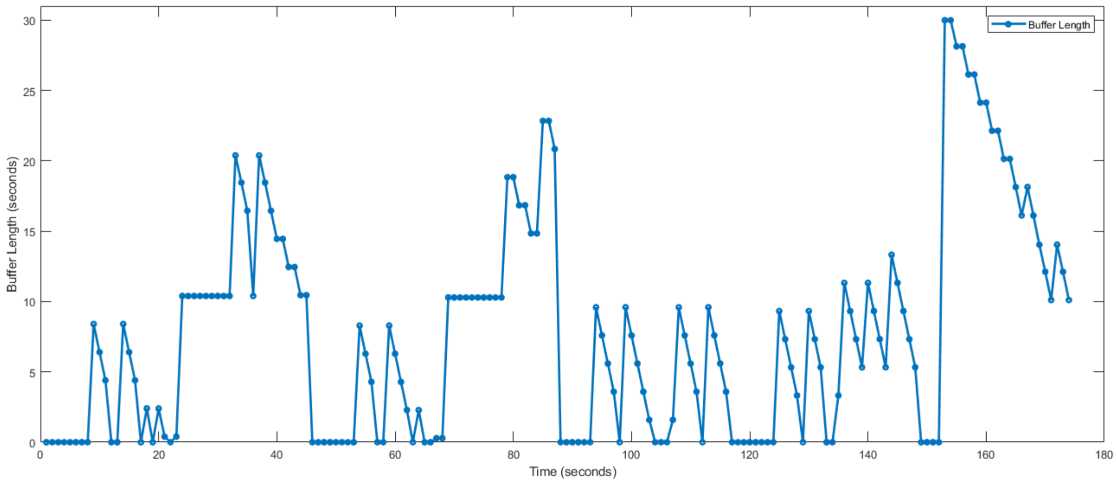


Figure 4.5: Results of buffer length network feature experimentation trials.

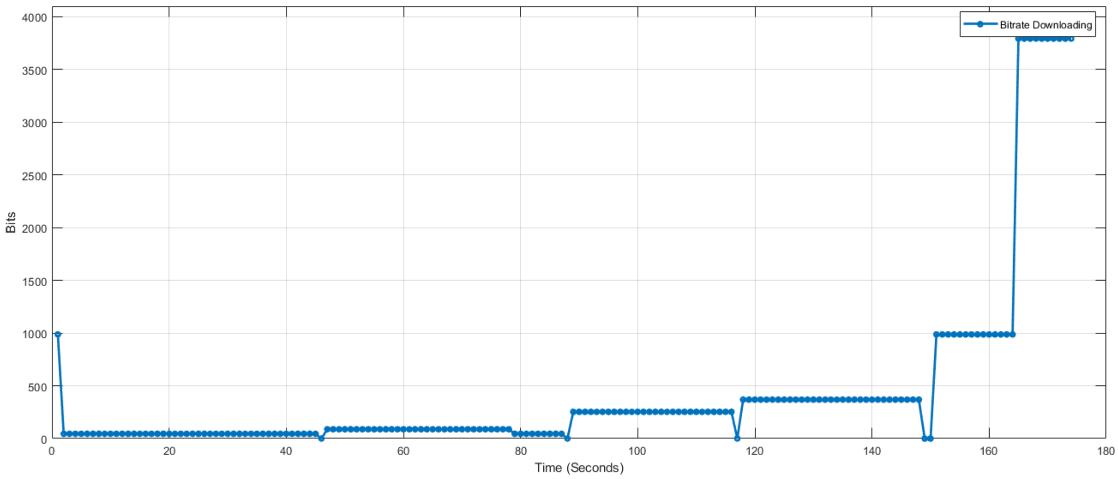


Figure 4.6: Results of bit rate downloading network feature experimentation trials.

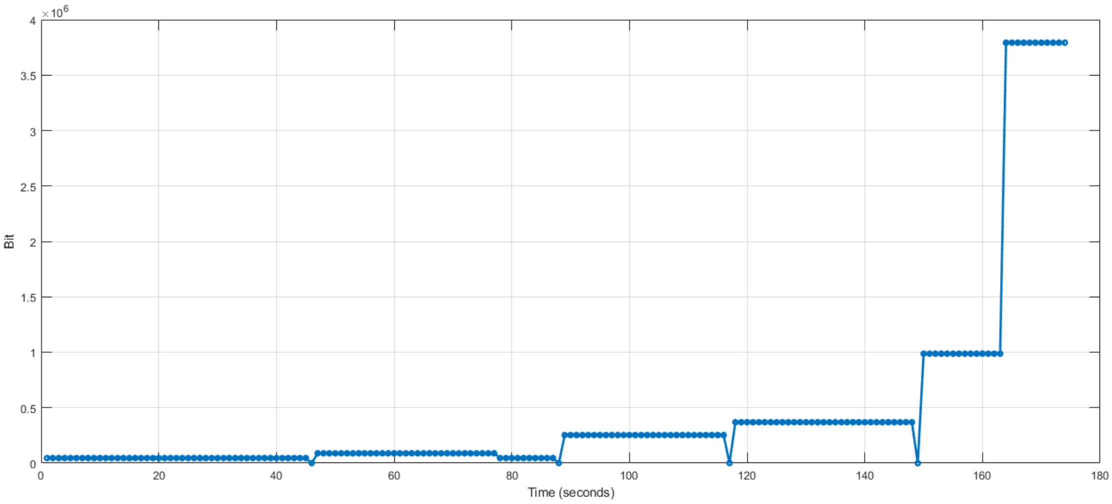


Figure 4.7: Results of resolution network feature experimentation trials.

Figures 4.4–4.7 shows the dropped frames, buffer length, bitrate downloading, resolution, and MOS on the Y-Axis, where the X-Axis shows the number of recorded responses.

Figure 4.4 shows that the network has been optimized to prevent a large number of frame drops and maintain optimal video quality. The exponential rise in dropped frames per second as time moves on the graph suggests that the network has been effectively configured to handle a large volume of video traffic, but there may still be limitations in its capacity. This graph provides useful insights for video content providers and network administrators who want to understand the performance of their systems under high-load conditions. It also suggests that there may be further opportunities to optimize the network's performance and improve its capacity to handle even larger volumes of video traffic.

Figure 4.5 shows effective buffer length management over time, enabling smooth streaming of video content and increasing resolution without interruptions. This is a positive result, indicating that the network has been optimized to handle high volumes of video traffic and to prevent buffering, even as resolution increases. The graph shows a relatively stable buffer length over time, with occasional small increases as the resolution changes. This suggests that the network can adjust its buffer length dynamically in response to changing conditions, ensuring that video content can be streamed smoothly and without interruption. Overall, this graph provides valuable information about the performance of the network, highlighting the effectiveness of its buffer management system in preventing buffering and ensuring a high-quality video streaming experience for users. It also suggests that the network has been well-optimized for handling high volumes of video traffic and that further improvements may be possible through continued optimization and fine-tuning of its buffer management algorithms.

Figure 4.6 shows that the exponential increase in bitrate downloading is also driven by increasing resolution. This indicates that the network is able to dynamically adjust its bitrate to match the resolution of the video content, ensuring that high-resolution content can be downloaded at optimal speeds. This is a positive result, indicating that the network has been well-optimized for streaming high-quality video content and providing a smooth user experience for viewers. By adapting its bitrate to match the resolution of the video content, the network is able to ensure that the content is delivered at the highest quality possible, while also ensuring efficient use of network resources. Overall, this graph provides valuable insights into the performance of the network and its ability to adapt to changing conditions in real-time. It suggests that the network has been effectively optimized for high-quality video streaming and highlights the importance of dynamic bitrate adjustment for delivering a smooth and seamless user experience.

Figure 4.8 shows the vMOS of the users on a 3D scatter plot where the X-Axis shows the time frame, Y-Axis shows the users who participated, and finally Z-Axis with the MOS ratings, this figure explains the number of recorded responses and the study can see the direct relation of the network features to the MOS and the time frame of ascension. It seems that the data presents a logical front of an increase in performance that directly leads to an increase in vMOS for the quality preserved by the users. These figures conclude that the thesis's chosen features directly satisfy the data generation for training-ready data as they show a linear increase in QoE as the network features abilities are less restricted.

4.3.3 Data Generated

First and foremost, the data generated by the thesis's experiment and network monitoring is quite noisy. "Noisy" is purposefully used in this thesis for multiple reasons. This thesis defined the generated data as noisy based on the following: In machine learning, noisy data refers to data that contains errors, inconsistencies, or random variations that deviate from the underlying patterns or trends in the data. These errors can be introduced during data collection, data preprocessing, or due to inherent variability in the data source. This thesis defined its data as noisy due to a high amount of network data's random variations and underlying patterns and trends. Now this noisy data can make it quite challenging to train models effectively because of an increased complexity: Noise adds complexity to the data by introducing

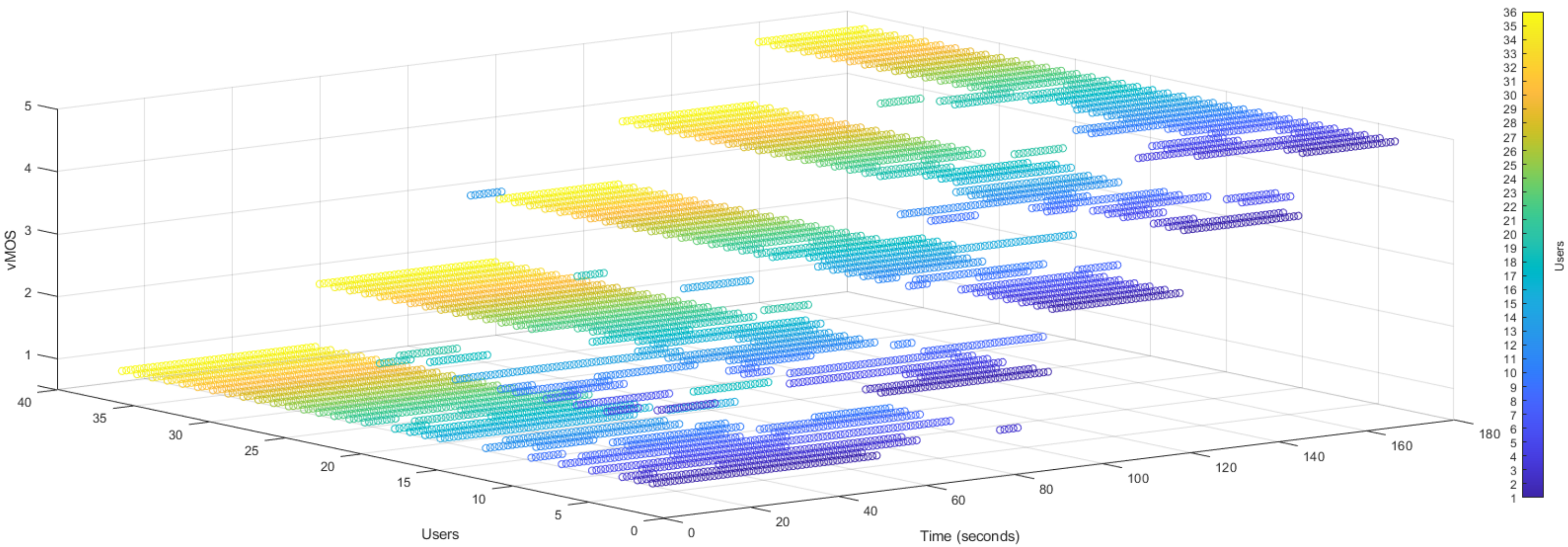


Figure 4.8: Results of vMOS experimentation trials

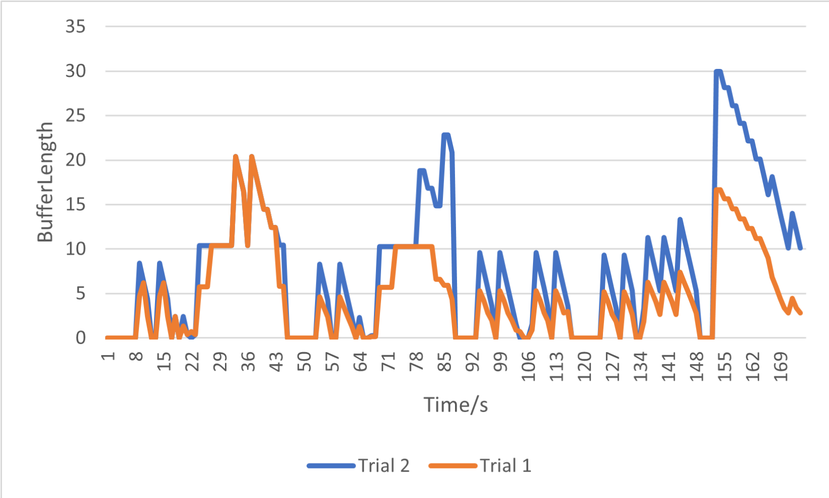


Figure 4.9: Buffer Length Comparison

irrelevant or misleading information. This can confuse the learning algorithm and make it difficult to extract meaningful patterns. The presence of noise can lead to overfitting, where the model memorizes the noise instead of capturing the true underlying patterns. Degraded generalization; Noisy data can hinder a model's ability to generalize well to unseen data. If the noise dominates the signal in the training data, the model may fail to learn the true underlying patterns and instead focus on the noise. Consequently, the model's performance may suffer when applied to new, unseen data. Loss of predictive power, in noisy data, can weaken the relationship between the input features and the target variable, reducing the model's predictive power. If the noise is substantial, it can overshadow the informative features, making it difficult for the model to make accurate predictions. All of this increases computational complexity, due to noise increases the complexity of the learning problem by introducing additional variability. This can lead to longer training times, as the model needs to process and learn from the noisy instances, potentially requiring more data or more complex algorithms to handle the noise effectively.

This thesis had to deal with such data. Dealing with noisy data requires careful preprocessing and feature engineering techniques to minimize the impact of noise. These techniques involved outlier detection and removal, filtering methods, and feature selection, to make the data more amenable to learning. Additionally, using robust machine learning algorithms that are less sensitive to noise, such as ensemble methods or regularization techniques, can help mitigate the adverse effects of noise on model training.

The provided figures 4.9, 4.10, 4.11 show a comparative analysis between "Trial 1" and "Trial 2" in the context of network attributes, specifically Buffer Length, Bitrate Downloading, and Dropped Frames, during video simulations conducted within the testbed environment. "Trial 1" represents the network features obtained without the utilization of the P4 implementation and QoE Prediction proposed in this thesis. Conversely, "Trial 2" represents the network features captured with the inclusion of this thesis's proposed implementation. Analysis of the data reveals that over time, both buffer length and bitrate

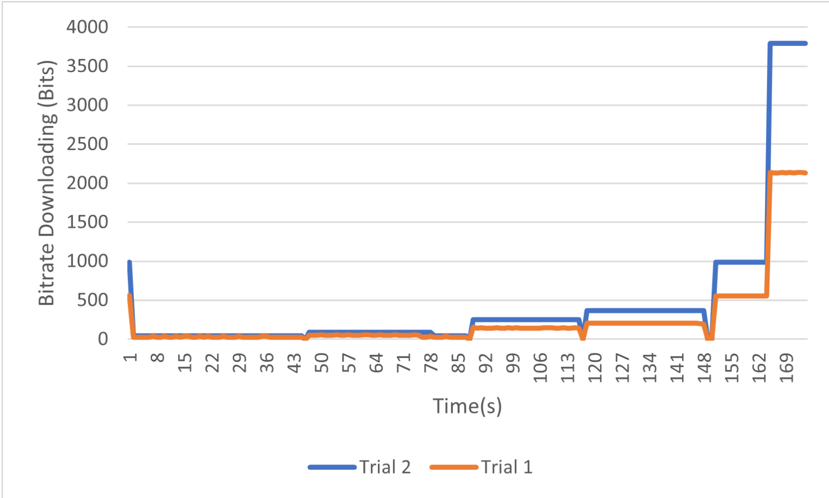


Figure 4.10: Bitrate Comparison

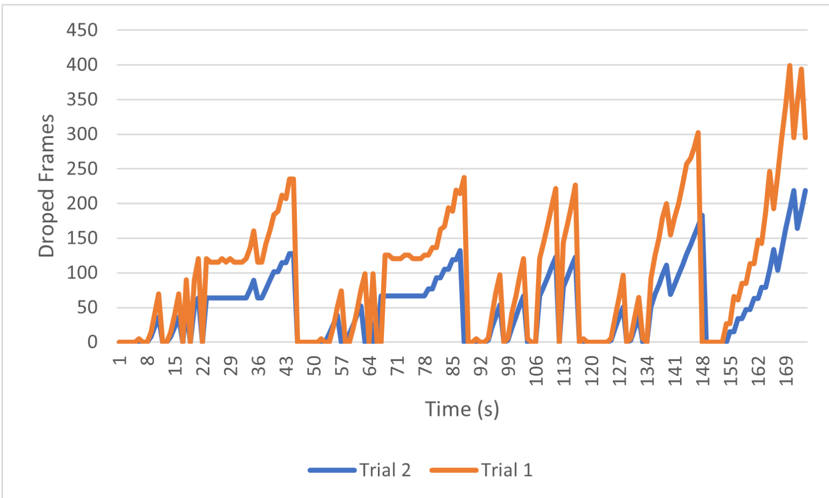


Figure 4.11: Dropped Frames Comparison

downloading exhibit an observable increase, indicating enhanced network efficiency. Additionally, as the resolution increases, a notable decrease in dropped frames is observed, further affirming the improved network efficiency at higher rates.

4.4 Chapter Summary

In this chapter, the study presents a study on predicting MOS for DASH based on network properties and video features. The study utilizes the DASHIF Reference Player Web Streaming Application for adaptive streaming and collects data on stream properties and user responses through quantitative questionnaires. The streaming server streams selected videos in fixed properties, including a collection of fixed frame rates and resolutions, based on chosen video segmentation, while fixed network properties control the experiment from a network back-end perspective. User data, including questionnaire responses, screen properties, and stream data, is captured, and an MOS rating is provided by the user based on their experience.

The study focuses on seven primary video features that are affected by network performance and can impact user satisfaction: initial buffer length, live buffer length, bitrate downloading, dropped frames, latency, and video resolution, along with three MOS ratings provided by each user while streaming. These features are used as input to the data training and aid the classification algorithms in learning and identifying patterns of user satisfaction. After analyzing the technical aspects of the experiment, the authors generate a predicted MOS based on the trained model, increasing the ease of analysis of applications and tests that require MOS prediction within the testbed without the need for more human ratings.

The study concludes by outlining the correct parameters, such as the manipulation of the resolution, that will be used in the next experiment as an editable user-choice configuration. Overall, this study provides valuable insights into predicting user satisfaction with DASH based on network and video features and can be useful for future research in this area.

CHAPTER 5

QoE Experimentation Framework Classification & Prediction

In a world of multimedia tech so vast,
Classification and prediction can be quite a task.
But with tools and frameworks to aid our sight,
We can make sense of it all and set things right.
So let us experiment, explore and learn,
And unlock the potential of tech at every turn.

Ahmed Al-Mashhadani

5.1 Introduction

QoE is a crucial metric in the evaluation of multimedia services and applications. With the rapid growth of new multimedia technologies and their wide-ranging applications, the need for QoE assessment has become increasingly important. QoE experimentation frameworks have been developed to facilitate the evaluation of QoE, providing researchers with a standardized method for conducting experiments and collecting data. In recent years, there has been a growing interest in the classification and prediction of QoE based on experimental data. This involves using machine learning techniques to analyze QoE data and develop predictive models that can accurately classify the QoE of multimedia services or applications. Such models can be used to identify the factors that affect QoE and to optimize multimedia systems and services accordingly.

This section aims to provide an overview of the thesis's results with machine learning prediction towards QoE. The researcher discusses the different machine learning techniques used for QoE classification and prediction, including their strengths and limitations. Finally, we provide the thesis's prediction results for the human experiments and will highlight some of the challenges and future research directions in this area.

5.2 Machine Learning Classification

Furthermore, a data analysing process had to happen on two different levels post the obtaining of raw data from the virtual testbed and the vMOS from the users. QoE scoring factors and network data had to go through data cleaning and normalisation. Then the features stated before were placed through training process against the MOS and vice versa to make a collection of classification predictions and place through a neural network to compare and contrast. All features were trained with Fine Tree, Medium Tree, Coarse Tree, Kernel Naive Bayes, Linear SVM, Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Medium Gaussian SVM, Fine KNN, Medium KNN, Coarse KNN, Cosine KNN, Cubic KNN, Weighted KNN, Boosted Trees, Bagged Trees, Subspace Discriminant, Subspace KNN, and RUSBoosted Trees. With no over-trained attempts, this way it can be compared and contrasted which model is best fitted for the data and the feature classified. Figure 5.1 shows an example of the data analysis process.

In Table 5.1, all classification methods mentioned above are used to classify the five features and predict their outcome if a new stream of data is inserted. The table shows the percentage of the predicted class against true class. From these models, the researcher selected the models with the highest accuracy along with the fastest prediction time. Bagged trees were selected for bit rate downloading and MOS and fine KNN for the buffer length, the dropped frames and the resolution. Table 5.2 shows the most ideal machine learning classification prediction methods on highly noisy data, such as the thesis's generated network data and MOS user feedback technique proposed in the thesis's previous paper [19]. It is important here to discuss the prediction speed along with the training time for the prediction of MOS. The training time, prediction speed and misclassification cost columns are representations of the different machine learning

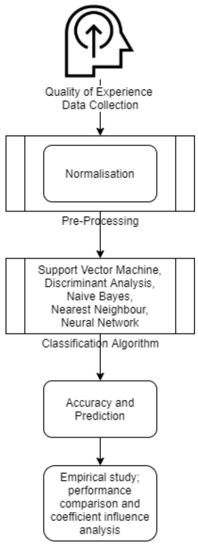


Figure 5.1: Data Analysis

classifications to predict MOS. The highest accuracy using bagged trees resulted in 10.69 s with 1200 observations processed per second.

Noisy data in machine learning is defined as data that has errors or random fluctuations in it, which makes it harder for algorithms to find patterns or predict the future with accuracy. Noisy data can appear in several ways, including:

- Data points that differ noticeably from the remainder of the dataset are referred to as errors or outliers. They could be the result of anomalies, equipment failures, or measurement errors.
- Inadequate Information; Noise in a dataset can be introduced by missing values or incomplete records, which can impact the analysis and predictions made overall.
- Incertitude or Lack of Clarity; Data can occasionally be unclear or ambiguous, which can cause discrepancies or make it harder to interpret the right information.

The noisy data in the context provided is specifically related to this thesis's generated network data and MOS user feedback technique. Such data could include complex network structures or user feedback that is inherently noisy due to a variety of factors such as:

- Network Data; Due to their complexity, generated networks may contain errors such as missing connections, inaccuracies in representing relationships, or noise introduced during the data collection process.
- MOS User Feedback; Because of factors such as individual biases, differences in perception, or errors in the feedback collection process, user feedback may contain inconsistencies, contradictions,

or imprecise ratings.

The difficulty in dealing with such noisy data is in developing machine learning models capable of effectively distinguishing meaningful patterns from noise, resulting in accurate predictions or classifications. For example, the table below (Table 5.2) compares the performance of various machine learning classification methods in predicting MOS from noisy data.

The columns for training time, prediction speed, and misclassification cost are critical metrics for assessing the efficiency and effectiveness of these classification methods in dealing with noisy data. The mention of bagged trees achieving the highest accuracy in 10.69 seconds, processing 1200 observations per second, demonstrates its ability to handle data noise efficiently while making reasonably accurate predictions for the MOS (Mean Opinion Score) with relatively fast processing speed.

5.2.1 Fine Tree Exploration

Algorithm 6: Fine Tree Algorithm for Video Quality Prediction

Input : File paths for training data and testing data

```
// Load training data from file
load trainData.mat;
// Define predictor variables and response variable
predictors = trainData(:, 1:4);
response = trainData(:, 5);
// Define fine tree model
fineTreeModel = fitrtree(predictors, response, 'MinLeafSize', 5, 'Surrogate', 'on', 'Prune', 'off',
  'MergeLeaves', 'off', 'MaxNumSplits', 10, 'PredictorNames', {'Resolution', 'BufferLength',
  'Bitrate', 'DroppedFrames'});
// View the fine tree
view(fineTreeModel, 'Mode', 'graph');
// Load testing data from file
load testData.mat;
// Use the fine tree model to predict MOS for the testing data
predictedMOS = predict(fineTreeModel, testData);
disp(predictedMOS);
```

Output: Predicted Mean Opinion Scores (MOS) for the testing data

Based on the results obtained from the Fine Tree algorithm, it can be concluded that this algorithm is particularly effective in predicting the resolution and bitrate of a video. The accuracy achieved in predicting the resolution was particularly impressive, with a rate of 98.8%. This indicates that the other features, such as buffer length, bitrate, and dropped frames, are strong predictors of resolution.

Moreover, the Fine Tree algorithm proved to be relatively effective in predicting the bitrate, with an accuracy of 97.6%. However, it should be noted that the accuracy of predicting MOS, dropped frames or buffer length is not recommended. Therefore, it is advisable to consider other algorithms that might be better suited for predicting these features. Overall, the Fine Tree algorithm proved to be a valuable tool for predicting certain features related to video quality, but it may not be suitable for predicting all aspects of video performance.

5.2.2 Medium Tree Exploration

Algorithm 7: MATLAB Medium Tree Algorithm for Predicting Resolution

Input: Buffer length, bitrate, dropped frames, user MOS

Output: Resolution

1. **Load training data from file** load('trainData.mat', 'trainData');
2. **Define predictor variables and response variable** predictors = trainData(:, 2:5); response = trainData(:, 1);
3. **Define medium tree model** mediumTreeModel = fitrtree(predictors, response, 'MinLeafSize', 10, 'Surrogate', 'on', 'Prune', 'off', 'MergeLeaves', 'off', 'MaxNumSplits', 20, 'PredictorNames', 'BufferLength', 'Bitrate', 'DroppedFrames', 'UserMOS');
4. **Load testing data from file** load('testData.mat', 'testData');
5. **Use the medium tree model to predict resolution for the testing data** testingData = [bufferLength, bitrate, droppedFrames, userMOS]; predictedResolution = predict(mediumTreeModel, testingData);
6. **Output the predicted resolution** disp(predictedResolution);

The medium tree algorithm was used to predict resolution based on buffer length, bitrate dropped frames, and user MOS. The results showed that the medium tree algorithm was highly effective for predicting resolution with 98.8% accuracy when using the other features as predictors. Additionally, the medium tree algorithm was also effective for predicting bitrate with almost identical results to the fine tree algorithm, achieving 97.6% accuracy. However, the accuracy for predicting MOS, dropped frames, and buffer length was found to be low, indicating that the medium tree algorithm is not recommended for predicting those features.

Additionally, the medium tree algorithm showed a significant improvement in predicting resolution when compared to the fine tree algorithm, with an increase in accuracy of 1.2%. However, the trade-off is that the medium tree algorithm is more complex and may require more computational resources to train and implement.

Furthermore, the low accuracy in predicting MOS, dropped frames, and buffer length using both the fine and medium tree algorithms indicates that these features may be influenced by other factors not captured by the current predictors. This highlights the importance of feature selection and engineering in developing accurate machine-learning models.

Overall, the results suggest that the medium tree algorithm is a promising approach for predicting video streaming resolution and bitrate, but further research is needed to improve the accuracy of other video quality metrics.

5.2.3 Fine KNN Exploration

Algorithm 8: MATLAB Fine KNN Algorithm for Predicting Buffer Length

Input: Buffer length, bitrate, dropped frames, user MOS

Output: Resolution

1 **Load training data from file** load('trainData.mat', 'trainData');

2 **Define predictor variables and response variable** predictors = trainData(:, 2:5); response = trainData(:, 1);

3 **Define Fine KNN model** fineKNNModel = fitcknn(predictors, response, 'NumNeighbors', 5, 'Standardize', 1, 'Distance', 'Euclidean', 'PredictorNames', 'BufferLength', 'Bitrate', 'DroppedFrames', 'UserMOS');

4 **Load testing data from file** load('testData.mat', 'testData');

5 **Use the Fine KNN model to predict buffer length for the testing data** testingData = [bufferLength, bitrate, droppedFrames, userMOS]; predictedBufferLength = predict(fineKNNModel, testingData);

6 **Output the predicted buffer length** disp(predictedBufferLength);

After running the Fine KNN algorithm, the researcher obtained some interesting results. The algorithm was highly effective in predicting buffer length, achieving an accuracy of 91.2% when using the other features as predictors. Additionally, it was also able to predict dropped frames with an accuracy of 91.8%, indicating that this algorithm is useful for identifying issues related to buffering and streaming quality.

However, it is not recommended to use the Fine KNN algorithm for predicting MOS, bitrate, or resolution, as the researcher obtained very low accuracies for these features. This suggests that the algorithm may not be suitable for more complex video quality metrics and that other techniques, such as the medium tree algorithm the researcher previously discussed, may be more appropriate. Overall, the Fine KNN algorithm is a valuable tool for identifying and diagnosing certain streaming issues, but it should be used in conjunction with other techniques to get a more comprehensive understanding of video quality.

5.2.4 Bagged Tree Exploration

Algorithm 9: Bagged Tree Algorithm for Predicting MOS

```
Input  : File paths for training data and testing data
// Load training data from file
load trainData.mat;
// Define predictor variables and response variable
predictors = trainData(:, 1:4); response = trainData(:, 5);
// Define bagged tree model
baggedTreeModel = TreeBagger(50, predictors, response, 'Method', 'classification',
  'NumPredictorsToSample', 'all', 'PredictorNames', 'Resolution', 'BufferLength', 'Bitrate',
  'DroppedFrames');
// Load testing data from file
load testData.mat;
// Use the bagged tree model to predict MOS for the testing data
predictedMOS = predict(baggedTreeModel, testData); disp(predictedMOS);
Output: Predicted Mean Opinion Scores (MOS) for the testing data
```

In this experiment, the Bagged Tree algorithm showed remarkable accuracy in predicting MOS with 99.9% accuracy using the resolution, bitrate dropped frames, and buffer length as predictors. The high accuracy suggests that this algorithm is a robust solution for predicting the user's quality of experience in video streaming applications. However, when the researcher attempted to predict other features such as bitrate, buffer length, dropped frames, or resolution, very low accuracies resulted for those features. This indicates that the Bagged Tree algorithm is not suitable for predicting those specific features in this particular dataset. Thus, this thesis recommends that the Bagged Tree algorithm be used primarily for predicting MOS in similar video streaming datasets.

The table 5.1 provides a comparison of classification metrics for different features and machine learning classifiers (MLCs). The metrics are computed based on the prediction of mean opinion scores (MOS), which is the classic metric that is commonly used in the evaluation of video quality. The MLCs considered in the table include Fine Tree, Medium Tree, Coarse Tree, Kernel Naive Bayes, Linear SVM, Quadratic SVM, Cubic SVM, Fine Gaussian SVM, Medium Gaussian SVM, Coarse Gaussian SVM, and Fine KNN. The features compared in the table include Resolution, Buffer Length, Bitrate, Dropped Frames, and MOS.

The table shows that the Fine Tree and Medium Tree MLCs perform the best across all features, with the highest resolution and bitrate scores and the lowest MOS and Total Miss-Classification Cost (TMCC) scores. These two MLCs also have the highest prediction speed and training time scores, which suggests that they are efficient in terms of computational time. The Coarse Tree MLC has the lowest resolution and bitrate scores, while the Gaussian SVM MLCs have the lowest scores for buffer length and MOS. The Kernel Naive Bayes MLC has the highest dropped frames score, which indicates that it is not well-suited for predicting video quality based on this metric. The Fine KNN MLC has the highest scores for buffer length and dropped frames, which suggests that it may be useful for predicting video quality based on these metrics. However, its overall performance in terms of MOS and TMCC is lower than that of the Fine Tree and Medium Tree MLCs.

Overall, the table shows that the Fine Tree and Medium Tree MLCs are the best performers across all features and metrics considered in this analysis.

5.2.5 Models Description & Neural Network

Table 5.1 presents a comprehensive overview of the prediction accuracy results for various classification models across multiple features. The table showcases the evaluation metrics for each model, including Resolution, Buffer Length, Bitrate, Dropped Frames, Mean Opinion Score (MOS), Total Miss-Classification Cost, Prediction Speed, and Training Time. Examining the results, it is evident that the Fine Tree model demonstrates excellent performance, achieving a high accuracy rate in predicting Resolution (98.8%) and Bitrate (97.6%). Furthermore, it exhibits a low percentage of Dropped Frames (47.1%) and an accurate estimation of MOS (99.4%). The model also exhibits a minimal Total Miss-Classification Cost (1) and operates at a moderate Prediction Speed (approximately 2300 observations per second) with a Training Time of 5.3971 seconds.

Similar observations can be made for other models, such as the Medium Tree and Fine Gaussian SVM, which also exhibit high prediction accuracy for Resolution and Bitrate. However, there are variations in their performance for other metrics, such as Dropped Frames and MOS, which may indicate differing levels of model effectiveness in capturing specific aspects of network behaviour. In contrast, models like Coarse Tree, Linear SVM, and Quadratic SVM exhibit lower prediction accuracy across multiple features. For instance, the Coarse Tree model demonstrates relatively lower accuracy in predicting Resolution (92.9%), Bitrate (84.1%), and Dropped Frames (40.0%). These results suggest that these models may not be as effective in capturing the intricate relationships between the selected features.

It is worth noting that the Kernel Naive Bayes model presents results only for Resolution and Dropped Frames, indicating limited applicability for the remaining features. The performance of the KNN models varies depending on the specific configuration, with some achieving high accuracy for features like Buffer Length and Dropped Frames, while others show lower accuracy in certain aspects. Overall, the table provides a comprehensive comparison of classification models' prediction accuracy across multiple features. These results offer valuable insights into the strengths and limitations of each model in capturing the complex dynamics of the network under analysis.

In the study conducted by the researcher, various hyper-parameters for each architecture were explored to identify the optimal fit for the proposal. The following aspects were varied: 1) the use of cross-validation folds to protect against overfitting by partitioning the data set into folds and estimating accuracy on each fold, ranging from 2 to 50 folds, 2) the number and types of splits such as diversity indexes and surrogate decision splits, 3) the training algorithm, 4) the number of layers in a network, and 5) the number of neurons for a layer. To validate the thesis's models, a random selection of 30% of the thesis's data was taken out during validation training. The classification of all features and their corresponding classes was considered, as illustrated in Table 5.1, and the use of a neural network on MOS prediction using the Bayesian Regularization with 2 layers, 10 neurons, and Mean Squared Error for performance rating. This resulted in a regression value of 0.999, indicating a highly accurate correlation between the output and the target.

Table 5.2: Resulting machine learning prediction methods on noisy network training data.

Feature/MLC Prediction	**Resolution**	**Buffer Length**	**Bitrate**	**Dropped Frames**	**MOS**
Fine Tree	98.8	Not ideal	97.6	Not ideal	Not ideal
Medium Tree	98.8	Not ideal	97.6	Not ideal	Not ideal
Fine KNN	Not ideal	91.2	Not ideal	91.8	Not ideal
Weighted KNN	Not ideal	91.2	Not ideal	91.8	Not ideal
Bagged Tree	Not ideal	Not ideal	Not ideal	Not ideal	99.9

After comparing and contrasting the classification metrics used, this thesis recommends the above-mentioned models for each feature as an accurate prediction method. The thesis's study shows that these models are effective for DASH-related streaming data and real user MOS. The neural network method with the selected options tends to be more accurate for small or noisy datasets but requires more time and computing power. It may not be efficient for placement within a testbed for auto prediction and redirection of resources.

However, using this method requires more time and computing power, making it inefficient for auto prediction and resource redirection within a test bed. When dealing with noisy data, prediction methods must be carefully chosen and tested for each feature that depends on all others. Figure 5.2 displays the precision of the bit rate framework after using bagged trees, while Figure 5.3 shows the resolution prediction results after implementing the fine tree method. These figures highlight the contrast between the two techniques when applied to the same data. This is useful for the research as it enables this work to identify correctly predicted true classes and analyze how the methods handle noisy data such as ours.

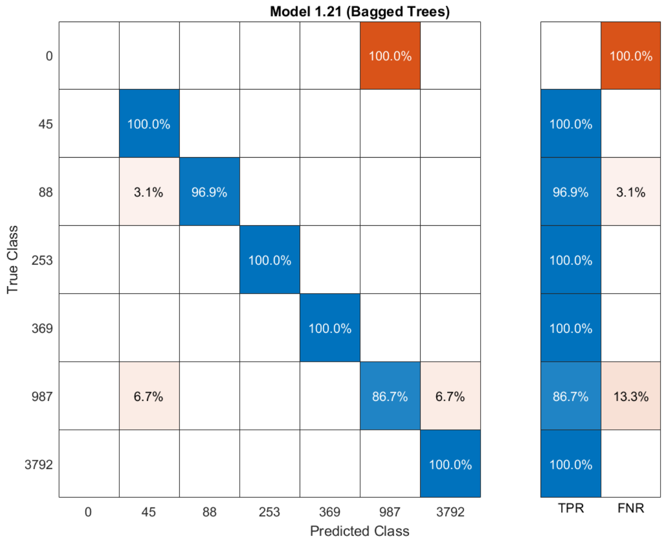


Figure 5.2: Framework bit rate prediction results.

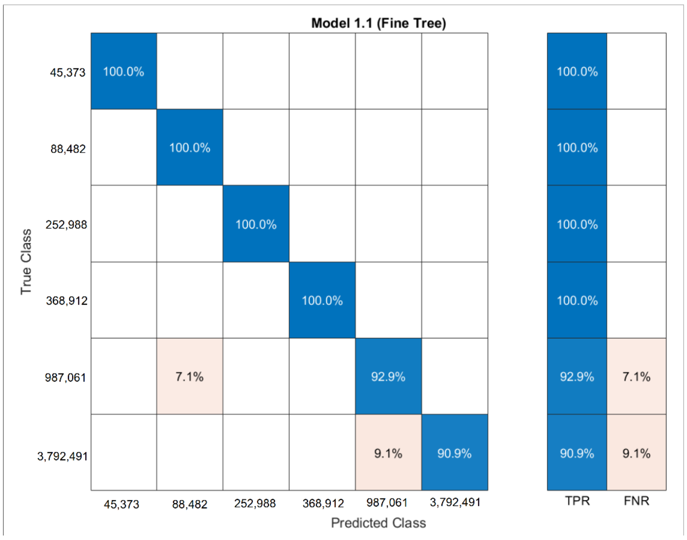


Figure 5.3: Framework resolution prediction results.

5.2.6 State-of-the-Art Comparison Discussion

It is quite difficult to compare the thesis's framework to other state-of-the-art solutions due to its multiple different outputs. However, some solutions provide parts similar to some of the thesis's outcomes. These are listed in Table 5.3. SDNDASH [144], [115] focuses on QoE requirements to build an SDN-based management and resource allocation system to maximize the quality perceived; However, their solutions do not consider the network data as a straight feature contribution for direct prediction. The thesis's approach focuses on using network data to automatically predict the QoE of a real user. With multiple network configurations, the thesis's predicted features and QoE reached new heights on known classifiers. SDNDASH focuses on a single user and recommends a certain bit rate and buffer levels, whereas the thesis's framework comes with pre-designed adjustable network features and shows the developers how each feature affects the QoE on a human level, thus eliminating testing with users. Another solution [122] proposed a network application controller called service manager, which reads video traffic and allocates resources fairly among competing congestion. The thesis's solution has elements of this step and shows the user the best segments with the most optimal resolution for their network configuration. The higher the configuration, the better the quality perceived and shown on the classifiers will be. A similar state-of-the-art solution [144] offers a caching technique where users are informed of the cache's content as well as a short-term prediction of the bottleneck bandwidth, whereas the work's framework shows the prediction for the entire duration of the segmented perceived file. QFF [145] proposed an open-flow-assisted QoE framework with optimizing QoE among HAS clients with heterogeneous device requirements. This is ideal because the device information can limit the allocation to certain devices and save the rest for devices that require more bandwidth. The work's test bed is missing this functionality, and it can be useful to adapt it. The work's framework can produce training-ready data; thus, this can be an addition to future work to enhance the test bed. However, it will not affect the prediction due to the fact that the prediction is based purely on network features.

Table 5.1: Comparing classification metrics across all features.

Feature/ MLC Prediction	Reso lution (%)	Buffer Length (%)	Bitrate (%)	Dropped Frames (%)	MOS (%)	Total Miss-Classification Cost (MOS)	Prediction Speed (obs/s) (MOS)	Training Time (s) (MOS)
Fine Tree	**98.8**	41.2	**97.6**	47.1	99.4	1	~2300	5.3971
Medium Tree	**98.8**	42.4	**97.6**	47.1	99.4	1	~2400	4.7336
Coarse Tree	92.9	41.2	84.1	40.0	99.4	1	~2400	3.7631
Kernel Naive Bayes	97.6	N.A.	N.A.	60.6	99.4	1	~360	12.252
Linear SVM	74.1	44.1	66.5	42.9	79.4	35	~1600	7.2777
Quadratic SVM	78.2	88.2	70.0	82.4	79.4	35	~1600	7.0385
Cubic SVM	78.2	88.2	70.0	86.5	79.4	35	~2000	6.3877
Fine Gaussian SVM	77.6	38.8	65.3	77.6	79.4	35	~2100	7.0037
Medium Gaussian SVM	78.2	38.8	65.9	46.5	79.4	35	~2000	7.5385
Coarse Gaussian SVM	70.6	39.4	55.9	44.7	79.4	35	~1700	7.4505
Fine KNN	78.2	**91.2**	78.2	**91.8**	79.4	35	~4200	8.1421
Medium KNN	75.9	40.0	78.2	41.2	79.4	35	~4000	8.0454
Coarse KNN	32.4	28.2	31.2	28.2	26.5	125	~3900	7.9715
Cosine KNN	75.9	40.0	78.2	41.2	79.4	35	~5700	7.9047
Cubic KNN	75.9	40.0	78.2	41.2	79.4	35	~2200	8.2537
Weighted KNN	78.2	**91.2**	78.2	**91.8**	79.4	35	~6000	8.1849
Boosted Trees	32.4	48.8	96.5	61.8	26.5	125	~4400	9.1509
Bagged Trees	97.1	36.5	**97.6**	39.4	**99.9**	0	~1200	10.69
Subspace Discriminant	81.2	30.6	52.9	78.2	86.5	22	~610	11.817
Subspace KNN	84.7	85.3	82.4	70.0	85.9	32	~560	11.567
RUSBoosted Trees	85.3	60.0	53.5	57.1	27.6	125	~9200	11.404

Table 5.3: State-of-the-Art Comparison

Solution	Approach	Network	Prediction	HAS Strategy	SDN Add-On	Weakness	Asset	Resolution Adaptation
Bhat [25], Bentaleb [26]	Hybrid	Fixed	No	Bit Rate Recommendation and buffer level	Internal and external SDN-based resource management components	Outdated User Communication interface	Optimized QoE per user.	No
Kleinrouweler [27]	Hybrid	Fixed	No	Chosen bit rates pushed to each user	HAS Aware Service Manager	Users have to manually cooperate with the service manager	Explicit adaptation assistance with fairness criteria	No
Bhat [25]	Hybrid	Fixed	ARIMA short term prediction	User assisted with information about cache location and link bandwidth	SABR Module	Overhead due to bandwidth and cache occupancy monitoring	Video segment decision remains at the user's control (scalable)	No
Bhattacharyya [28]	Hybrid	Fixed	No	Optimum bit rate that ensures fairness pushed to user	Orchestrating OpenFlow Module	Utility functions need to be precalculated and stored for all video content at each resolution	Optimized QoE, Heterogeneity support, Fairness	No
Liotou [29]	Bit Rate Guidance	Mobile	Longer-term (cluster based)	Rate-guided, prediction-based	QoE-SDN APP	Assumes VSP- MNO collaboration	Network Exposure feedback enabled, no change needed at HAS clients	No
The researcher's Experimentation Framework	Hybrid Resolution Adaptation for best QoE Prediction	Fixed, Virtual	Prediction for overall QoE using video's features and vice versa prediction for video's features for further experimentation design on network level.	Initial buffering, stalling and switching influenced by network restrictions.	P4 based dataplane configurations to customize network limitations depending on experimentations to export QoE training-ready data.	Overall predictions are exterior to the SDN Testbed	Framework produces network features and QoE noisy data in training ready status along with recommended classifiers for each feature. Open-source Test bed image for recreation and further testings.	Yes

5.3 Conclusions & Chapter Summary

This research concludes by answering and researching multiple questions about the quality of experience and its prediction. Firstly, the researcher aimed to ease the replication of this research by building a new simulation environment for future researchers in similar fields. This simulation can run both P4 and Openflow, along with Python 2 and 3 instances, DASH and Mininet. With all essential packages installed, an error-free test environment for running this and other related experiments and applying previous and similar work to compare and contrast results is installed. Furthermore, with this build, a P4 SDN test bed over Mininet was designed with the ability to control DASH initial buffering, stalling, switching, monitoring, bit rate adaptation and bandwidth limitation over selected ports. The test bed provides the capacity for comprehensive user experiments and data collections, which lead to the work's insights and analysis of congestion for congestion-related experiments, along with full reconfigurability over data plane. This being an open-source project clears the difficulties and eases the research methodology into media streaming applications, tests and experiments. Was it possible to create a hybrid environment for testing on a user level and predicting with machine learning by simply inserting the network or user feedback data to generate the other? With this open-source project, the researcher cleared that hurdle. Moreover, this thesis proposed an experimentation framework structure through programmable network management for the generation of ML training-ready data and MOS/QoE prediction with the aid of a human experiment with QoE MOS-based feedback to benchmark the accuracy of the predicted QoE and network features. The work's analysis of state-of-the-art machine learning algorithms, along with the creation of the work's framework for feature evaluation in network experiments is unique in its attribute and network design; however, it is reprogrammable to match any type of media experimentation. Understanding user-effective network data has become as crucial as the QoE on individual user devices. This paper discusses different features of network-level experimentation and the prediction of QoE in multimedia applications. Additionally, the discussion encompassed the interconnection between perceivable Quality of Experience (QoE) and resource allocation, as well as traffic engineering at the network level. The potential role of emerging programmable networks like SDN was explored as a means to enhance user feedback and utilize the gathered data for predicting human feedback. An automated data collection and prediction framework is also proposed to harness the capabilities of new network designs and the growing availability of computing resources in future networks for fairness-aware content distribution. As shown in Table 5.1, the ML classification methods are used and compared for the features that are used for prediction. It can be concluded that the work's generated dataset and experimentation framework can support the development of a high-accuracy machine learning model for QoE estimation. The use of existing neural networks also proved effective with the work's data but consumed more computational power and time as mentioned in the previous section. The work's future work will look into implementations of this QoE/MOS predictor in a live smart home environment.

CHAPTER 6

Conclusions, Remarks & Future Work

The journey of technology, oh what a ride,
Full of twists and turns, with no place to hide.
We've reached the end, our goals in sight,
But there's still much to do, before we call it a night.
Conclusions are drawn, remarks to be made,
Lessons to learn, mistakes to evade.
The future is bright, with progress to be found,
Innovation and creativity, will always be around.
We leave behind a legacy, of technology's past,
A foundation for the future, that's built to last.
So let us look forward, to what lies ahead,
And continue to innovate, with passion and stead.

Ahmed Al-Mashhadani

6.1 Research Questions

In conclusion, this thesis has successfully addressed the challenges arising from the escalating demand for the online distribution of high-quality and high-throughput content. The proliferation of media applications vying for network resources has had a substantial impact on both network efficiency and the QoE perceived by users. In a multi-user multi-device environment, the measurement and maintenance of perceivable user feedback and fairness have become vital considerations alongside individual user application QoE attainment. To surmount these challenges, this thesis has proposed innovative framework designs utilizing programmable networks, particularly SDN. Through the automation of perceptible user feedback measurement and maintenance, the proposed framework has facilitated improvements in both QoE and network efficiency. Notably, the framework incorporates ML techniques to construct a prediction model based on subjective user experiments, obviating the need for physical experiments and streamlining QoE prediction processes.

Moreover, the framework has examined adaptive streaming within a software-defined network environment, encompassing the evaluation and analysis of media streams, pertinent influencing factors, and the network infrastructure. By scrutinizing network features and their direct correlation with perceived QoE, the framework has achieved enhanced network optimization and reduced QoE disparities among user devices. Lastly, this thesis has delved into the management of fairness at both application and human levels concerning networked multimedia applications. It has demonstrated how novel network designs employing programmable networks like SDN can effectively address fairness concerns. By concurrently ensuring fairness and QoE, the proposed framework offers a comprehensive solution to the challenges presented by the mounting demand for the online distribution of high-quality and high-throughput content. In conclusion, this thesis has made contributions to advancing the field of online content distribution, particularly in optimizing network efficiency, predicting QoE, and managing fairness. The proposed framework, founded on programmable networks and ML techniques, has the potential to revolutionize the delivery of high-quality content while ensuring an equitable user experience.

The central findings and contributions of this thesis are summarized and organized by the research questions that served as inspiration for this study:

- **Question I** What is Fairness in media technology and the relationship between Fairness and the Quality of Experience? Contribution 1: This thesis has made contributions to the advancement of knowledge and understanding of fairness in online multimedia applications. Firstly, existing definitions of fairness were reviewed and their applicability in the context of online multimedia was analyzed, yielding valuable insights into the conceptualization and measurement of fairness in this domain. Secondly, a fairness flow model was developed to assess fairness in online multimedia applications across various levels, providing a comprehensive framework for evaluating fairness, this is shown in Chapter 2. Thirdly, a novel consideration of human-level fairness was introduced, emphasizing the importance of incorporating the human perspective in the design of fair multimedia systems. Lastly, the impact of human-to-computer fairness on media applications was investigated, offering valuable insights into enhancing fairness by addressing user needs and preferences, this is shown in Chapter 3. This contribution has been published in the paper titled "A Study on Fairness in Online Multimedia Applications" [15].

- **Question II** How to measure/classify Fairness and relate it to QoE? Contribution 2: Furthermore, the study has developed various tools and resources to facilitate the evaluation of fairness in multimedia systems. Firstly, a Network Inspection Tool and a QoE Measurement Tool were created to accurately monitor network performance and user experience, enabling researchers to assess fairness in online multimedia applications reliably and comprehensively, this is shown in Chapter 3. Secondly, a segmented content database was developed, encompassing a wide range of encoding configurations and resolutions. This database serves as a valuable resource for evaluating multimedia systems and conducting experiments, this is shown in Chapter 4. Thirdly, a hybrid simulation environment was established using P4 and OpenFlow, providing a flexible and customizable platform to evaluate the effectiveness of proposed solutions, this is shown in Chapter 3. Collectively, these contributions form a comprehensive framework for evaluating fairness in online multimedia applications and offer valuable tools and resources for conducting research in this field. The researcher's open-source test bed configuration and database can be found in the provided link [18]. This contribution has been published in this paper [19].

- **Question III** How is QoE preserved from SDN fairness and how can it be predicted most efficiently? Contribution 3: The third contribution of this research focuses on enhancing the accuracy of QoE predictions in multimedia systems using machine learning (ML) techniques. Firstly, a framework structure for experimentation was proposed, leveraging programmable network management to generate ML training-ready data and predict MOS/QoE, this is shown in Chapter 4. This practical solution enables more precise QoE predictions, leading to an enhanced user experience. To evaluate the effectiveness of this proposed solution, human experiments were conducted, gathering QoE MOS-based feedback. This benchmarking process served as a reliable and accurate method for measuring the accuracy of predicted QoE and network features, this is shown in Chapter 4. The obtained results from these experiments were utilized to validate the proposed experimentation framework and affirm its effectiveness in improving QoE predictions, this is shown in Chapter 5. Moreover, a comprehensive analysis of state-of-the-art machine learning algorithms was conducted, and an experimentation framework for feature evaluation in network experiments was developed. This framework provides valuable insights into how machine learning can contribute to enhancing fairness in multimedia systems. By identifying the most effective ML algorithms and features for predicting QoE, which is shown in Chapter 5, this research contributes to the advancement of more efficient and accurate multimedia systems. These contributions and findings have been published in this published journal paper [20].

Therefore, the primary focus of this thesis was to advance the testbed environment, enabling more efficient experimentation, and develop a new database specifically designed for multimedia video-related applications in research. By training the QoE prediction model using real human data and network features, the need for future real experiments with actual users to evaluate QoE is eliminated. Additionally, the open-source testbed offers the flexibility to conduct numerous experiments in various environments, while the proposed P4 load balancing ensures network fairness and data efficiency.

6.2 Future Work

Although this thesis has made contributions to addressing the challenges in online content distribution, optimizing network efficiency, predicting QoE, and managing fairness, several areas could be explored further in future research. These include:

- Integration of emerging technologies: As technology continues to evolve, it is crucial to consider the integration of emerging technologies into the proposed framework. Exploring the potential of technologies such as edge computing, 5G networks, virtual reality (VR), and augmented reality (AR) could further enhance the delivery of high-quality content and improve user experience.
- Enhancing fairness assessment: While this thesis has provided a comprehensive framework for evaluating fairness in online multimedia applications, further research can be conducted into more nuanced aspects of fairness. This includes investigating the impact of bias and discrimination in multimedia systems and developing techniques to mitigate their effects. Additionally, exploring fairness metrics and algorithms specific to different types of multimedia applications other than video-focused could lead to more accurate and tailored fairness assessments.
- User-centric design: While the proposed framework considers user feedback and human-level fairness, there is room for further exploration of user-centric design principles. Future work could involve incorporating user preferences, behaviours, and context into the design and optimization of multimedia systems. This could lead to personalized content delivery and tailored QoE prediction models that cater to individual user needs and preferences.
- Deployment and scalability: To validate the effectiveness of the proposed framework in real-world scenarios, it is crucial to deploy and test it in larger-scale networks and diverse environments. Future research can focus on deploying the framework in collaboration with industry partners or deploying it in cloud-based environments to assess its scalability, performance, and compatibility with existing infrastructure.
- Security and privacy considerations: With the increasing concerns around data privacy and security, future work should explore techniques to ensure the privacy and security of user data in the proposed framework. This may involve developing privacy-preserving algorithms for QoE prediction, implementing secure communication protocols, and adhering to privacy regulations and standards.

By addressing these areas in future research, we can further advance the field of online content distribution, network optimization, QoE prediction, and fairness management, ultimately enhancing the user experience in multimedia systems.

6.3 Remarks & Thesis Summary

This research concludes by answering and researching multiple questions about the quality of experience and its prediction. Firstly, the researcher aimed to ease the replication of this research by building a new simulation environment for future researchers in similar fields. This simulation can run both P4 and Openflow, along with Python 2 and 3 instances, DASH and Mininet. With all essential packages installed, an error-free test environment for running this and other related experiments and applying previous and similar work to compare and contrast results is installed. Furthermore, with this build, a P4 SDN test bed over Mininet was designed with the ability to control DASH initial buffering, stalling, switching, monitoring, bit rate adaptation and bandwidth limitation over selected ports. The test bed provides the capacity for comprehensive user experiments and data collections, which lead to this work's insights and analysis of congestion for congestion-related experiments, along with reconfigurability over the data plane. This being an open-source project clears the difficulties and eases the research methodology into media streaming applications, tests and experiments. Was it possible to create a hybrid environment for testing on a user level and predicting with machine learning by simply inserting the network or user feedback data to generate the other? With this open-source project, the researcher cleared that hurdle. Moreover, this thesis proposed an experimentation framework structure through programmable network management for the generation of ML training-ready data and MOS/QoE prediction with the aid of a human experiment with QoE MOS-based feedback to benchmark the accuracy of the predicted QoE and network features. This work's analysis of state-of-the-art machine learning algorithms, along with the creation of this work's framework for feature evaluation in network experiments is unique in its attribute and network design; however, it is reprogrammable to match any type of media experimentation. Understanding user-effective network data has become as crucial as the QoE on individual user devices. This paper discusses different features of network-level experimentation and the prediction of QoE in multimedia applications. The researcher also discussed how perceivable QoE is linked to resource allocation and traffic engineering at the network level how emerging programmable networks such as SDN can be used as a tool to improve user feedback and how that data can be used to predict human feedback. An automated data collection and prediction framework is also proposed to harness the capabilities of new network designs and the growing availability of computing resources in future networks for fairness-aware content distribution. As shown in Table 5.1, the ML classification methods are used and compared for the features that are used for prediction. This thesis concludes that our generated dataset and experimentation framework can support the development of a high-accuracy machine learning model for QoE estimation. The use of existing neural networks also proved effective with our data but consumed more computational power and time as mentioned in the previous section. Our future work will look into implementations of this QoE/MOS predictor in a live smart home environment.

CHAPTER 7

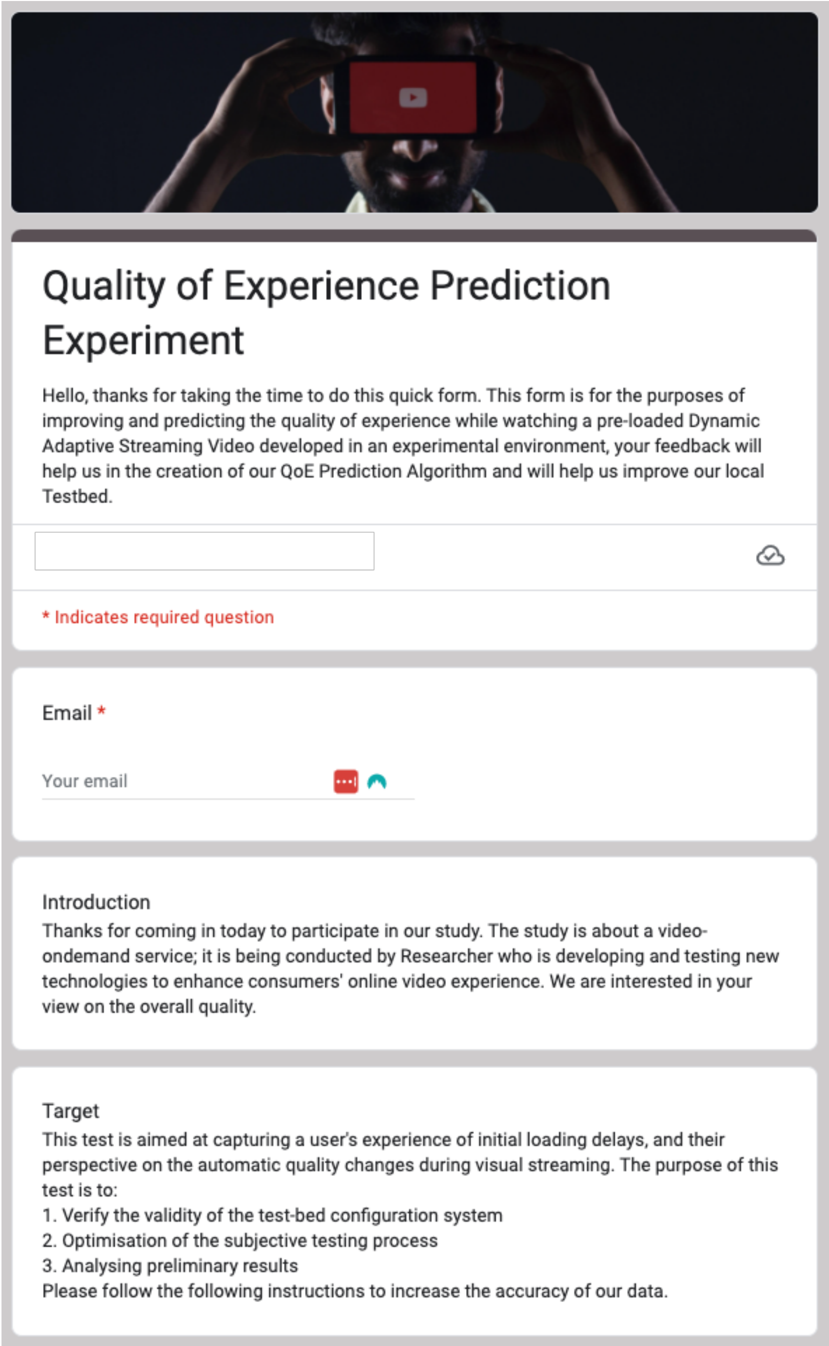
Appendix

"In my thesis, an appendix appears,
A realm of knowledge, unknown and clear.
Supplemental treasures, charts and sources,
Enriching the work with added forces."

Ahmed Al-Mashhadani

7.1 Appendix A

7.1.1 QoE Experiment Form



Quality of Experience Prediction Experiment

Hello, thanks for taking the time to do this quick form. This form is for the purposes of improving and predicting the quality of experience while watching a pre-loaded Dynamic Adaptive Streaming Video developed in an experimental environment, your feedback will help us in the creation of our QoE Prediction Algorithm and will help us improve our local Testbed.

* Indicates required question

Email *

Your email

Introduction

Thanks for coming in today to participate in our study. The study is about a video-ondemand service; it is being conducted by Researcher who is developing and testing new technologies to enhance consumers' online video experience. We are interested in your view on the overall quality.

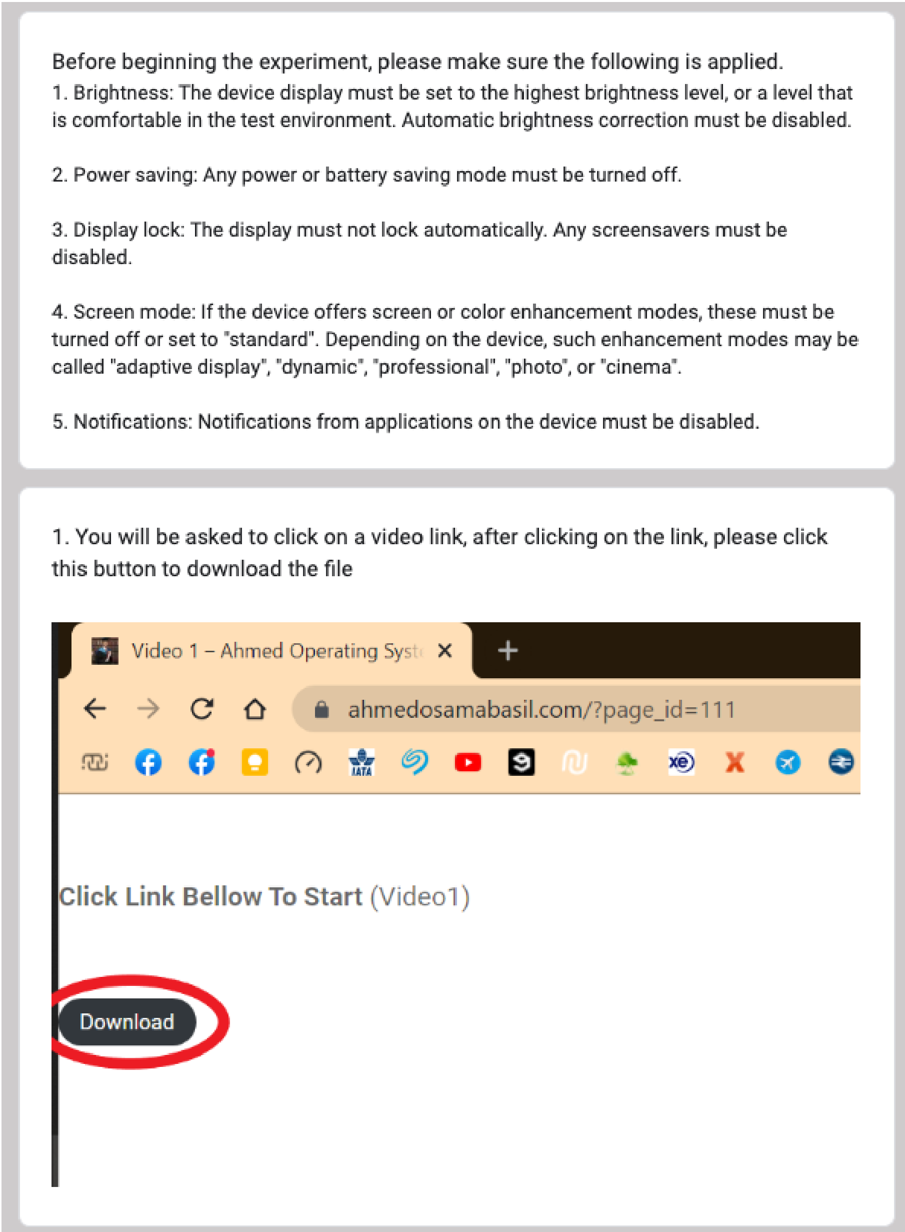
Target

This test is aimed at capturing a user's experience of initial loading delays, and their perspective on the automatic quality changes during visual streaming. The purpose of this test is to:

1. Verify the validity of the test-bed configuration system
2. Optimisation of the subjective testing process
3. Analysing preliminary results

Please follow the following instructions to increase the accuracy of our data.

Figure 7.1: Experimentation Form First Page



Before beginning the experiment, please make sure the following is applied.
1. Brightness: The device display must be set to the highest brightness level, or a level that is comfortable in the test environment. Automatic brightness correction must be disabled.

2. Power saving: Any power or battery saving mode must be turned off.

3. Display lock: The display must not lock automatically. Any screensavers must be disabled.

4. Screen mode: If the device offers screen or color enhancement modes, these must be turned off or set to "standard". Depending on the device, such enhancement modes may be called "adaptive display", "dynamic", "professional", "photo", or "cinema".

5. Notifications: Notifications from applications on the device must be disabled.

1. You will be asked to click on a video link, after clicking on the link, please click this button to download the file

Video 1 – Ahmed Operating Syst

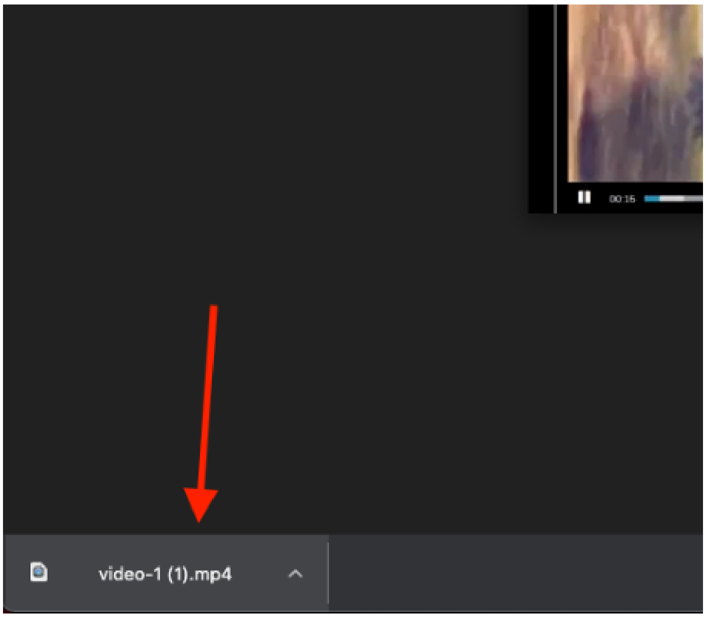
ahmedosamabasil.com/?page_id=111

Click Link Bellow To Start (Video1)

Download

Figure 7.2: Experimentation Form Second Page

2. Click and Open the downloaded video



3. Click the Play button. If you notice delay in the recorded video, please reflect it in your score. Note that the start of the recorded video is perfectly synchronised with the start of the recording.

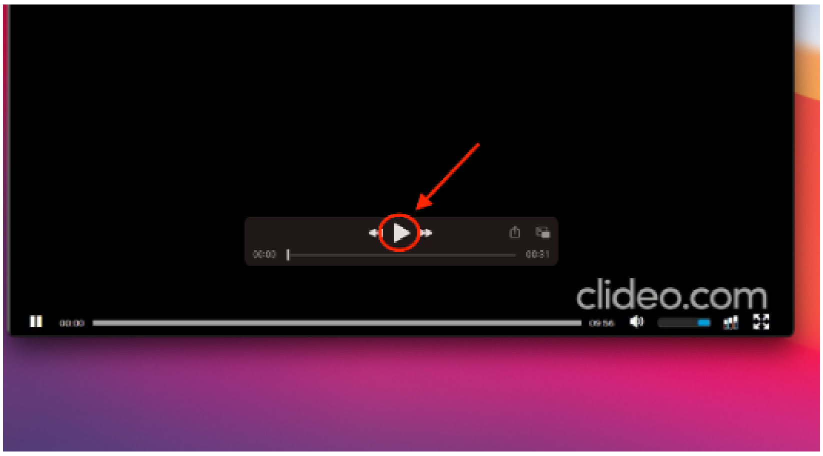
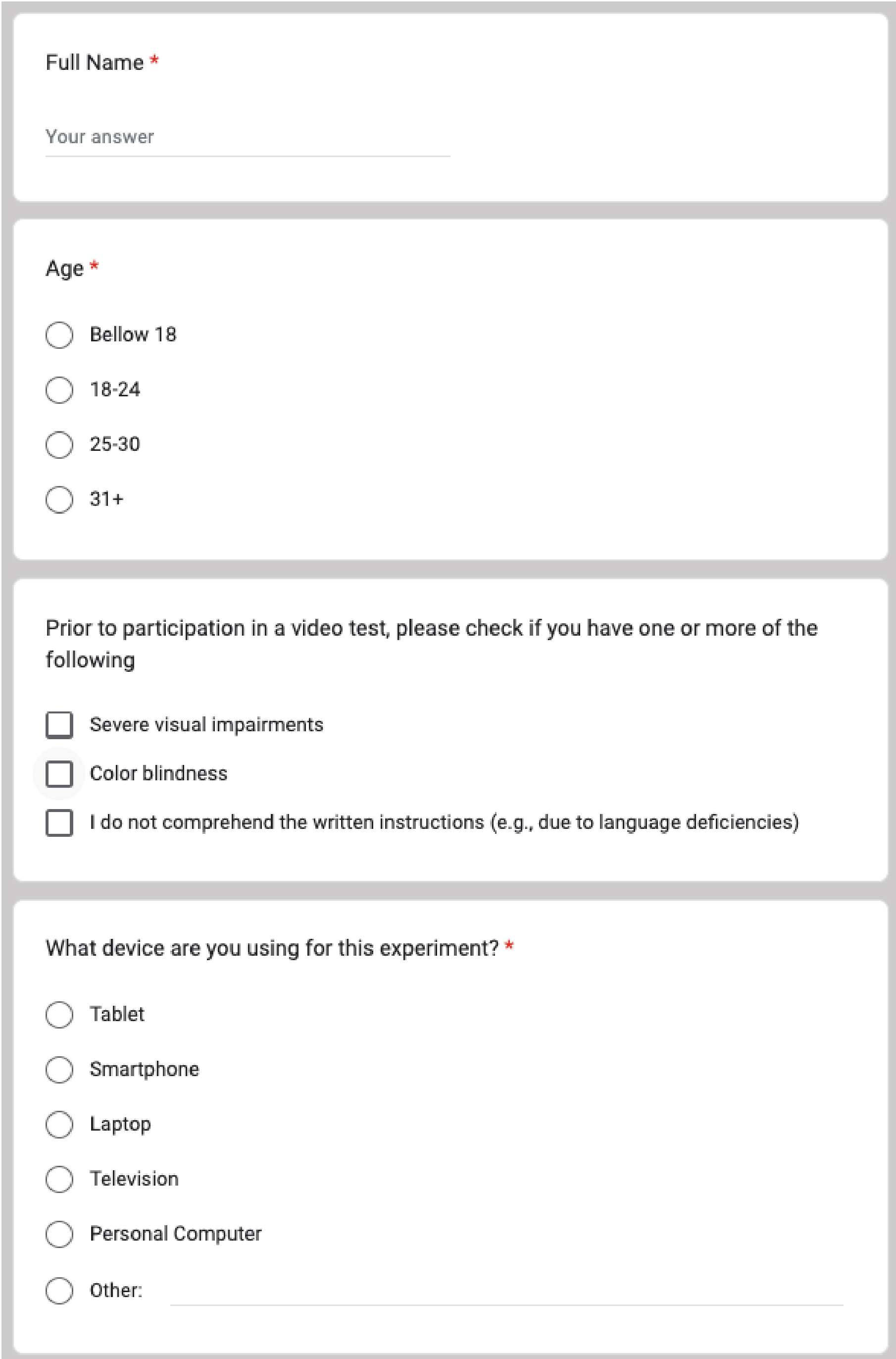


Figure 7.3: Experimentation Form Third Page



Full Name *

Your answer

Age *

- Bellow 18
- 18-24
- 25-30
- 31+

Prior to participation in a video test, please check if you have one or more of the following

- Severe visual impairments
- Color blindness
- I do not comprehend the written instructions (e.g., due to language deficiencies)

What device are you using for this experiment? *

- Tablet
- Smartphone
- Laptop
- Television
- Personal Computer
- Other:

Figure 7.4: Experimentation Form Forth Page

What is your screen resolution settings? *

- HD
- FHD
- QHD
- 4K
- 8K

How fast is your internet connection at home? *

- Less than 2 Mbps
- 2-10 Mbps
- 10-50 Mbps
- 50-100 Mbps
- 100+
- Not Known

Rating Information

The Mean Opinion Score is expressed as a single rational number, typically in the range 1–5, where 1 is lowest perceived quality, and 5 is the highest perceived quality.

Video 1 out of 5

Video Link: https://ahmedosamabasil.com/?page_id=111

Figure 7.5: Experimentation Form Fifth Page

Video 1 out of 5
Video Link: https://ahmedosamabasil.com/?page_id=111

How was the quality of your initial loading experience? *

- ◯ 1 (Bad)
- ◯ 2 (Poor)
- ◯ 3 (Fair)
- ◯ 4 (Good)
- ◯ 5 (Excellent)

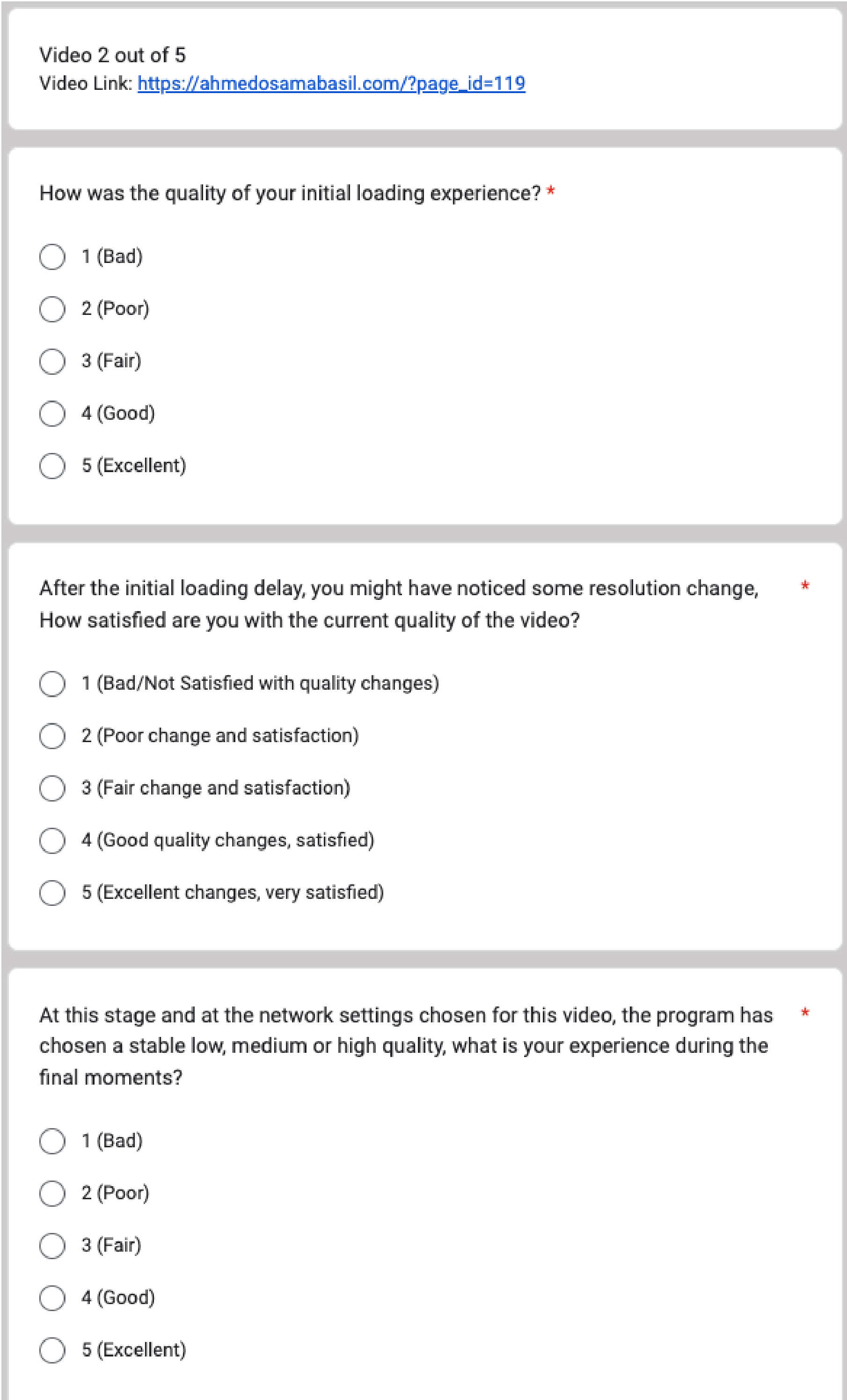
After the initial loading delay, you might have noticed some resolution change, How satisfied are you with the current quality of the video? *

- ◯ 1 (Bad/Not Satisfied with quality changes)
- ◯ 2 (Poor change and satisfaction)
- ◯ 3 (Fair change and satisfaction)
- ◯ 4 (Good quality changes, satisfied)
- ◯ 5 (Excellent changes, very satisfied)

At this stage and at the network settings chosen for this video, the program has chosen a stable low, medium or high quality, what is your experience during the final moments? *

- ◯ 1 (Bad)
- ◯ 2 (Poor)
- ◯ 3 (Fair)
- ◯ 4 (Good)
- ◯ 5 (Excellent)

Figure 7.6: Experimentation Form Sixth Page



Video 2 out of 5
Video Link: https://ahmedosamabasil.com/?page_id=119

How was the quality of your initial loading experience? *

- 1 (Bad)
- 2 (Poor)
- 3 (Fair)
- 4 (Good)
- 5 (Excellent)

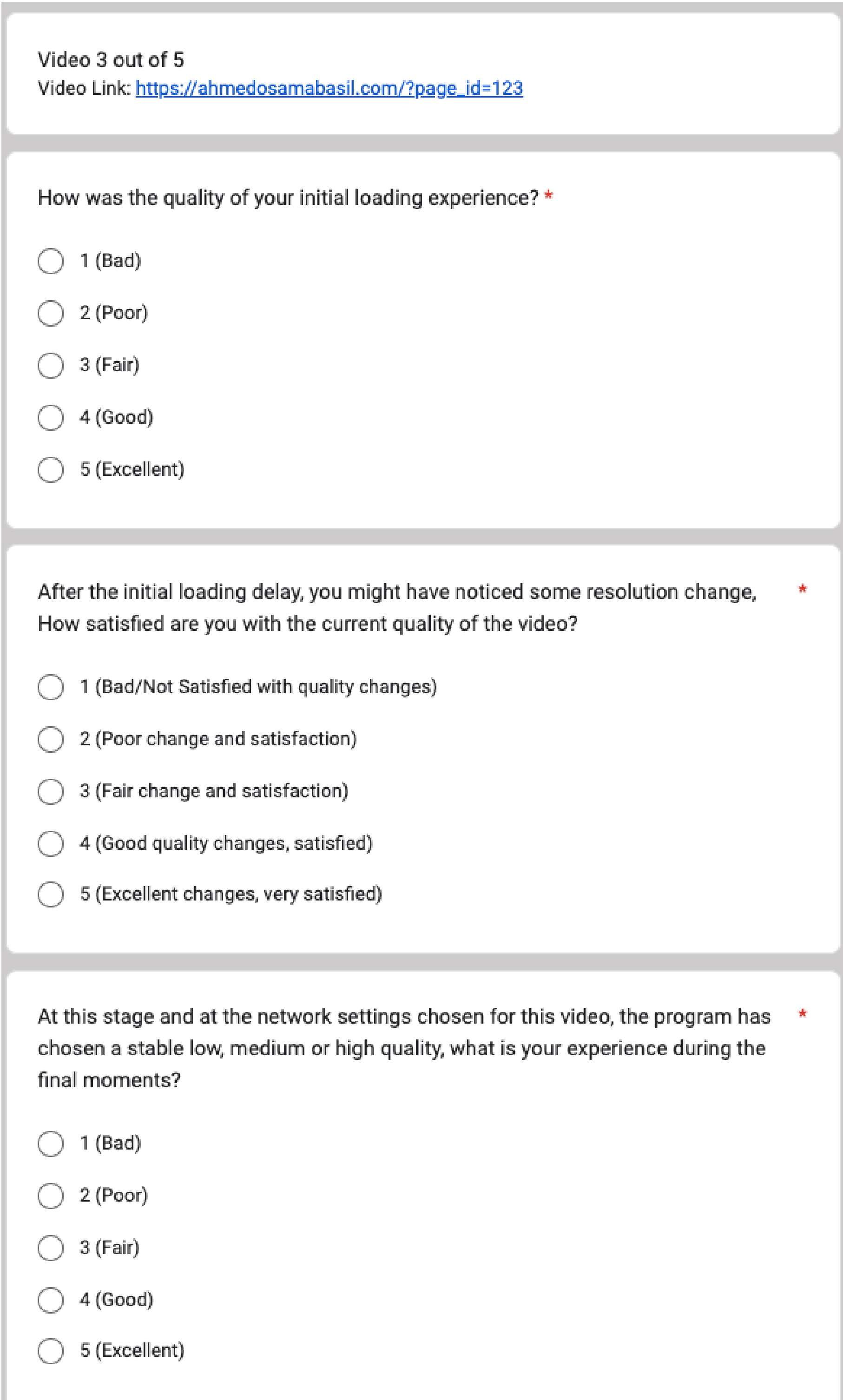
After the initial loading delay, you might have noticed some resolution change, How satisfied are you with the current quality of the video? *

- 1 (Bad/Not Satisfied with quality changes)
- 2 (Poor change and satisfaction)
- 3 (Fair change and satisfaction)
- 4 (Good quality changes, satisfied)
- 5 (Excellent changes, very satisfied)

At this stage and at the network settings chosen for this video, the program has chosen a stable low, medium or high quality, what is your experience during the final moments? *

- 1 (Bad)
- 2 (Poor)
- 3 (Fair)
- 4 (Good)
- 5 (Excellent)

Figure 7.7: Experimentation Form Seventh Page


Video 3 out of 5
Video Link: https://ahmedosamabasil.com/?page_id=123

How was the quality of your initial loading experience? *

- 1 (Bad)
- 2 (Poor)
- 3 (Fair)
- 4 (Good)
- 5 (Excellent)

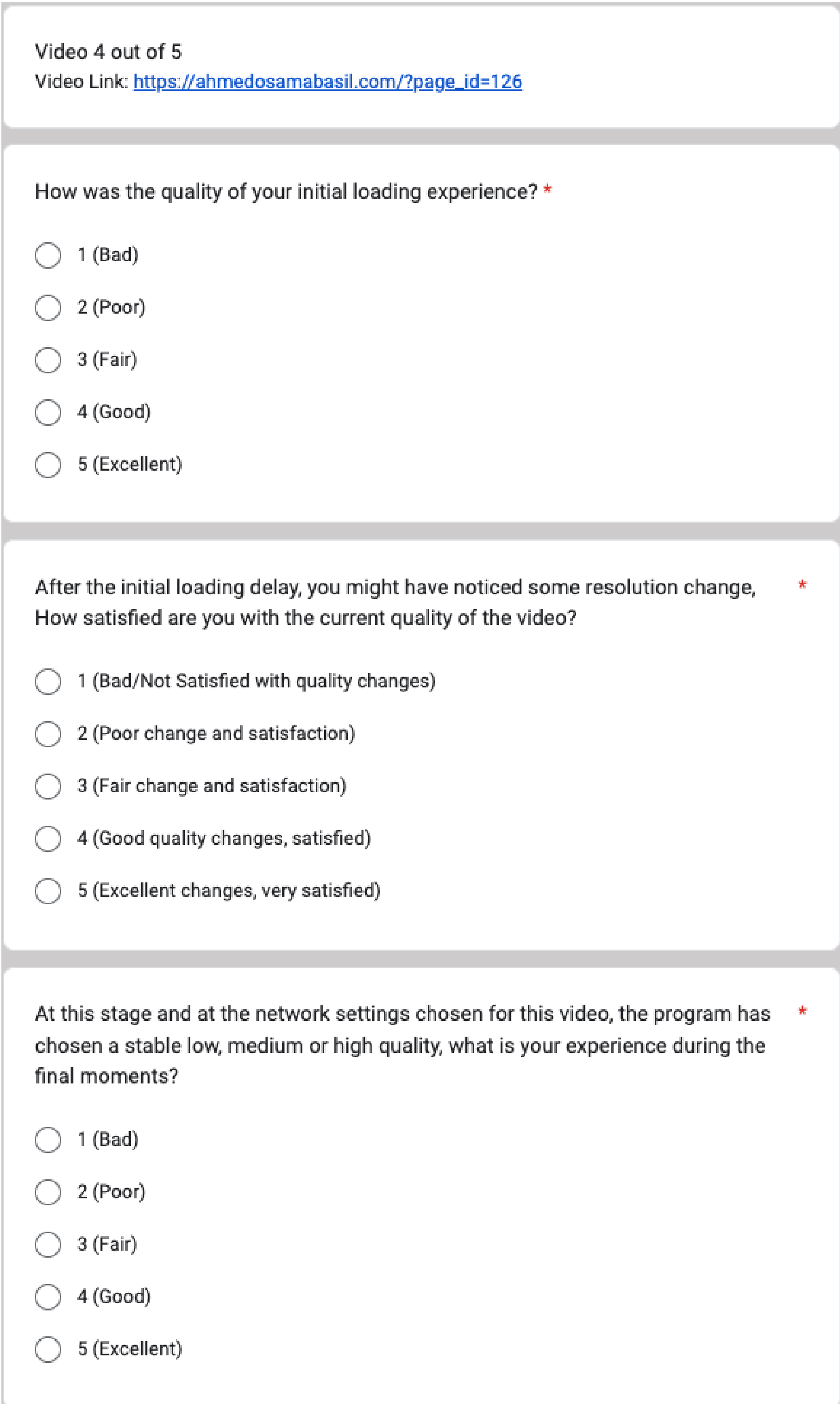
After the initial loading delay, you might have noticed some resolution change, How satisfied are you with the current quality of the video? *

- 1 (Bad/Not Satisfied with quality changes)
- 2 (Poor change and satisfaction)
- 3 (Fair change and satisfaction)
- 4 (Good quality changes, satisfied)
- 5 (Excellent changes, very satisfied)

At this stage and at the network settings chosen for this video, the program has chosen a stable low, medium or high quality, what is your experience during the final moments? *

- 1 (Bad)
- 2 (Poor)
- 3 (Fair)
- 4 (Good)
- 5 (Excellent)

Figure 7.8: Experimentation Form Eighth Page



Video 4 out of 5
Video Link: https://ahmedosamabasil.com/?page_id=126

How was the quality of your initial loading experience? *

- 1 (Bad)
- 2 (Poor)
- 3 (Fair)
- 4 (Good)
- 5 (Excellent)

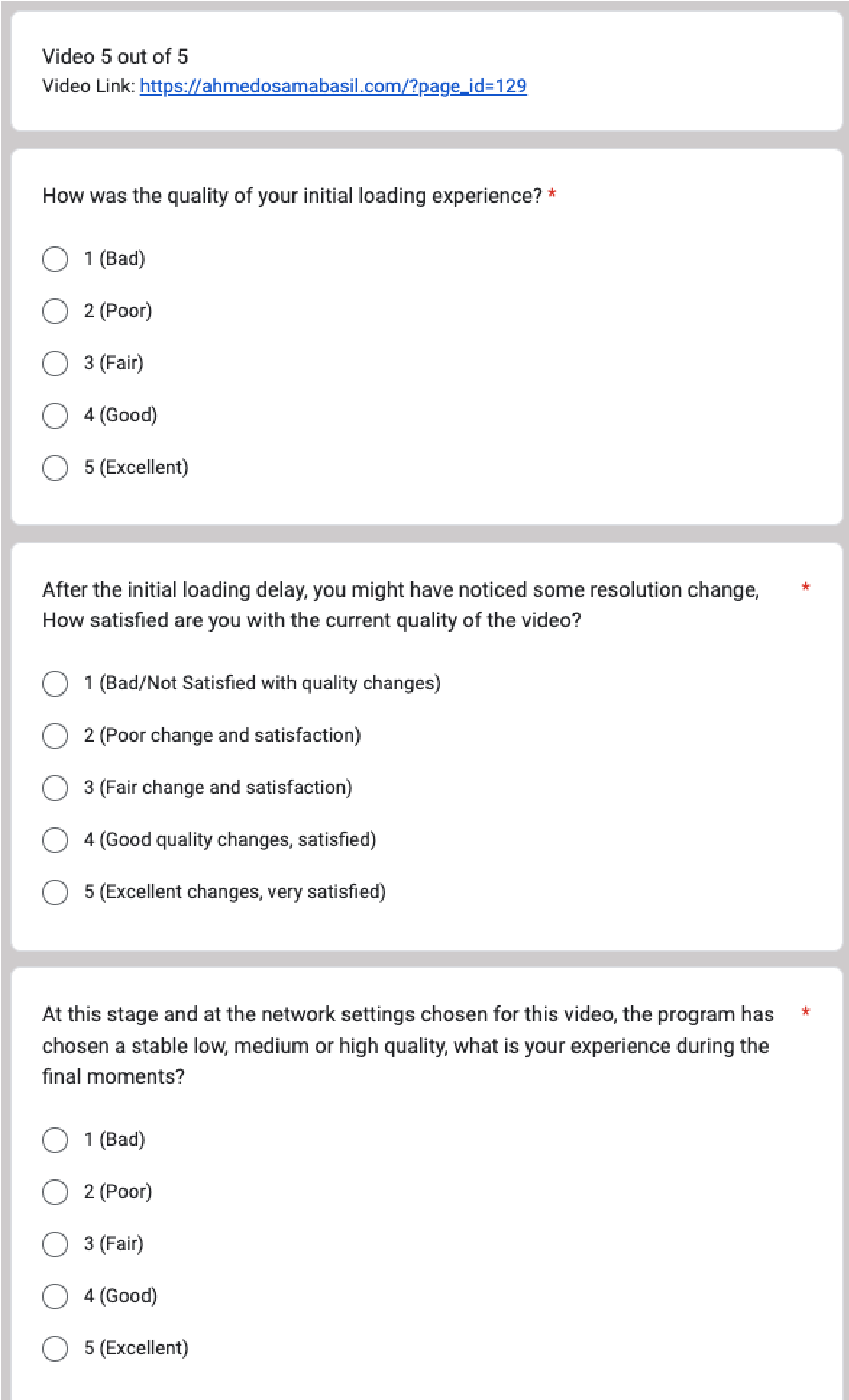
After the initial loading delay, you might have noticed some resolution change, How satisfied are you with the current quality of the video? *

- 1 (Bad/Not Satisfied with quality changes)
- 2 (Poor change and satisfaction)
- 3 (Fair change and satisfaction)
- 4 (Good quality changes, satisfied)
- 5 (Excellent changes, very satisfied)

At this stage and at the network settings chosen for this video, the program has chosen a stable low, medium or high quality, what is your experience during the final moments? *

- 1 (Bad)
- 2 (Poor)
- 3 (Fair)
- 4 (Good)
- 5 (Excellent)

Figure 7.9: Experimentation Form Ninth Page



Video 5 out of 5
Video Link: https://ahmedosamabasil.com/?page_id=129

How was the quality of your initial loading experience? *

- 1 (Bad)
- 2 (Poor)
- 3 (Fair)
- 4 (Good)
- 5 (Excellent)

After the initial loading delay, you might have noticed some resolution change, How satisfied are you with the current quality of the video? *

- 1 (Bad/Not Satisfied with quality changes)
- 2 (Poor change and satisfaction)
- 3 (Fair change and satisfaction)
- 4 (Good quality changes, satisfied)
- 5 (Excellent changes, very satisfied)

At this stage and at the network settings chosen for this video, the program has chosen a stable low, medium or high quality, what is your experience during the final moments? *

- 1 (Bad)
- 2 (Poor)
- 3 (Fair)
- 4 (Good)
- 5 (Excellent)

Figure 7.10: Experimentation Form Tenth Page

I Agree to share my experience with the researcher to improve his/her research result's accuracy and to aid in the improvement of the future development of QoE Prediction Algorithms and Dynamic Adaptive Streaming over HTTP Services. (Data will be anonymised and your name and personal data will not be included in the publicly published material). This experiment is being conducted on Google Forms presenting videos that were created in a virtual environment by the researcher. The purpose, procedure, and risks of participating in this experiment have been explained above. I voluntarily agree to participate in this experiment. I understand that I may ask questions, and that I have the right to withdraw from the experiment at any time. I also understand that University of Northampton Researchers may exclude me from the experiment at any time. I understand that any data I contribute to this experiment will not be identified with me personally, but will only be reported as a statistical average. *

- [ ] I Agree

Submit

Clear form

Figure 7.11: Experimentation Form Eleventh Page

Bibliography

[1] F. A. Cruz-Perez and L. Ortigoza-Guerrero, "Equal resource sharing allocation with qos differentiation for conversational services in wireless communication networks," *IEE Proceedings - Communications*, vol. 150, pp. 391–, Oct 2003.

[2] M. I. Kazantzidis, L. Wang, and M. Gerla, "On fairness and efficiency of adaptive audio application layers for multihop wireless networks," in *1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99) (Cat. No.99EX384)*, pp. 357–362, Nov 1999.

[3] K. Singh and P. Jayashree, "Improving wlan qos for multimedia application using mac and application layer (map) approach," in *2014 International Conference on Recent Trends in Information Technology*, pp. 1–7, April 2014.

[4] H. Du, J. Ge, C. Zhang, and P. Tian, "Fairness-aware sub-optimal relay selection for physical-layer security in af relaying networks," in *2016 8th International Conference on Wireless Communications Signal Processing (WCSP)*, pp. 1–5, Oct 2016.

[5] H. Wu and L. Guo, "An improved security tcp handshake protocol with authentication," in *2011 International Conference on Internet Technology and Applications*, pp. 1–4, Aug 2011.

[6] A. V. Babu and L. Jacob, "Fairness analysis of ieee 802.11 multirate wireless lans," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 3073–3088, Sep. 2007.

[7] *Network World State of the Network Survey*, 2017.

[8] B. Butler, *What SDN is and where it's going. [online] Network World*, 2019.

[9] D. C. Marinescu, "Chapter 2 - cloud service providers and the cloud ecosystem," in *Cloud Computing (Second Edition)* (D. C. Marinescu, ed.), pp. 13 – 49, Morgan Kaufmann, second edition ed., 2018.

[10] Y. Wang, *Survey of objective video quality measurements.* EMC.

[11] U. Engelke and H.-J. Zepernick, *Perceptual-based quality metrics for image and video services: A survey.* Proc. 3rd EuroNGI Conf. Netw., Trondheim, Norway, 2007. pp. 190–197.

[12] W. Lin and C. C. J. Kuo, *Perceptual visual quality metrics: A survey.* J. Vis. Commun. Image Represent, vol. 22, no. 4, pp. 297–312 ed., May 2011.

[13] M. R. S. Chikkerur, V. Sundaram and L. Karam, *Objective video quality assessment methods: A classification, review, performance comparison.* IEEE Trans. Broadcast, ieee trans. broadcast., vol. 57, no. 2, pp. 165–182 ed., June 2011.

[14] A. O. Basil, M. Mu, and M. Opoku Agyeman, "A Multi-Modal framework for future emergency systems," in *IEEE Smart World Congress 2019 (SWC 2019)*, (Leicester, United Kingdom (Great Britain)), Aug. 2019.

[15] A. O. Basil, M. Mu, and A. Al-Sherbaz, "A software defined network based research on fairness in multimedia," in *Proceedings of the 1st International Workshop on Fairness, Accountability, and Transparency in MultiMedia*, FAT/MM '19, (New York, NY, USA), pp. 11–18, ACM, 2019.

[16] C.-H. Ke, "P4: Switch: Mac layer and ip layer forwarding."

[17] M. Alsaeedi, M. M. Mohamad, and A. A. Al-Roubaiey, "Toward adaptive and scalable openflow-sdn flow control: A survey," *IEEE Access*, vol. 7, pp. 107346–107379, 2019.

[18] A. O. Basil, "Experimenation datasets and algorithms." https://github.com/AhmedOsamaBasil/PhD, 2022.

[19] A. O. Basil, M. Mu, and A. Al-Sherbaz, "Novel quality of experience experimentation framework through programmable network management," in *2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)*, pp. 485–486, 2022.

[20] A. O. B. Al-Mashhadani, M. Mu, and A. Al-Sharbaz, "Quality of experience experimentation prediction framework through programmable network management," *Network*, vol. 2, no. 4, pp. 500–518, 2022.

[21] H. SHI, R. V. Prasad, E. Onur, and I. G. M. M. Niemegeers, "Fairness in wireless networks:issues, measures and challenges," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 5–24, First 2014.

[22] K. Zeng, H. Yeganeh, and Z. Wang, "Quality-of-experience of streaming video: Interactions between presentation quality and playback stalling," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 2405–2409, Sep. 2016.

[23] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming," *IEEE Communications Surveys Tutorials*, vol. 17, pp. 469–492, Firstquarter 2015.

[24] J. P. D. Ghadiyaram and A. C. Bovik, *A Time-varying Subjective Quality Model for Mobile Streaming Videos with Stalling Events.* San Diego, CA, USA, proceedings of spie applications of digital image processing xxxviii, ed., 2015.

[25] M. Seufert, N. Wehner, P. Casas, and F. Wamser, "A fair share for all: Novel adaptation logic for qoe fairness of http adaptive video streaming," in *2018 14th International Conference on Network and Service Management (CNSM)*, pp. 19–27, Nov 2018.

[26] C. Dictionary, "Definition of fairness," 2019.

[27] O. Dictionaries, "fairness | definition of fairness in english by oxford dictionaries," 2019.

[28] Dictionary, "Definition of fairness," 2019.

[29] H. B. Can Emre Koksal, Hisham Kassab, ed., *An analysis of short-term fairness in wireless media access protocols*, (New York, NY, USA), Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, June 18 - 21, 2000.

[30] R. Amin, M. Reisslein, and N. Shah, "Hybrid sdn networks: A survey of existing approaches," *IEEE Communications Surveys Tutorials*, vol. 20, pp. 3259–3306, Fourthquarter 2018.

[31] Y. Jiang and J. Qin, "Achieving fairness and efficiency for association control in the heterogeneous wireless networks," in *2011 IEEE 3rd International Conference on Communication Software and Networks*, pp. 264–268, May 2011.

[32] Q. Zhu and J. Oh, "Deep reinforcement learning for fairness in distributed robotic multi-type resource allocation," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 460–466, Dec 2018.

[33] Y. Li, D. Li, W. Cui, and R. Zhang, "Research based on osi model," in *2011 IEEE 3rd International Conference on Communication Software and Networks*, pp. 554–557, May 2011.

[34] S. Orzen, "Interaction understanding in the osi model functionality of networks with case studies," in *2014 IEEE 9th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI)*, pp. 327–330, May 2014.

[35] Hagen, *TCP Fairness.* Jauu.net, Aug. 2011.

[36] D. Lee, B. E. Carpenter, and N. Brownlee, "Observations of udp to tcp ratio and port numbers," in *2010 Fifth International Conference on Internet Monitoring and Protection*, pp. 99–104, May 2010.

[37] B. Hu, Y. Yang, Z. Wang, and L. Xing, "Ccmf: A congestion control method with fairness for multipath transport," in *2017 International Conference on Information Networking (ICOIN)*, pp. 689–694, Jan 2017.

[38] S. Floyd, *High Speed TCP for Large Congestion Windows.* The Internet Society, Mar. 2004.

[39] Z. Chi-Min, "Fairness improvement of high speed tcp congestion control algorithm," in *2013 International Conference on Computational and Information Sciences*, pp. 1130–1133, June 2013.

[40] D. X. Wei, "Known problems in tcp algorithms in linux 2.6.16.3," June 2006.

[41] K. Lepikhov, "Highspeed tcp." Online, Apr. 2018.

[42] C. M. Margaret Rouse, George Lawton, "Udp (user datagram protocol)." Online, June 2018.

[43] Zhaojuan Yue, Yongmao Ren, and Jun Li, "Performance evaluation of udp-based high-speed transport protocols," in *2011 IEEE 2nd International Conference on Software Engineering and Service Science*, pp. 69–73, July 2011.

[44] X. Wu and A. A. Chien, "Evaluation of rate-based transport protocols for lambda-grids," in *Proceedings. 13th IEEE International Symposium on High performance Distributed Computing, 2004.*, pp. 87–96, June 2004.

[45] M. Kazantzidis and M. G. and, "Permissible throughput network feedback for adaptive multimedia in aodv manets," in *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*, vol. 5, pp. 1352–1356 vol.5, June 2001.

[46] T. Ozcelebi, M. O. Sunay, M. R. Civanlar, and A. M. Tekalp, "Application-layer qos fairness in wireless video scheduling," in *2006 International Conference on Image Processing*, pp. 1673–1676, Oct 2006.

[47] J. Otwani, A. Agarwal, and A. K. Jagannatham, "Optimal scalable video scheduling policies for real-time single- and multiuser wireless video networks," *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 2424–2435, June 2015.

[48] J. L. Sobrinho and A. S. Krishnakumar, "Equb-ethernet quality of service using black bursts," in *Proceedings 23rd Annual Conference on Local Computer Networks. LCN'98 (Cat. No.98TB100260)*, pp. 286–296, Oct 1998.

[49] R. A. Djeujo and C. Ruland, "Qoss definition and session description protocol for secure communication using compressive sensing," in *2017 Sixth International Conference on Future Generation Communication Technologies (FGCT)*, pp. 1–7, Aug 2017.

[50] J. Park, J. Kim, M. Jang, and B. Lee, "Time-based interest protocol for real-time content streaming in content-centric networking (ccn)," in *2013 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 512–513, Jan 2013.

[51] C. Liu and C. Zhou, "Providing quality of service in ieee 802.11 wlan," in *20th International Conference on Advanced Information Networking and Applications - Volume 1 (AINA'06)*, vol. 1, pp. 817–824, April 2006.

[52] A. Ksentini, A. Gueroui, and M. Naimi, "A new ieee 802.11 mac protocol with admission control for sensitive multimedia applications," in *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, vol. 5, pp. 5 pp.–3006, Nov 2005.

[53] G. Ye, C. Wu, J. Yue, S. Cheng, and W. Cheng, "Research on design of structured application layer multicast prototype about self-similar structured multicast (s3m)," in *2008 International Seminar on Business and Information Management*, vol. 1, pp. 384–387, Dec 2008.

[54] X. Zhang, X. Li, W. Luo, and B. Yan, "On improving the multicast performance of application layer multicast," in *2009 First International Workshop on Education Technology and Computer Science*, vol. 1, pp. 903–907, March 2009.

[55] X. Rao, J. Xu, Q. Xu, and Y. Zhou, “Study of the efficiency optimization of application layer multicast,” in *2009 IEEE Symposium on Industrial Electronics Applications*, vol. 1, pp. 170–173, Oct 2009.

[56] A. Mourad and M. Ahmed, “A scalable approach for application layer multicast in p2p networks,” in *2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 498–503, March 2008.

[57] J. Makishi, D. Chakraborty, T. Osada, G. Kitagata, A. Takeda, K. Hashimoto, and N. Shiratori, “A fair load sharing scheme for multi-source multimedia overlay application layer multicast,” in *2010 International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 337–344, Feb 2010.

[58] B. P. R. S. N. . W. F. McAuliffe, K., “The developmental foundations of human fairness,” *Nat. Hum. Behav. 1, 0042*, 2017.

[59] Aristotle, “The nicomachean ethics,” *Harvard Univ. Press*, 1999.

[60] J. Rawls, “A theory of justice,” *Harvard Univ. Press*, 1971.

[61] J. S. Adams, “Inequity in social exchange,” *Adv. Exp. Soc. Psychol. 2, 267–299*, 1965.

[62] M. Deutsch, “Equity, equality, and need: what determines which value will be used as the basis of distributive justice?,” *J. Soc. Issues 31, 137–149*, 1975.

[63] T. Hoßfeld, P. E. Heegaard, L. Skorin-Kapov, and M. Varela, “No silver bullet: Qoe metrics, qoe fairness, and user diversity in the context of qoe management,” in *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, May 2017.

[64] S. Cicalò, N. Changuel, V. Tralli, B. Sayadi, F. Faucheux, and S. Kerboeuf, “Improving qoe and fairness in http adaptive streaming over lte network,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, pp. 2284–2298, Dec 2016.

[65] T. Hoßfeld, L. Skorin-Kapov, P. E. Heegaard, and M. Varela, “Definition of qoe fairness in shared systems,” *IEEE Communications Letters*, vol. 21, pp. 184–187, Jan 2017.

[66] G. Chao, “Human-computer interaction: Process and principles of human-computer interface design,” in *2009 International Conference on Computer and Automation Engineering*, pp. 230–233, March 2009.

[67] N. Elmarzouqi, E. Garcia, and J.-C. Lapayre, “Cscw from coordination to collaboration,” in *Computer Supported Cooperative Work in Design IV* (W. Shen, J. Yong, Y. Yang, J.-P. A. Barthès, and J. Luo, eds.), (Berlin, Heidelberg), pp. 87–98, Springer Berlin Heidelberg, 2008.

[68] S. Kiesler and P. Hinds, “Introduction to this special issue on human-robot interaction,” *Hum.-Comput. Interact.*, vol. 19, pp. 1–8, June 2004.

[69] S. Thrun, "Toward a framework for human-robot interaction," *Hum.-Comput. Interact.*, vol. 19, pp. 9–24, June 2004.

[70] M. El Mesbahi, N. Elmarzouqi, and J.-C. Lapayre, "Fairness properties for collaborative work using human-computer interactions and human-robot interactions based environment: "let us be fair"," in *Distributed, Ambient, and Pervasive Interactions* (N. Streitz and P. Markopoulos, eds.), (Cham), pp. 319–328, Springer International Publishing, 2014.

[71] M. Mirzaee and S. Akhlaghi, "Achievable secrecy rate of wiretap channels incorporating multi-input–multi-output amplify and forward relaying," *IET Communications*, vol. 10, no. 3, pp. 300–308, 2016.

[72] E. Aral, Y. Hirano, T. Murase, and M. Oguchi, "Empirical study on a session layer resource control over 802.11 dcf wireless lan," in *2008 7th Asia-Pacific Symposium on Information and Telecommunication Technologies*, pp. 29–34, April 2008.

[73] H. Shimonishi, T. Hama, and T. Murase, "Tcp congestion control enhancements for streaming media," in *2007 4th IEEE Consumer Communications and Networking Conference*, pp. 303–307, Jan 2007.

[74] A. Salem and J. W. Atwood, "Formal validation of the security properties of amt's three-way handshake," in *2011 IEEE 36th Conference on Local Computer Networks*, pp. 227–230, Oct 2011.

[75] W. K. Lai and C. Y. Pan, "Achieving inter-session fairness for layered video multicast," in *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)*, vol. 3, pp. 935–939 vol.3, June 2001.

[76] L. Xiaobin, X. Yinlong, Z. Yulin, and O. Kaiqian, "Lexicographically fair bandwidth allocation in multi-session layered media multicast using network coding," in *2009 2nd IEEE International Conference on Broadband Network Multimedia Technology*, pp. 168–173, Oct 2009.

[77] S. Ramesh and S. K. Kasera, "Best effort session-level congestion control," in *2007 IEEE International Conference on Network Protocols*, pp. 236–245, Oct 2007.

[78] M. Di Penta and D. A. Tamburri, "Combining quantitative and qualitative studies in empirical software engineering research," in *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pp. 499–500, May 2017.

[79] L. R. Student, R. Goubran, and F. Kwamena, "Computer vision-assisted human-in-the-loop measurements: augmenting qualitative by increasing quantitative analytics for ci situational awareness," in *2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pp. 1–6, June 2018.

[80] S. F. Bessane, M. S. Camara, I. Fall, and A. Bah, "Causal model of performance measurement systems by combining qualitative and quantitative models for robust results," in *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pp. 1–7, April 2018.

[81] R. Jain, D. M. Chiu, and H. WR, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," *CoRR*, vol. cs.NI/9809099, 01 1998.

[82] W. Foundation, "Fairness measure," May 2019.

[83] R. Jain, "Fairness: How to measure quantitatively," *ResearchGate*, 05 2019.

[84] Y. Li, M. Sheng, C. W. Tan, Y. Zhang, Y. Sun, X. Wang, Y. Shi, and J. Li, "Energy-efficient subcarrier assignment and power allocation in ofdma systems with max-min fairness guarantees," *IEEE Transactions on Communications*, vol. 63, pp. 3183–3195, Sep. 2015.

[85] J. Choi, "Power allocation for max-sum rate and max-min rate proportional fairness in noma," *IEEE Communications Letters*, vol. 20, pp. 2055–2058, Oct 2016.

[86] R. Sun, M. Hong, and Z. Luo, "Joint downlink base station association and power control for max-min fairness: Computation and complexity," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 1040–1054, June 2015.

[87] N. Kongwatmai and K. Rojviboonchai, "Qualitative study of tcp newrapid," in *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pp. 1–4, May 2012.

[88] M. Han and J. P. Hong, "Balancing qoe and fairness of layered video multicast in lte networks," in *2012 14th International Conference on Advanced Communication Technology (ICACT)*, pp. 387–392, Feb 2012.

[89] A. Sideris, E. Markakis, A. Trigonis, G. Alexiou, E. Pallis, and C. Skianis, "Mpeg-dash over idvb-t: The qoe unfairness issue," in *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 70–74, Dec 2014.

[90] M. Seufert, N. Wehner, and P. Casas, "A fair share for all: Tcp-inspired adaptation logic for qoe fairness among heterogeneous http adaptive video streaming clients," *IEEE Transactions on Network and Service Management*, vol. 16, pp. 475–488, June 2019.

[91] E. Y. B. M. M. M. R. N. Georgopoulos, Panagiotis, ed., *Towards network-wide QoE fairness using openflow-assisted adaptive video streaming*, vol. Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking, ACM SIGCOMM, 2013.

[92] L. S. F. J. D. T. F. Petrangeli Stefano, Claeys Maxim, "A multi-agent q-learning-based framework for achieving fairness in http adaptive streaming," *IEEE Network Operations and Management Symposium (NOMS)*, 2014.

[93] H. Yousef, J. L. Feuvre, and A. Storelli, "Abr prediction using supervised learning algorithms," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, 2020.

[94] Y. Sani, D. Raca, J. J. Quinlan, and C. J. Sreenan, "Smash: A supervised machine learning approach to adaptive video streaming over http," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, 2020.

[95] H. Mao, R. Netravali, and M. Alizadeh, “Neural adaptive video streaming with pensieve,” in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM ’17, (New York, NY, USA), p. 197–210, Association for Computing Machinery, 2017.

[96] Z. Meng, J. Chen, Y. Guo, C. Sun, H. Hu, and M. Xu, “Pitree: Practical implementation of abr algorithms using decision trees,” in *Proceedings of the 27th ACM International Conference on Multimedia*, MM ’19, (New York, NY, USA), p. 2431–2439, Association for Computing Machinery, 2019.

[97] Y. Zhu, L. Song, R. Xie, and W. Zhang, “Sjtu 4k video subjective quality dataset for content adaptive bit rate estimation without encoding,” in *2016 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–4, 2016.

[98] A. Ben Letaifa, “Adaptive qoe monitoring architecture in sdn networks: Video streaming services case,” in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1383–1388, 2017.

[99] L. Wang and D. T. Delaney, “Qoe oriented cognitive network based on machine learning and sdn,” in *2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN)*, pp. 678–681, 2019.

[100] T. Abar, A. Ben Letaifa, and S. El Asmi, “Machine learning based qoe prediction in sdn networks,” in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1395–1400, 2017.

[101] T. Abar, A. B. Letaifa, and S. E. Asmi, “Real time anomaly detection-based qoe feature selection and ensemble learning for http video services.,” in *2019 7th International conference on ICT Accessibility (ICTA)*, pp. 1–6, 2019.

[102] A. E. Al-Issa, A. Bentaleb, A. A. Barakabitze, T. Zinner, and B. Ghita, “Bandwidth prediction schemes for defining bitrate levels in sdn-enabled adaptive streaming,” in *2019 15th International Conference on Network and Service Management (CNSM)*, pp. 1–7, 2019.

[103] A. K. Moorthy, L. K. Choi, A. C. Bovik, and G. de Veciana, “Video quality assessment on mobile devices: Subjective, behavioral and objective studies,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 652–671, 2012.

[104] C. Chen, L. K. Choi, G. de Veciana, C. Caramanis, R. W. Heath, and A. C. Bovik, “Modeling the time—varying subjective quality of http video streams with rate adaptations,” *IEEE Transactions on Image Processing*, vol. 23, no. 5, pp. 2206–2221, 2014.

[105] D. Ghadiyaram, A. C. Bovik, H. Yeganeh, R. Kordasiewicz, and M. Gallant, “Study of the effects of stalling events on the quality of experience of mobile streaming videos,” in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 989–993, 2014.

[106] Z. Duanmu, K. Zeng, K. Ma, A. Rehman, and Z. Wang, “A quality-of-experience index for streaming video,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 1, pp. 154–166, 2017.

[107] C. G. Bampis, Z. Li, A. K. Moorthy, I. Katsavounidis, A. Aaron, and A. C. Bovik, "Study of temporal effects on subjective video quality of experience," *IEEE Transactions on Image Processing*, vol. 26, no. 11, pp. 5217–5231, 2017.

[108] Z. Duanmu, A. Rehman, and Z. Wang, "A quality-of-experience database for adaptive video streaming," *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 474–487, 2018.

[109] I. DASH.

[110] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http," in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, MMSys '11, (New York, NY, USA), p. 157–168, Association for Computing Machinery, 2011.

[111] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, timid, and unstable: Picking a video streaming rate is hard," in *Proceedings of the 2012 Internet Measurement Conference*, IMC '12, (New York, NY, USA), p. 225–238, Association for Computing Machinery, 2012.

[112] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '12, (New York, NY, USA), p. 97–108, Association for Computing Machinery, 2012.

[113] D. Yun and K. Chung, "Rate adaptation for http video streaming to improve the qoe in multi-client environments," *KSII Transactions on Internet and Information Systems*, vol. 9, pp. 4519–4533, 11 2015.

[114] A. Bentaleb, A. C. Begen, and R. Zimmermann, "Sdndash: Improving qoe of http adaptive streaming using software defined networking," in *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, (New York, NY, USA), p. 1296–1305, Association for Computing Machinery, 2016.

[115] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous, "Sdnhas: An sdn-enabled architecture to optimize qoe in http adaptive streaming," *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2136–2151, 2017.

[116] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," *ACM SIGCOMM Computer Communication Review*, vol. 45, pp. 325–338, 08 2015.

[117] S. Xiang, L. Cai, and J. Pan, "Adaptive scalable video streaming in wireless networks," in *Proceedings of the 3rd Multimedia Systems Conference*, MMSys '12, (New York, NY, USA), p. 167–172, Association for Computing Machinery, 2012.

[118] R. Kuschnig, I. Kofler, and H. Hellwagner, "An evaluation of tcp-based rate-control algorithms for adaptive internet streaming of h.264/svc," in *Proceedings of the First Annual ACM SIGMM*

Conference on Multimedia Systems, MMSys '10, (New York, NY, USA), p. 157–168, Association for Computing Machinery, 2010.

[119] V. Krishnamoorthi, N. Carlsson, D. Eager, A. Mahanti, and N. Shahmehri, "Helping hand or hidden hurdle: Proxy-assisted http-based adaptive streaming performance," pp. 182–191, 08 2013.

[120] S. Schwarzmann, T. Zinner, and O. Dobrijevic, "Quantitative comparison of application–network interaction: a case study of adaptive video streaming," *Quality and User Experience*, vol. 2, no. 1, p. 7, 2017.

[121] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide qoe fairness using openflow-assisted adaptive video streaming," in *Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human-Centric Multimedia Networking*, FhMN '13, (New York, NY, USA), p. 15–20, Association for Computing Machinery, 2013.

[122] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, "Delivering stable high-quality video: An sdn architecture with dash assisting network elements," in *Proceedings of the 7th International Conference on Multimedia Systems*, MMSys '16, (New York, NY, USA), Association for Computing Machinery, 2016.

[123] H. Egilmez, S. Civanlar, and A. Tekalp, "An optimization framework for qos-enabled adaptive video streaming over openflow networks," *IEEE Transactions on Multimedia*, vol. vol. 15, pp. pp. 710–715, 04 2013.

[124] H. Egilmez and A. Tekalp, "Distributed qos architectures for multimedia streaming over software defined networks," *Multimedia, IEEE Transactions on*, vol. 16, pp. 1597–1609, 10 2014.

[125] H. E. Egilmez, S. T. Dane, K. T. Bagci, and A. M. Tekalp, "Openqos: An openflow controller design for multimedia delivery with end-to-end quality of service over software-defined networks," in *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 1–8, 2012.

[126] N. Bouten, J. Famaey, S. Latré, R. Huysegems, B. D. Vleeschauwer, W. V. Leekwijck, and F. D. Turck, "Qoe optimization through in-network quality adaptation for http adaptive streaming," in *2012 8th international conference on network and service management (cnsm) and 2012 workshop on systems virtualiztion management (svm)*, pp. 336–342, 2012.

[127] H. K. Yarnagula, P. Juluri, S. K. Mehr, V. Tamarapalli, and D. Medhi, "Qoe for mobile clients with segment-aware rate adaptation algorithm (sara) for dash video streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, jun 2019.

[128] A. El Essaili, D. Schroeder, E. Steinbach, D. Staehle, and M. Shehada, "Qoe-based traffic and resource management for adaptive http video delivery in lte," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 25, p. 988–1001, jun 2015.

[129] I. INT, "P.913 : Methods for the subjective assessment of video quality, audio quality and audiovisual quality of internet video and distribution quality television in any environment," June 2021.

[130] M. Chiesa, G. Lospoto, M. Rimondini, and G. Di Battista, "Intra-domain routing with pathlets," *Computer Communications*, vol. 46, pp. 76–86, 2014.

[131] H. Zhang, C. Lumezanu, J. Rhee, N. Arora, Q. Xu, and G. Jiang, "Enabling layer 2 pathlet tracing through context encoding in software-defined networking," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, HotSDN '14, (New York, NY, USA), p. 169–174, Association for Computing Machinery, 2014.

[132] J. Naous, M. Walfish, A. Nicolosi, D. Mazières, M. Miller, and A. Seehra, "Verifying and enforcing network paths with icing," 2011. Cited by: 96; All Open Access, Green Open Access.

[133] K. Agarwal, E. Rozner, C. Dixon, and J. Carter, "Sdn traceroute: Tracing sdn forwarding without changing network behavior," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, HotSDN '14, (New York, NY, USA), p. 145–150, Association for Computing Machinery, 2014.

[134] R. Braga, E. Mota, and A. Passito, "Lightweight ddos flooding attack detection using nox/openflow," pp. 408–415, 10 2010.

[135] Y. Feng, R. Guo, D. Wang, and B. Zhang, "Research on the active ddos filtering algorithm based on ip flow," vol. 02, pp. 628–632, 01 2009.

[136] M. M. Kenji KURATA, Go HASEGAWA, ed., *Fairness Comparisons Between TCP Reno and TCP Vegas for Future Deployment of TCP Vegas*, 2016.

[137] B. Pfaff, J. Pettit, T. Koponen, E. J. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, P. Shelar, K. Amidon, and M. Casado, "The design and implementation of open vswitch," in *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation*, NSDI'15, (USA), p. 117–130, USENIX Association, 2015.

[138] Q. Huynh-Thu and M. Ghanbari, *Scope of validity of PSNR in image/video quality assessment.* Electronics letters, vol. 44, no. 13, June 19, 2008.

[139] M. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," in *IEEE Transactions on Broadcasting, vol. 50, no. 3*, September 2004.

[140] Z. Wang and Q. Li, "Video quality assessment using a statistical model of human visual speed perception," in *Journal of the Optical Society of America A, vol. 24, no. 12, pp. B61-B69*, December 2007.

[141] M. Latora, Vito; Marchiori, *Efficient Behavior of Small-World Networks.* Phys. Rev. Lett. 87 (19): 198701, 17 October 2001.

[142] S. Choi, X. Long, M. Shahbaz, S. Booth, A. Keep, J. Marshall, and C. Kim, "The case for a flexible low-level backend for software data planes," in *Proceedings of the First Asia-Pacific Workshop on Networking*, APNet'17, (New York, NY, USA), p. 71–77, Association for Computing Machinery, 2017.

[143] p4lang, "Acm sigcomm august 2019 tutorial on programming the network data plane," Aug. 2019.

[144] D. Bhat, A. Rizk, M. Zink, and R. Steinmetz, "Network assisted content distribution for adaptive bitrate video streaming," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, MMSys'17, (New York, NY, USA), p. 62–75, Association for Computing Machinery, 2017.

[145] R. Bhattacharyya, A. Bura, D. Rengarajan, M. Rumuly, B. Xia, S. Shakkottai, D. Kalathil, R. K. P. Mok, and A. Dhamdhere, "Qflow: A learning approach to high qoe video streaming at the wireless edge," *IEEE/ACM Trans. Netw.*, vol. 30, p. 32–46, feb 2022.