UNIVERSITY OF ZULULAND

ENABLING NETWORK STABILITY IN A SCALABLE SOFTWARE-DEFINED DATA CENTRE NETWORK

By

Ayotuyi Tosin, **AKINOLA**

(201426777)

A thesis submitted in fulfilment of the requirements for the degree

of

Doctor of Philosophy in Computer Science

Faculty of Science, Agriculture and Engineering

Department of Computer Science

KwaDlangezwa, RSA.

Supervisor: Prof. M.O. Adigun

Co-Supervisor: Dr. P. Mudali

2022

DECLARATION

This thesis represents the author's own research work and has not been submitted in any form to any other tertiary education for another degree or diploma. All resources have been duly acknowledged and referenced in the text.

Signature

DEDICATION

This thesis is dedicated to the Almighty God, the giver of life and all privileges. It is also dedicated to my mother, Mrs. Olushola Comfort Akinola, for her prayers and for being there all the time.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisors, Prof. M.O Adigun and Dr Pragasen Mudali for their guidance, support, and patience throughout this research. I would also like to acknowledge the contributions of the Holy Spirit who inspires and gives direction to the source of all knowledge. I say, ‘thank you’, to A.O Adebayo, Sizoluethu Ndlovu, N. I Ezeji, N. C Sibeko, P. Tarwireyi, O.A Oki, my fellow researchers, in the Centre for mobile e-Services, for the support you gave me through the development of the ideas that resulted in this thesis. I cannot forget Thabile Nthuli for all her assistance at the commencement of this degree and to Karen Enslin, ‘the prompt angel’ who took over from her. The character she portrayed especially during the difficult phases of the research was a great encouragement.

My profound appreciation also goes to the Deeper Life Bible Church KwaZulu Natal at large for their spiritual support. Pastor Dr. and Mrs. Olusanya, Pastor and Mrs. Ilesanmi, Pastor Ogunniyi, Brother Victor Oluwatosinloba, Dr Oladejo and Wife, Sister Idowu Taiwo, Sister Akintayo Elizabeth, Sister Olalekan Marvellous Damilola, and her younger sister, Favour. You are all precious and will never be forgotten for your immense faith in me that this can be done. I cannot forget my ‘generous mummy’ - my supervisor’s wife - Mummy Adigun, for her prayers, comforts, encouragement, faith and kindness to me as I got to know her briefly before this research ended. May God gladden her heart and fill it with unending joy and peace in Jesus’ name.

I also want to acknowledge the support from my sponsors, The South African Telkom Company and the Centre of Excellence in the University of Zululand for their financial assistance beyond the normal duration for me to complete this research. I am grateful.

Lastly, I would like to appreciate ‘the tree supporting my barn’ in the person of my biological mother, Mrs. Comfort Olusola Akinola, for her motherly prayers. I am immensely indebted to you.

ABSTRACT

In the recent past, Data Centre Network (DCN) has received a lot of research attention because of the indispensable role that DCNs play in modern-day network communication as; the data storage container for on-demand service provisioning. One of the major responsibilities of the DCN is the management of the network's need for optimization, thereby addressing request services. One consequence of the increasing requests is network flow interference that incurs network instability. Therefore, the task of addressing flow interference within a network becomes very important because it plays a vital role in restoring network stability.

Network scalability continues to be the concern of many networks management engineers while less consideration is given to the effectiveness and efficiency of their adopted network stability solutions. Furthermore, very few approaches to network stability gave significant attention to achieving an optimal solution. Several related works notably still left unaddressed slow convergence, computational complexity, complex memory requirements and lack of adaptive stability mechanisms among others. Therefore, the need to enhance network stability in a scalable software-defined and data-oriented DCN remains an open issue for the research community to contend with.

This research work addressed this knowledge gap by conducting a comprehensive literature review on relevant existing approaches used by other researchers to address network stability. This provided useful insight and motivation to launch a search for a more robust and appropriate approach to fast network adaptation, and economical utilization of network resources with no negative impairment on the expected optimal performance. The literature review revealed that the concept of Adaptive Rendering Technique (ART) in Computer Graphics was generic and analytically simple enough to adopt for high-level applicability to network flow.

The methodology for this study emerged from using ART as the analogy for modelling the traffic flow interference which causes network instability. A Multi-Objective Optimization Crosspoint Queue (MOCQ) was modelled after the primary stages of the ART's adaptive pipeline processes. The MOCQ process consisted of a control flow list, sorting entry matching, matching flow table and matching operation. This process enabled the achievement of stable controller-switch states.

The efficacy of the proposed ART-based MOCQ approach has been extensively validated using numerical analysis, miniature testbed and computer simulations.

The approach resulted in improved performance over existing approaches as demonstrated by faster stability, improved response time and network throughput, and speedy rate of convergence. Furthermore, a balanced perspective of individual interests on the sides of both service providers and end-users was attained. The emerging facts were confirmed by a measure of a 42.9% reduction in the rate of network response delays, and over 150% point improvement in correcting the existing rate of switch failures when compared to the ordinary Crosspoint Queue approach. The approach also assisted in determining the level of stability attained and in the case of the setup in this thesis, a level of 0.7 was achieved for the hierarchical data centre network we are considering. This determination is an improvement over existing systems as they do not go all the way to achieve this goal.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS, ACRONYMS, AND DEFINITIONS OF TERMS

ABU	Average Bandwidth Utilization
ABW	Available Bandwidth
ADMM	Alternating Direction Method of Multipliers
ART	Adaptive Rendering Technique
AST	Average Service Time
AUE	Average User Experience
AWT	Average Waiting Time
CDF	Cumulative Distribution Function
CoV	Coefficient of Variance
CQ	Crosspoint Queue
CQRD	Crosspoint Queue with Random-Drop
DART-MCP	Deterministic Adaptive Rendering Technique- Multi Constrained Path
DCN	Data Centre Network
DRR	Deficit Round-Robin
dRM	resistance constraints in Reactive Mode
ECN	Explicit Congestion Notification
ForCES	Forwarding and Control Element Separation
FST	Flow Setup Time
HCF	Hashed Credits Fair
IETF	Internet Engineering Task Force
KS	Kalai-Smorodinsky
LLDM	Link Layer Discovery Module
MOCQ	Multi-Objective Optimization Crosspoint Queue
MTBF	Mean Time Between Failure
MTTR	Mean Time to Repair
OMNeT++	Objective Modular Network Testbed in C++

OQ	Output Queue
PA	Pathload Algorithm
P-C	Proxy Controller
PCE	Path Computation Elements
PDQ	Preemptive Distributed Quick
PHMS	Public Health Management System
RCP	Routing Control Platform
SDN	Software Defined Networking
SF	Stability Function
SQA	Specified Quality Approach
TBP	Ticket Based Probing
TCP	Transmission Control Protocol
VL	Virtual Layer

LIST OF PUBLICATIONS

Akinola, A.T. and Adigun, M.O., **2017**. Performance Evaluation of Data Centre Network Setup Architectures. In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC), Barcelona, Spain* (pp. 166-171).

Akinola, A.T., Adigun, M.O. and Mudali, P.,**2018.** Effects of Scalability measures on Control Plane Architectural Designs in different Data Centre Architectures. In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC), Arabella, Hermanus, Western Cape, South Africa pp. 198-203, (2 - 5 September 2018)*

Akinola, A.T., Adigun, M.O. and Mudali, P.,**2019** Effect of SDN Controller-Proxy Deployment on Network Stability**"** In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)* Fairmont Zimbali Resort, Ballito RSA. (1-4 September 2019)

Akinola, A.T., Adigun, M.O. and Mudali, P., **2019,** September. Network Stability Based on the Amount of Available Bandwidth in a Software Defined Networking. In *2019 IEEE AFRICON* (pp. 1-6). IEEE.

Akinola, A.T., Adigun, M.O. and Mudali, P., **2020.** Performance Modeling of Software-Defined Networking Paradigm in a Public Health Management System. ***International Journal*** *of Information and Management Sciences*, *31*(2), pp.123-148.

Akinola A.T., Adigun M.O., Mudali P. **(2021)** Incorporating Stability Estimation into Quality of Service Routing in an SDN-Based Data Centre Network. In: Yang XS., Sherratt S., Dey N., Joshi A. (eds) Proceedings of Fifth International Congress on Information and Communication Technology. Advances in Intelligent Systems and Computing, ***vol 2 Issue 1184***. pp 406-416 ***Springer***, Singapore. https://doi.org/10.1007/978-981-15-5859-7_40

Akinola A.T., Adigun M.O., Mudali P. **2022**. A Multi-Criteria Routing Algorithm for SDN Based Deterministic Adaptive Rendering Technique, Indian **Journal** of Computer Science and Engineering (IJCSE) Vol 13 No 1 pp 170-187, Jan-Feb 2022.

Chapter 1

INTRODUCTION

1.1 Preamble

The art of computer networking represents a very cogent aspect of modern computing which results in a well-developed interconnected global world that surrounds us today (Aboubakar et al 2021). The importance of this connectivity via the network layouts cannot be over-emphasised due to various high-level applications that are often required for various purposes which varies from entertainment (games, music and video streaming) to other real-time interaction services (office tools), thus playing a cogent role in people's daily life (Akinola and Adigun, 2017, Kuteyi and Winkler, 2022). Therefore, the need for a fast and reliable means of communication is very important for the smooth running of these applications not only for enabling sharing of resources but also for optimal performance during their execution. In another perspective, the crux of a major aspect of modern computer networking has been the cogent need of people to communicate smoothly and ensure that vital information is being accessed from anywhere around the globe. These two key needs have fuelled the development of networks over the years, making the Internet ubiquitous in its nature (Lima et al 2022).

The ubiquitous nature of the Internet environment allows various devices from different regions in the world to have access to the resources on the Internet (D'Avila et al 2020, Filippetto et al 2021). Most of these resources are rendered on the Internet as services such as web search, social networking, online shopping, content streaming, instant messaging, video calling, digitised newspapers, books, or research articles, form the online landscape today (Cisco, 2018). However, due to the huge number of users and clients that access these services, enormous computing resources are needed for such platforms to run successfully. This demand has led to the development of specialised warehouse-scale computers that are housed in what is termed data centres. The data centres are embodiments of many hundreds of thousands of such computers, which usually power the services that are mentioned earlier as well as many others (Shafiq et al 2021).

Due to the need for the deployment of complex systems, system experts who are good at designing and operating the data centre becomes a necessity for the management of data centres. Only very few companies such as Microsoft, Oracle, IBM, Google, and Amazon were able to manage their own data centres effectively while a host of others were daily combating various levels of other challenges (Rygielski et al 2013, Hujainah et al 2021). Several other areas in the performance analysis of DCN have been researched and successes which include its availability (Wahyuni et al 2021), reliability (Gaur et al 2021), security (Zhu and Du, 2021), flexibility (Safron et al 2022), scalability (Akinola et al 2018, Gemawat, 2021) have been recorded. Thus, the existence of these and several other complexities have given birth to these lethal challenges due to increasing users to also carry out several transactions, rent business, store information among others on the cloud platform. The natural business environment usually depends on the ability to build an interactive platform that is reliable and conducive for end users, not only in smooth access to several rented resources as mentioned earlier but also with high predictable performance while using the interactive applications (Farhady et al 2015, Sridharan et al 2017, Mwesigwa and Oladipo, 2021).

Other innovative ideas or projects include the end-users' expectations as an encoded script into a contract often referred to as Service Level Agreements with specific service level objectives carrying strict violation punishments. The objectives that are usually stipulated under this contract may include metrics such as response time, throughput, latency and availability. Even though the contracts come with strict penalties, several end users still experience poor service delivery, as most of the promised performance is not strictly adhered to (Peeta and Yang 2003, Li et al, 2019). Adhering to the contract requires that proper and adequate mechanisms are put in place to continuously optimize the system for corresponding improved performance. Most poor service performance occurs as a result of traffic flow interference when two or more applications overlap when routing through a switch over a frame of a time interval.

At other times, they contend for some of the shared resources at the switches, such as the queue memory, link capacity and many other resources (Abdallah et al 2018, Huang et al 2019). This is because some flows often called giant flows (such as replication and backup flows) occupy the resources available for a while before completion. The persistence of network flow interference is one of the daunting tasks requiring the attention of network engineers. Moreover, when this task is not properly addressed, network stakeholders (end users and service providers) would notice intolerable delay in the network's performance of real-life events which is indicative of instability within the network. This thesis addresses issues relating to delayed network performance as a result of network instability.

1.2 Why is Network Stability important?

The tendency of a network to be able to maintain a bearable operational quality of service output in the presence of some external and internal influences is referred to as Network Stability (Rehman et al, 2021). This concept becomes very important in the network environment because several internal and external forces often interrupt the normal operation of networks; a situation that is intolerable in a real-time mode.

As the demand of network users for businesses and daily operations in life increases, many human activities rely on the availability of the Internet which is the operational tool for the smooth running of a data reservoir (Data Centre). This reservoir houses various applications and its performance, and therefore, requires continuous optimization processes (Chen et al 2015, Sridharan et al 2017, Wilson et al, 2021). The need for this improvement is not far-fetched because of the inherent short delay-sensitive flows that often occur in real-time interactive sessions on the Internet, thus reducing the flow performance in the data reservoir (Chen et al 2014, Sridharan et al 2017, Altaheini et al, 2022). The implication of this is that both long flows and short flows compete and contest for bandwidth and buffer resources that are available in the network in real-time. The competition, therefore, results in the breakdown of network quality of service thereby extending the duration of network flow completion time especially for the short, small flows (Alizadeh et al 2013, Zhang et al, 2022, Cheraghi et al, 2022).

Several recent studies have proven that the poor or deterred performance of the data centre as mentioned above is due to the unsuited network architecture as well as the coarse commodity data centre switch management schemes which result in set-up flow interferences (Zhang et al 2014, Atieh, 2021, Jia et al, 2022). The work of Zhu and Zhang further mentioned that interference occurs when two or more flows are suddenly left with the option of sharing the same flow path, queue memory, link capacity, and (or) network channel. Moreover, the diagram in Figure 1 describes the nature of flow characteristics in a real-time operation Data Centre Network (DCN) (Chen et al 2014, Liu et al, 20211. The workload on the graph showed that close to 90% of the traffic flow results from small flows while roughly the giant flows accrued around 4% of the total flow. The diagram shows that the network engineers unfolded the existence of significant flow-tussles as a result of the contentions between the minority giant flows and majority small flows (Chen et al 2014, Liu et al, 2021).

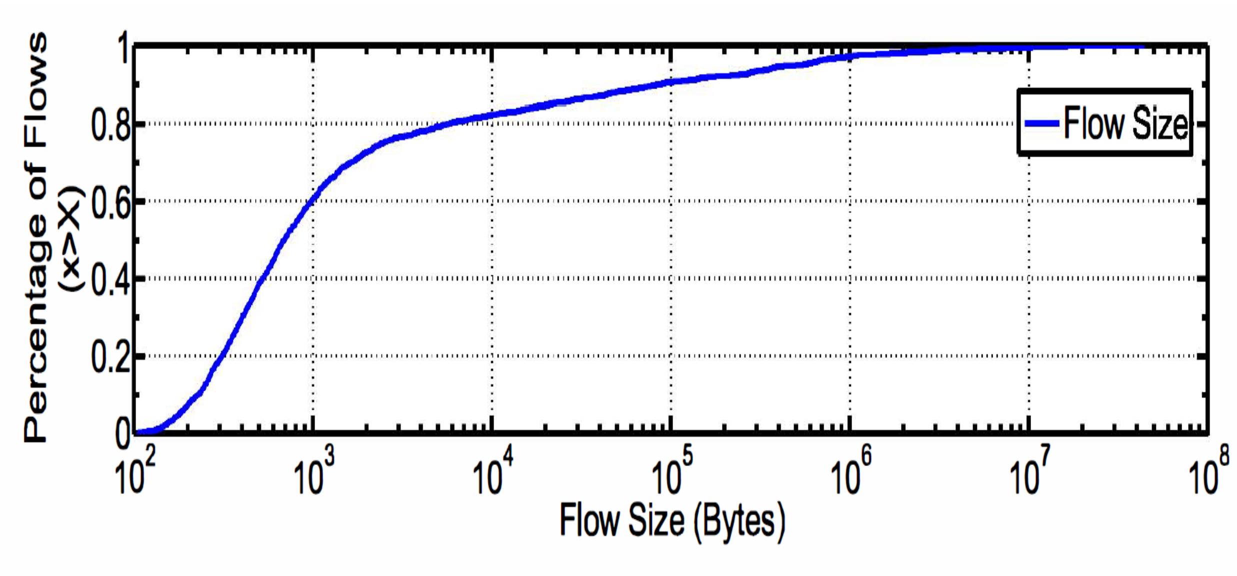


Figure 1.1: Traffic workload derived from real data centre operation (Chen et al 2014).

Considering a 48 by 48 stackable managed Gigabit Ethernet switch being used in the data centre shown in Figure 1, the switch has a remainder of 48 by 47 port paths in total with the assumption that there are no traffic flows through the input port. If there are approximately 6 giant flows switch paths currently running on the network, then, it implies that the remaining 2250 ports are meant to be used by hundreds of thousands of the small flows in the network. Hence, the switch has the same number of the output of 6 by 47 from the remaining 2250 ports whose output queues are similar and therefore also compete for the use of such paths within the switch ports.

In addition to the tussle for resources usage among the flows as identified above, there are several effects that these generate on a dynamic network environment. These effects include flow supply fluctuations (incidents), flow instability, dynamic variability in the differentiation of flow types in the traffic stream, the randomness in time-dependent sink destination request, information provision delay, and slow traffic response to the supplied information (Zhang et al 2014, Park and Bischof, 2022). The existence of all these effects around a network environment necessitates the provision of a stable traffic system which is pivotal to an efficient real-time operation expected to deliver an acceptable flow throughput within a limited response time (Liu et al 2021).

Hence, the observation from Figure 1 calls for an adaptive flow management approach that is more fine-grained to provide the solution to DCN flow interference. The decision to address the setup flow interference within the DCN is therefore vital and would primarily result in devising a strategic solution to network instability in the data centre. This is because the problem of network instability has not received enough attention and the perspective regarding an adaptive solution with multiple approaches is highly imperative. Hence, the utilization of a single approach is insufficient in addressing network instability which until now has been the norm in DCN.

Consequently, this research adopted the innovation of deploying an improved network management approach towards addressing the problem. Similarly, the aspect of a balanced mutual network benefit between service providers and end-users has not been adequately considered concerning network management systems to the best of our knowledge. This is also because most solutions either benefit service providers at the expense of end-users or vice versa. Failure in addressing this challenge would deny the participants the necessary openness and fairness, hence the need to balance the stakeholders' interest cannot be over-emphasised as it relates to network stability in DCN.

Therefore, this thesis proffered a solution that conceptualises the balance of stakeholders' interests in the context of traffic flow interference. Typically, this conceptualisation should involve explicit analysis and exhaustive discussions based on experimental results to solve the stakeholders' interest and satisfaction concerning network stability. Similarly, the concept should ensure that traffic contention issues are handled by the controlling programmable flows within the DCN. Thus, the necessity for a control programmable platform that can implement the aforementioned conceptualisation issues is provided within the context of the control plane in Software-Defined

Network (SDN). The SDN platform advocates for the separation of the control plane from the data plane to enhance the deployment of suitable, optimizable control flow programs thereby ensuring an efficient control flow in the DCN. Moreover, SDN enables taking the available resources (switch ports, buffer, and others) into consideration while optimization is being carried out for efficient resource utilization in the process. While addressing all these, this study intends to answer the following research questions in Section 1.3.

1.3 Research Questions

The main research question in this thesis is:

Why are possible SDN responses to the inadequacy of current scalability approaches not achieving the required DCN stability?

1.3.1 The Sub-Questions

The following are the sub-questions for in-depth exploration of the main research question:

1. Why are the current approaches not able to achieve the required network stability for the future directions of DCN platforms?
2. How can the control plane be structured in relation to the topology of data planes (switches) to enhance stability in a scalable DCN?
3. Why are the existing network control management approaches incapable of providing the adequate stability that is required for the smooth extension of a DCN?
4. What network control mechanism can be adapted to enhance stable network control management with respect to the DCN?
5. How can this network control mechanism achieve an optimal and stable result in a scalable SDN data centre?

1.4 Overview of Contributions

The focus of the study is to develop a stable network control management system that is capable of reducing to the minimum the occurrence of set-up flow interference which increases the latency of light-sensitive application flow within a scalable DCN environment. This research focuses on improving the network performance of a typical DCN to attain a level of optimal network service delivery whose stability is guaranteed.

This thesis established that the stability of a scalable DCN experiencing traffic instability through network flow set-up interference can be enhanced by taking advantage of the dynamic set-up flow optimization (balancing) offered by the MOCQ approach through the ART model.

Firstly, this research explored and analysed the complexity in the set-up flow of network interference within a network. This will give us a clear overview of the various challenges that are inherent in addressing network interference and why approaches deployed by other scholars have not fully addressed the problem of network interference, especially in the DCN.

Secondly, the research discussed the application and impact of deploying an Adaptive Rendering Technique (ART) to address the interference problem that exists in the network environment. A similar principle was harnessed in a network environment to depict the traffic flow behaviours (which resembles light rays) within a DCN to address the interference problem based on its analysis.

In general, the contributions to knowledge based on the publications that emanated from this thesis are summarized into three major categories. These are:

1. **Developing a unique mathematical model for network rendering system management**:

One of the major concerns from the existing research has been the type of architectural designs for the DCN. This has inhibited the performance of earlier approaches from efficiently addressing the traffic flow interference due to inefficient flow control mechanisms being deployed. Several approaches were discovered while carrying out this study. One of them is the optical switch architecture which was found to be the best. However, because it is expensive, the Elastic-tree architecture with the hierarchical setup was found to be more suitable for the network control programming approach that is proposed in this study. The finding is significant in its contribution to rendering a suitable model for network setup during the simulation and experimentation. This was published in the article cited below:

Akinola, A.T. and Adigun, M.O., 2017. Performance Evaluation of Data Centre Network Setup Architectures. In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC), Barcelona, Spain* (pp. 166-171).

Furthermore, the occurrence of flow interference is primarily due to the inadequate and inefficient utilization of the network resources, poor buffer regulation approach as well as the lack of a stringent mechanism for low delay realisation in service requests. Literature confirmed that similar challenges were often encountered in a typical public health management system where prompt uploading, and assessment of health-related records are saved. This write up proposed a network monitoring and regulatory flow mechanism which utilises the concept of programmable network controllers to enhance an intelligent control of public data in an SDN platform. This whole concept (whose contribution was from the intelligent mathematical model) of addressing network instability was first modelled on a hierarchical SDN setup platform to test the proof of concept, the efficiency of our ideas in a Public Health Management System environment. This was accepted as a journal publication and is cited below:

Akinola, A.T., Adigun, M.O. and Mudali, P., 2020. Performance Modeling of Software-Defined Networking Paradigm in a Public Health Management System. *International Journal of Information and Management Sciences*, *31*(2), pp.123-148.

2. **Deployment of Adaptive Rendering Technique**:

After a successful proof of concept, the next idea was a step-by-step implementation of an approach that entails the deployment of both the switch-based mechanism approach and the transport layer approach to address the network flow interference issue. Research survey acknowledged from the literature the existence of the inadequacies of a single approach (switch-based or transport layer and predictive flow scheduling) while addressing the challenge of network interference. This motivated and compelled us to address the weak points left by these approaches in terms of maintaining assigned priorities and ensuring adequate fairness. I proposed the incorporation and deployment of the Proxy-Controller (P-C) approach towards maintaining network stability in a typical DCN. This is a novel contribution with the deployment of the ART approach and the outcome was published in the following publications:

Akinola A.T., Adigun M.O., Mudali P. 2020. Incorporating Stability Estimation into Quality of Service Routing in an SDN-Based Data Centre Network. In: Proceedings of Fifth International Congress on Information and Communication Technology. Advances in Intelligent Systems and Computing, ***vol 2 Issue 1184***. pp 406-416 ***Springer***, Singapore. https://doi.org/10.1007/978-981-15-5859-7_40

Akinola, A.T., Adigun, M.O. and Mudali, P.,2019. Effect of SDN Controller-Proxy Deployment on Network Stability**”** In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)* Fairmont Zimbali Resort, Ballito RSA. (1-4 September 2019)

In addition, the concept of ART was further deployed as part of the mechanism from the switch-based approach which was integral to the development of the adaptive model in this work. The approach proposed in our study enhances a series of transformations and reformations for optimal rendering, thus making it adaptable to solving traffic flow problems through adapting the network buffer mapping system. The report of this analysis is made available in the publications cited below:

Akinola A.T., Adigun M.O., Mudali P. 2020. Rate of Network Convergence Determination using Deterministic Adaptive Rendering Technique 12th EAI International Conference on E-Infrastructure and e-Services for Developing Countries. Pp. 140-150, ***Springer***,.

Akinola A.T., Adigun M.O., Mudali P. 2022. A Multi-Criteria Routing Algorithm for SDN Based Deterministic Adaptive Rendering Technique, Indian Journal of Computer Science and Engineering (IJCSE) Vol 13 No 1 pp 170-187, Jan-Feb 2022.

The stability of a prioritized network parameter among a range of network parameters becomes a challenge in literature from the perspective that most of these parameters are relatively dependent on one another and separating one at the expense of the other practically causes instability. This work was able to utilize one of the parameters that make network instability very challenging in terms of bandwidth. This research devised an ART adaptation mechanism that ensures network stability as a Specified Quality Approach (SQA). The findings also contributed to knowledge as the deployment of the adaptive rendering model has helped in the stability of the DCN. The model has ensured the maintenance of a specified network metric threshold, thus guaranteeing the performance of the data centre which formally were undeterminable based on findings from literature. The reports of these findings were published in:

Akinola, A.T., Adigun, M.O. and Mudali, P., 2019. Network Stability Based on the Amount of Available Bandwidth in a Software Defined Networking. In *2019 IEEE AFRICON* (pp. 1-6). IEEE. (2019, September).

3. **Extending and providing solutions that consider Stakeholders' interests**:

A further concern that was paramount to maintaining the performance of a network environment relates to the level of satisfaction that is derived while considering both the DCN service provider and the end-user perspectives. Thus, the missing link deepens with the importance of balancing our proposed network stability solutions such that both the service provider and the end user can protect their interests without one affecting the other. This can be enhanced by ensuring a balance between high bandwidth utilization and traffic flow stability. As much as the end user is supposed to have a great influence in terms of their desire for the best user experience with relatively fair treatment while transacting through the data centre, so also is the service provider is expected to optimize the resources that were consumed while running the network.

This research argued that these two strong observations above are very pivotal resultant effects on a stable network because they are measurable quantities to ascertain the performance of the network environment in the long run. This measure (often referred to as the Resistance Constraint) helps to determine the "elastic limit" for any network. This was found to be around 0.7 in our experiment. which is very important to the service provider to achieve desired cost maximization and on the other hand for the end-users to attain optimal user experience. Therefore, this research made another significant contribution with regards to establishing a balance of stakeholders' interest in these findings in the field of research. The report regarding this was presented for publication in:

Akinola A.T., Adigun M.O., Mudali P. Balancing Service Provider and End-User request interest in an SDN oriented Data Centre Network is currently in *process under submission and reviewing at the Journal of Computer Network and Communications*. Hindawi.

In conclusion, the various publications highlighted above are included as the chapters of this thesis. Most of the discussions emanated from the results of the contributions from each of the publications. Overall, the ART essentially contributed to the MOCQ algorithm which helped in tackling the flow interference challenge as well as provided us with an optimal solution in comparison with the existing state-of-the-art scholarly works.

1.5 Delimitations of the Study

The limitations of this research include the following:

a. The controller logic oversees the construction of a set of rules used in managing the connected switches. This forms the core idea behind the deployed SDN technique. The same traffic flow may match different rule versions but not different rules so that distinct rules cannot conflict with each other on the same flow. This research set these limitations knowing that there are several rules which might be preferred and deployed based on certain reasons. To avoid unexpected conflicts, the study assumed flows that aligned with specific versions of acceptable rules and not all the general ones. Moreover, the experiments in this study did not focus on the compatibility of different rules.
b. This research assumed that an SDN switch, e.g., OpenVSwitch, is installed in the network switches and that the switches are controlled by the SDN controller only. Literature reveals that other SDN setups distribute some portion of intelligence into the switches. However, this study addressed the occurrence of interference that deployed a central control system with a global network view as against the need for intelligent switches.
c. This research loaded flows through the Wikipedia request traces as used in Gebrehiwot, (2017) which happens to be real-world traces, were used to represent the network traffic arising from the end users. This is easy access to sending a larger number of small flows for experimental purposes which have also been used by other scholars in the field.
d. All the controllers used in each experiment are assumed to have equal processing capacity depending on the experiment that is being carried out i.e., the number of flow-in messages that can be processed per second. This was taken into consideration for uniformity and experimentation.

1.6 Thesis Outlines

The remainder of this thesis consists of seven chapters which are highlighted below:

- In Chapter 2, this write up will discuss the road to a programmable network as the key towards laying hold of dynamic and stable network implementation. This research emphasised the inadequacies of the old traditional networking approaches as being unsuitable for the kind of dynamic state that is required in modern networks. The existing data centre networking approaches were explored and classified basically into the transport layer approach, switch-based approach, predictive flow scheduling and the Ethernet techniques. The strengths and limitations of these approaches were also presented and finally, the chapter opened up the major problem that was left out which serves as the basic contribution of this thesis. This research concluded the chapter by proposing a MOCQ approach through the ART model to attain an optimal solution for DCN network stability.
- Chapter 3 introduces the DART_MCP mathematical model and its corresponding algorithm which serves as the foundation of the ART solution. The chapter describes the problem with the analysis of the network description. The various phases of adaptive rendering are discussed in the limelight of the mathematical model description to attain optimality with the DART_MCP model.
- In Chapter 4, this research presents the use of the model by depicting its usage as a Proxy-Controller (P-C) approach deployment. This deployment specified two operational modes with a working scenario that inculcates the use of MOCQ from the ART model. The performance of our P-C deployment was tested using simulation and numerical analytical analysis to determine the network fairness and relative stability.
- Chapter 5 presents the detailed simulation that addresses one of the daunting tasks of the thesis in maintaining stability/balance between the service provider and the end-user benefits. This chapter leverages on the stability of the bandwidth as a key metric to take the maintenance of network stability beyond the environment to the network beneficiaries (service provider and end-users).
- In Chapter 6, the Public Health Management System (PHMS) was used as a case study for the deployment of the proposed approach for network stability. The evaluation was done by comparing the proposed approach in our research with the other comparative selections

which were selected based on the literature as explained in Section 6.6. Several network performance metrics were explained and used to test the benefits accrued from the service provider, therefore confirming the relative maximization of the providers' resources.

- Chapter 7 documents the performance evaluation of the PHMS concerning the end users using the stability performance metrics. The evaluation was done by comparing the proposed approach in our research with the other comparative selections which were selected based on the literature as explained in Section 7.3 in the same chapter. A combination of simulation and literal test bed methodology was deployed to test the performance of the MOCQ approach on real-life patients' data. This chapter enabled us to validate the efficacy of the proposed model through the attainment of an optimal solution in comparison with others.
- In Chapter 8, this research concluded this thesis with appropriate recommendations while highlighting the researchers' contributions to knowledge. The Chapter also provides the direction for future works.

1.8 Chapter Summary

This chapter presented the background for this research study by introducing the concept of network instability due to network flow interference and how the need to design an adaptive algorithm through the ART procedure that uses the MOCQ approach to ensure that the instability in the network is alleviated.

The instability challenge is taken further by ensuring that the interests of the provider of the services were not achieved at the expense of the end-user. This chapter elaborated on the need to attain a balanced network state that maximizes the outputs while minimizing the effects of the introduced constraints towards achieving a stable DCN environment. The next chapter classifies the existing DCN stability approaches while unfolding the reason they have not been able to efficiently address the issue of network instability. It also provides insight into the reason why an ART would be a suitable approach in providing an optimal solution to the discovered challenge.

Chapter 2

TRENDS ON ENABLING STABILITY IN DATA CENTRE NETWORKING

2.1 Introduction

The demand for the use of cloud, big data, security, mobility of services and other applications have engineered the need to scale up the computer network architecture. Virtually, almost all the recent research and several daily life activities rely on and carry out most of their businesses via the Internet by connecting several devices. Hence, the performance of most of the devices is relatively dependent on the vascularized storage resource where most of the services and applications are hosted is called Data Centre. Over some decades now, the traditional networking approach has been in use to connect these computer networks, and this has gradually exposed many of its defects. Overcoming these defects has been a serious concern for both industry and academia at large. It was envisaged that if the demand for network services keeps increasing at the present rate, it is otherwise impossible to keep adopting the current networking approach hence various architectural proposals were made towards this growing interest of how the future network will look like (Kastrinogiannis et al 2010, Rehman et al, 2021).

Some of the research projects that met the requirements for the proposal with regards to future network architecture include the New Arch Project (Braden et al 2000), the new generation network project that was developed in Japan (NWGN) which consist of a series of subsets of various projects from industry and academics (Esaki et al 2007) and the Architecture and design for the future Internet (Correia et al 2011) which advocated for the need for structural change through restriction of shifting innovation to the application level on the network. These highlighted futures of network research projects and much more related research depicted to the scholars that if the foreseen development in network connectivity is to be realised, the advancements will definitely be hindered by the closeness approach which traditional network offers.

The closeness affects the network elements in various ways among which include the following observations (Esaki et al 2007, Chakravarthy and Amutha, 2022):

a. **The challenge of deploying new network protocols:** the existing closed network elements lack the standard APIs which are key for incorporating user-tailored and customizable protocols on the network. This has practically hindered the needed innovations that are required on the network, thereby preventing the required expansion.
b. **The challenge of guaranteed network service provisioning:** the need for a guaranteed quality of services is better achieved under a network that can examine the whole network under a global view approach. However, under a closed network system, such a task is difficult to achieve. The Routers only transmit depending on the current traffic load without the consideration of the quality of service that is transmitted.
c. **The challenge of managing large-scale networks:** due to the lack of a unified public network management system, it was difficult to manage a larger network while there is always a need for the network administrator to implement the configuration of a lot of network elements at once.
d. **The challenge of controlling the network elements:** the black boxes which were used in the traditional network are primarily made up of hardware and software. However, for enhanced network control in the modern networking environment, standard and open network interfaces are needed to achieve scalability and the required flexibility.

Hence, the above reasons have contributed towards affecting and discontinuing the deployment of the traditional network approach in networking environments.

2.2 Historical Perspective of Programmable Network

The closed nature of the architecture offered by the traditional network has resulted in dwindling interest in favour of a more advanced open network architecture that promotes network interface programming. Several research scholars and the industry are searching for the means of designing improved programmable and fundamental features of the future network (Aboubakar et al 2021).. To this end, the Software Defined Networking (SDN) was proposed as the fundamental basis for the construction of the expected future network in incubation (Chakravarthy and Amutha, 2022). SDN was designed to break the closed network architecture through the restructuring of the network planes by separating the control plane from the forwarding plane to enable network programmability.

The systematic progressions from the primitive approaches of computer networking towards one of the first successful SDN use cases in terms of network virtualization are depicted in Figure 2. The whole story evolved in the mid-1990s to early 2000s through the introduction of programmable functions into the existing networks to engineer some innovations. Most of the work in this period is majorly on the concept of Active networks such as smart packets, High Perf, ANTS, and many others as shown in Figure 2. The major concept of the active networking pursed two programming models which are primarily the capsule model, where the in-band in data packets carried the codes at the nodes for execution (Vasantharaj et al, 2021) and the programmable routers/switch model, where the codes to be executed at the nodes were established by out-of-band mechanisms as seen in (Zhang et al, 2021). The capsule model is often used for active networking through the installation of new data-plane functionality across the network with codes in the data packets while the programmable routers enhance the operators' manipulations through programming towards the extensibility of the network.

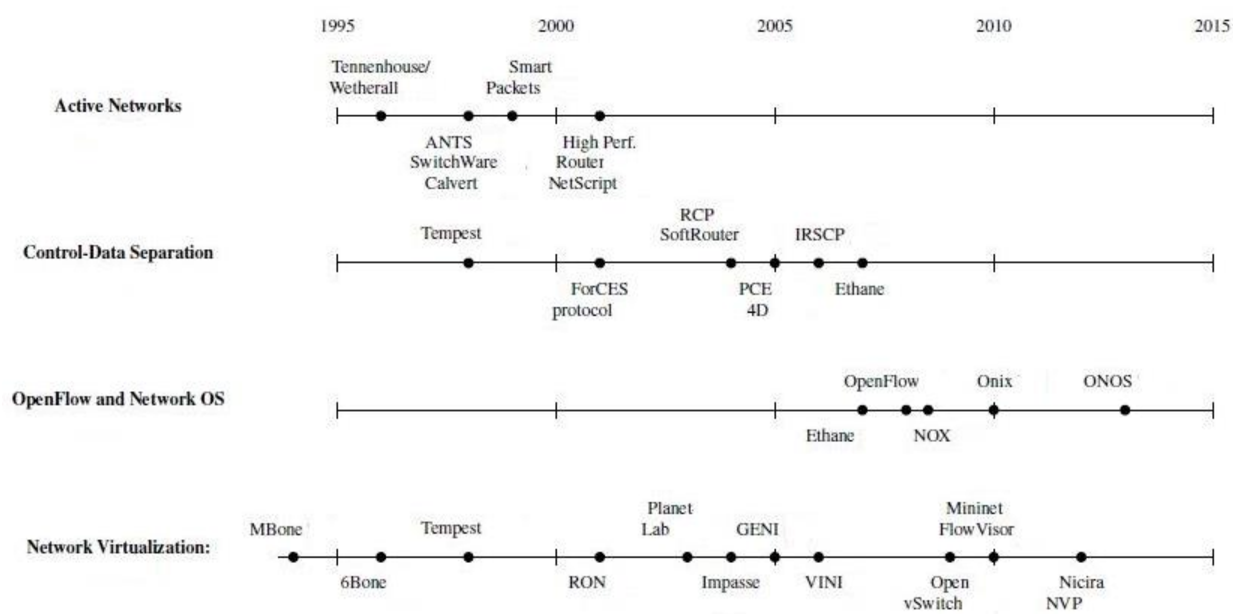


Figure 2.1: Selected development in programmable networking over the past 20 years, and their chronological relationship to advance in SDN (Feamster et al 2014).

Figure 2 also transcends from an active network towards Control-Data separation which began around the early 2000s to feature in the need for greater network reliability, predictability, and improved performance. With the need for speed at the backbone of each network to improve rapidly, several vendors and equipment owners are trying to implement faster packet forwarding logic into the hardware, thus separating the control-plane software. These trends, therefore, subdivided the innovations into two major segments thus:

a. The logically centralized control of the network which is typically seen in Routing Control Platform (RCP) (Feamster et al 2014, Ahmad and Mir, 2021), Path Computation Elements (PCE) (Merle et al, 2021), protocol at the Internet Engineering Task Force (IETF) and Soft-Routers architectures (Clayman and Sayit 2021). Some of the other proposals for the separation of control and data planes evolved from academic environments (Khorsandroo et al 2021) in support of the active network propositions.
b. An open interface between the control and data planes**,** for example, the Forwarding and Control Element Separation (ForCES) (Almadani et al 2021) has an interface standardized by the IETF along with the kernel packet-forwarding functionality in Linux (Mayer et al 2021, Anerousis et al 2021).

The realization of OpenFlow (Zhang et al 2022) in Figure 2 opened up its usage as the Network Operating System which became the first practical approach through which the control of programmable network operates. However, with these innovations in the area of network management, there are still many people and businesses which solely depend on the Internet for everyday activities and work (Mishra and Tyagi, 2022). Since the applications used by these businesses and people are usually stored in the data centre which consists of vascularized networked computer systems, it, therefore, becomes necessary that these networked devices in the data centre, as well as its resident applications (services), be properly managed.

The need for proper management arises due to several services required for different purposes of competing for limited network resources as fast as possible within the network (Maia et al 2021). Some of the resources that are in constant demand include the network buffer, bandwidth, and (or) link capacity for packet-in flow messages and while competing for these resources, the network state often degenerates creating unnecessary prolonged delay as a result of the increase in flow completion time for the execution of the request (Naouri et al 2021, Alotaibi and Nayak 2021).

Thus, this undesirable situation within the DCN is caused by the flow interference of one network on another, resulting in a term called network instability (Marin and Hampton 2019, Summers and Dinneen 2021, Gu et al 2022).

This challenge becomes prominent with the old hardware-centric traditional and coarse approach which does not enable network programming to be tailored towards solving inherent network problems. However, the evolution of the programmable network especially with the advent of the central control logic system, makes it easier to control and adapt a network to address the envisaged problems. Thus, the ability to program the control plane has made it possible to improve on the approach for addressing the network flow interference in the data centre. This usually involves the deployment of appropriate mechanism(s) on top of the control plane in the DCN thus taking the advantage to address any form of network problems. This thesis addresses network instability resulting from traffic flow interference.

In conclusion, coding appropriate flow tailored algorithm has been one of the resolutions of network engineers in recent times which speaks to improvement in network performance. This realization of a programmable network proffers several solutions that are often reliant on the manner and approach of programming to address these identified challenges. The notable ones among these approaches are grouped into four major categories as they relate to network flow management. They are explained in subsequent sections.

2.3 Existing Approaches to Addressing the Flow Interference Problem

Several approaches have been adopted in literature to address the flow interference problem which often results in instability with the network. Few among these approaches are those that deploy solutions via the transport layer, pre-emptive flow scheduling, switch-based and metric-based solutions (Chen et al 2015, Polese et al 2019, Gonzalez et al 2019, Pandi et al 2019). The target of these approaches is often to devise the means of optimizing the flow rate assignment along prioritizing scheduling thereby ensuring that the switch memory is kept empty as much as possible for the incoming flow to use. However, the daunting task with these approaches is the need to modify the end hosts' protocol stack which often incur some level of difficulties (Samaras et al 2012, Linguaglossa et al 2019, Jagannath et al, 2022).

2.3.1 Transport Layer Approach

The transport layer of the OSI model accepts data from the layer above to enhance its splitting for movement through the network layer. Thus, the major goal of this approach is to reduce the rate of completion for network flows via the transport layer control mechanisms. The adaptive rate control scheme was used by the scholars in Alizadeh et al (2019) to manage the rate of flows which are designated as a giant flow (100kb as short/small flow, above 100kb as large/long flow while 1Mb and above as giant flows).

The emptiness of the switch queue size capacity through this control mechanism ensures that the traffic flows are properly managed to prevent the possibility of interference and flow delay. A similar approach was implemented in the work of Alizadeh et al (2019) via the Explicit Congestion Notification (ECN) to the IP addresses (Ramakrishnan and Floyd 1999). Even though the current operation of the ECN to ip congestion avoidance does not rely on the number of packet drops alone, the rise in the number of the deployed applications as well as intolerance in the delay which occurs as a result of the same has made the approach inadequate for the highlighted challenge. The current research findings proved that the normal congestion notification mechanism is highly insufficient to address the challenge thus resulting in sub-optimal solutions.

With the proposition of the distributed controllers mechanism to address network latency and flow set up delays in a scalable data centre the stability of the network is often still at stake (Yeganeh et al 2019, Tootoonchian and Ganjali, 2020). The proposed layered distributed controller architecture by Yang et al (2016) with good scalable features is not without both the challenges of delayed set-up flow and stability due to interference. The research of Vamanan et al (2022) and Wilson et al (2021) both deployed the use of deadline information from the various incoming packets to set up the control rate for packet-in messages to be attended to in the memory buffer. Thus, the rate of flow of a particular packet is dependent on the ascribed deadline information to the packet in question.

A scalable ElastiCon was proposed in Dixit et al (2019) containing elastic distributed controllers which pool a dynamic control to either shrink or grow the network based on the traffic flow conditions, thus enabling the loads to also be dynamically balanced across the control plane. The proposal from the research has not been fully implemented especially in the load adaptation module section. Also, the deployment of concurrent running controllers to perform equal roles in attaining optimal performance is a tasking approach and it has a huge cost implication. The research reported in Selvi et al (2019) addressed switch-controller assignments along with the load-aware operation measure to inculcate a stringent and efficient traffic flow performance within the network. However, important parameters like queue sizes and flow table sizes were not considered in the research and this nullified the solutions' general acceptability.

A fair end-to-end window-based congestion control was proposed by Jeonghoon and Jean in Mo and Walrand (2020). It derived a protocol that is deployable for packet-switched networks with first come-first served routers. An optimisation problem in terms of network fairness was derived from a generalized proportional fairness definition which addresses the compromise between resource utilization and user fairness. However, the end user cannot decipher precisely the value of the delay experienced hence leading to problems issues in the case of rerouting in packet-switched networks. The gaps in the approach here do not make the proposed protocol suitable in addressing the network flow interference which often makes the DCN unstable.

The work of the group of scholars in Greenberg et al (2019) addressed the dynamic network resource allocation by enhancing the agility and cost-effectiveness of DCN. This proposed architecture used the flat addressing mode to allow instant placement of service requests on any server on the network. It also enabled the use of valiant load balancing as a technique in ensuring that loads are spread uniformly across the network channels and lastly, it eradicated the occurrence of complexity through the deployment of a suitable mechanism for network address resolution, especially with the large-scale server pools.

However, the measure of or reliability in terms of the stability that this architecture offered was not factored into the design hence that aspect needed to be addressed. These and several other pieces of research have been proposed in these areas to enhance better network management in terms of performance and effectiveness with efficiency. However, one of the integral aspects is the area of stability which several approaches were silent about according (Ashouri & Setayesh, 2018, Hadley et al 2017).

2.3.2 Switch Based Approaches

The network elements are one of the key devices in the network environment whose settings and designs typically influence the control and management of the network traffics. Several approaches are dependent on the management of the traffic queues using various schemes to provide efficient and guaranteed flow fairness levels for network flows. One such typical research is that of Shreedhar and Varghese (1996) which proposed an efficient way of managing the traffic queue through the Deficit Round-Robin (DRR) mechanism. The major goal of this work is to provide throughput fairness that is implementable for queuing at routers and network gateways. The research was able to achieve near-perfect isolation of traffic queue flows by satisfying Golestani's definition of throughput fairness as provided in (Golestani, 1994). The definition advocated for the use of a normalized bandwidth allocation on any pile-up flows to be made for relatively equal intervals which made the outcome of the paper more generalisable and implementable for several network scenarios.

The research finding was typically tested on datagram network packets and was found to be highly suitable and efficient in terms of its performance. Further testing was carried out using the ATM network packets with fixed cells where it was also found useful especially with networks requiring a weighted fair queuing system. However, the challenging aspect of the research had to do with the need for the packet processing task to be low thus DRR necessitates devising the size of the quantum packet to be modified into the maximum packet size to avoid the problem of delay bounds. Another challenge reflected the need for the test to be performed in a typical DCN to ascertain its effectiveness in terms of performance. Most of these solutions were designed for the traditional LANs as well as local routers and switches with the major characteristics of the DCN in mind.

The efforts of scholars in (McKenney, 1990) also refers to the management of the network traffic flows through regulating the Transmission Control Protocol (TCP) via a class of probabilistic variants of Shenker's fairness algorithm for queueing. It was found very suitable when there was a need for trade-offs in terms of memory, CPU capacity as well as fairness performance especially with regards to network communications with high-speed movements. The availability of several resources has proven to enable the approach under this research to attain a good output concerning queueing fairness however the main goal of many of the network challenges is to see how to effectively and efficiently manage the insufficient resources with the deployment of proven methods and algorithms for proper management.

One of the highly related useful research is that of (Shpiner et al 2012) which uses the Hashed Credits Fair (HCF) to resolve the network flow interference challenge through providing a proven scheme for the management of switch queueing better than the already discoursed works under this subheading. It performs highly better than the traditional output queue (OQ) flow which has been existing earlier in the sense that a transparent lightweight data centre switch algorithm was designed based on a fair TCP flow with credits which was able to attain an appreciable time complexity of 0(1) under less significant resource consumption. Furthermore, the HCF approach increased the goodput of the small TCP traffic flows which are usually affected by network incast challenge and at the same time alleviated the long TCP starvation challenge. However, one of the effects of using the hash flows is the aspect of uniformity in the use of a hash function which sometimes attains a static state that could invariably lead to incurring more hardware costs on the network setup.

Fine-grained work on the retransmission of TCP during data centre communication was discussed in the work of Vasudevan et al (2009) which uses practical analysis to eliminate the retransmission of timeout bound which could result in undesirable congestion backlogs with TCP incast collapse. The realization of this approach was tested both in simulation and real-world experiments with the high-resolution timers being able to eradicate the microsecond granular traffics within the specified timeouts. Though this work is well adopted to address management problems in the data centre, however, the aspect of interference of network flow which results in an unstable network was not addressed. Therefore, our work addresses this weakness in providing stringent low delay traffics which can enhance higher capacity requirements in the same vein guarantees a degree of stability.

2.3.3 Predictive Flow Scheduling

Predictive flow scheduling happens to be one of the views to combating the network management challenges based on literature (Chen et al 2015). Several pieces of research have utilised various flow scheduling mechanisms by considering a certain number of priorities to reduce the rate of flow completion time and address flow congestions alongside flow interferences. Typical among such research is the report from Alizadeh and others (Alizadeh et al 2019). The research proposed a transport design called pFabric whose main concept was to ensure that the data centre flow transport must decouple the flow scheduling entirely from the rate control. The research shows that by devising solutions separately for both the rate control as well as the flow, scheduling would go a long way towards outstanding performance in DCN management. Further investigations from the research affirm that larger buffers and complex rate control do not ultimately determine improved performance but rather the implementation of simple mechanisms which does not follow the improved throughput and bandwidth-centric TCP goals.

However, the change in the authors' view subjected the research to reduced stability in some aspects of the network as a result of size-based traffic prioritisation deployed in the study. Though the authors claimed that the issue of stability has something to do with linear network topologies, it becomes evident that the trade-offs incurred in prioritizing the small flow imbalances the flow which is destined for a longer route. Another open challenge to this work is the issue that is related to the Nondeterministic Polynomial-time problem (NP-hard problem) which is incurred in the computation of the global optimal flow scheduling for the separated goals in the work.

The research documented in (Hong et al 2022) discourages the use of a fair sharing approach in addressing the low latency problem in the data centre but proposed a Preemptive Distributed Quick (PDQ) flow scheduling protocol for managing the network flows and enhancing the promptness in meeting the flow deadlines. The idea of this approach is to approximate the range through emulating the shortest task first algorithm and enable the small flows to gain upper hand with the aid of certain priority measures. The issue of path diversity was also addressed with the enhanced multipath version of the proposed PDQ scheduling approach. The result of the research showed a tremendous improvement on the existing research such as (Dukkipati and McKeown, 2020) and (Wu et al 2012) in terms of resilience to packet loss and preservation of performance gains. However, at some point, the latency for scheduling for short flows are longer than desired thus enabling the interference of the small flows on the long flows to occur. Moreover, this approach is an almost clean-slate solution that needs a new end-host protocol stack and possibly the deployment of a switch hardware design that is unrealistic on the other end.

Furthermore, the authors in (Verloop et al 2005) illustrated the effects of stability concerns in the size-based scheduling approach to multi-resourced systems flow control. The approaches such as Shortest Remaining Time First (SRTF) and the Least Attained Service First (LASF) were tested and the varied stability conditions in various limiting regimes were deciphered in the course of the experiments. The reported research revealed that the available resources were inefficiently utilized, resulting in an instability effect especially on the giant network flows dues to performance degradation that is experienced. Hence, this task requires more than just a scheduling approach but also incorporation with a dynamic, intelligent system that can adapt to the present situation of the network at hand.

2.3.4 Ethernet Techniques

A standard prototype under this technique proved to be very good although it comes with a daunting implementation challenge. The report from Ek (1999) depicted the proposal of 802.1/Ethernet related techniques which could help to eradicate the flow interference issues in the DCN. Currently, the IEEE 802.1p standard has 3-bit enabled priorities to favour the differentiation of at least three major traffic flows while operating on the same link. Thus, it becomes easy to identify which of the flows should be saddled with the priorities and so on in other to maintain optimal flow performance. The major challenge with this standard as earlier depicted has to do with the deployment of the substantial difficult areas of the prototypes as the current switches and network hosts are incapable of some important features of the techniques.

The fairness of Internet impact was examined in the research by Bonald and Massoulié (2001) for its performance evaluation. The bandwidth sharing was proposed as a fluid model so that the impact of flow dynamics is well taken care of for stable network performance. The analytical results depicted that a class of fair bandwidth allocation ensures efficient use of the network resources in that the summation of the competing flows requirement within a particular network remains finite as far as the request is less in comparison to the capacity of the available links. However, the approach in this particular research is meant to be operated on a fewer number of network nodes, especially on specific network topology so that its effect on a data centre is not validated and proven.

2.4 Gaps in the existing works

Network stability is a challenge that is typically common as depicted from several reviewed research works (Marin and Hampton 2019, Summers and Dinneen 2021, Gu et al 2022). Several of these research works have been reviewed in section 2.3. Some of the reviewed pieces of research deployed the Transport Layer Approach (i.e., adaptive rate control schemes, Explicit Congestion Notification), Switch Based approach (i.e., Deficit Round Robin, deployment of fairness through queuing routers and network gateways), Predictive flow scheduling and the Ethernet Techniques. However, this study deduced from literature that none of these types of research evaluated the network stability while optimising the benefits regarding the service providers (the owner of the data centre) and the end-users. Literature also provided some reports that either dwell on maximizing the benefits of the service provider only and others on the end-user experience only. This research summarise some of the literature below to briefly discuss the solutions that were provided by the scholars as they relate to the gaps in the current discussion.

2.4.1 Maximisation of Service Provider Benefits

Boloor and others in Boloor et al (2019) presented a novel approach to data-oriented dynamic service-request allocation using the gi-first in first out (gi-FIFO) algorithm thereby improving the performance of the data centre through increasing the derived bandwidth utilization to enhance higher profit for the owners. The results from the research showed that the proposed approach performed better than other static allocation approaches like weighted round-robin and the ordinary first in first out options. However, as this research explained in the introduction of this section, the approach does not give the audience to the respective end-user experience in terms of the stability performance accrued. The research by Xu and Li in Xu and Li (2012) presented an effective service request allocation distributed algorithm based on sub-gradient and dual decomposition methods that appropriately channelled request flows in a manner that both the energy cost as well as the network bandwidth was properly managed. In the same vein, this also is a provider-oriented approach with no report on the stability of the flows on the end-user side.

The research documented in Liu et al (2019) depicted the process of maximizing the gains from energy consumption through the use of green energy while carrying out flow allocation policy on the network requests. The proposal first delivers two distributed algorithms for deriving an optimal geographical load balancing and in addition ensures that the load balancing enhances significant reduction in the brown energy use. Again, the research focused on the providers' benefit maximization leaving behind the end-user whose satisfaction could not be guaranteed either in terms of the level of stability experienced or otherwise.

Another research that maximised the benefits of the provider while carrying out the balance between traffic flow and the data centre capacity to process the flow effectively was the work of Gao and others in Gao et al (2019). The research advocated for the design of a request-routing framework called FORTE which provides three-way trade-offs in terms of carbon footprint, power cost, and the latency accrued. This framework allows the network operator to navigate the three trade-offs in response to the sudden increase in the amount of traffic flow experienced. The last work on this section also follows a similar fashion as it provided a cutting edge to reduce the power consumption bill for the Internet-scale systems.

2.4.2 Maximisation of End-User Experience Benefits

Few pieces of research were reported while reviewing the literature under this section among which includes the work of Xu and Li (2021) which was based on joint request mapping and response routing with multiple distributed data centre environments. An optimisation problem was formulated to address the overview workload that the system needs to address. This was solved by a distributed algorithm that uses the Alternating Direction Method of Multipliers (ADMM) approach. This approach enhances the parallel implementation of flow requests in a data centre after carrying out a decomposition-coordination method. The algorithm converges faster after an average of ten iterations to produce a near-optimal solution of the whole problem. Real-world workload traces were used to test the performance of the proposed approach with the result showing greater efficiency in comparison to the existing systems. The end-users, therefore, experience fairness in the process of request processing as a result of the alternating directions method of multipliers used for the allocations.

The approach in Wendell et al (2021) proposed a DONAR prototype, a distributed system that provided the interface for specifying mapping policies to avoid the burden of replica selection while assigning servers to users' requests in a data centre. This was made possible with a simple running algorithm that efficiently coordinates the possible replica selections of the users. The approach solves an optimization challenge that addressed both the user performance and the server load thus proving that the approach enhances an effective and stable system. Looking at the several pieces of research that have been highlighted in these sections, this reserach understood that the increase in the traffic flow within a data centre is inevitable and as such, it comes with the challenge of still maintaining an optimal performance even with the limited available network resources.

2.5 Major problem to be addressed in this thesis

Once applications are deployed on an SDN-based data centre which is managed by several controllers, the management of traffic flows to avoid network interference becomes a major problem as this often results in network instability to both the data centre platform provider and the end user as well. This concern is related majorly to poor network performance. The challenge of interference in traffic flows, therefore, revolves around the process of efficient resolution towards the management of massive traffic flow requests among the multiple distributed controllers within a data centre. Hence, apart from alleviating network instability as a result of the occurrence of flow interference, two basic goals must also be addressed from both the end-user perspective as well as that of the data centre providers.

Firstly, high bandwidth utilization and traffic balance must be attained from the general overview of data centre providers. Secondly, the end-users must have a great influence in terms of desire for one of the best user experiences coupled with fair treatment while transacting through the data centre. This research argued two strong observations for these basic goals considering the large-scale Internet application platforms such as social networking (Facebook), data retrieval (Google), and bandwidth contending video streaming interface like (Netflix). Google for example among these platforms retrieves several millions of requests each day, and they have already invested a lot of money in building data centre servers to the tune of 450,000 servers in their various data centres for efficient services (Guo et al 2018). Therefore, it is very important that such a big investment is not overloaded to the extent of being susceptible to failures and eventual poor performances which will be intolerable to the end-users (Singh et al 2018). In the same vein, every resource that is acquired must be optimally utilized such that demands like bandwidth is on high utilization mode and is evenly assigned to the multiple controllers to avoid overwhelming some groups in comparison to others in the data centre. The second observation for this motivation is also that of business enterprises where customers who are primarily the clients/end-users are supposed to be on optimal user experience level to keep them perpetually in demand by ensuring services are acquired at low cost, that there is fairness in treatment as well as low latency (Buyya et al 2010, Mahmud et al 2019).

In conclusion, addressing this kind of gap in a network environment usually requires combinations of mechanisms for effective and efficient performance to be guaranteed. Looking at the highlighted research problem, the study placed priority not only on the approach used in solving the problem but also on ensuring the satisfaction of the participating stakeholders in the entire system.

2.6 Chapter Summary

This chapter contains the existing research in the field of computer network especially the ones which deal with network management. The research explained that network management is vital to the networking environment because of the rate at which the demand for connectivity increases. However, the old hardware-centric approach to network management will not be able to meet up these current challenges. This creates the need for a programmable network platform. The programmable network has grown over the years to the evolution of Software Defined networking which advocates for the separation of the control plane from the data plane. However, as the flows from the network elements (switches) are directed towards the controller(s), there arises a state at which proper management must be in place to avoid flow interference which results in the instability of the network both for the end user and service provider. This research acknowledged this gap alongside other scholars, as they also confirmed it as a network challenge. This study devised an appropriate solution to address the identified challenge. The Software Defined Network platform enhances a fast deployment of programmable control flows in the network as soon as a new flow control model is developed. This will be discussed in the next chapter.

Chapter 3

RESEARCH DESIGN FEATURING ADAPTIVE RENDERING TECHNIQUE

3.1 Introduction

Chapter 2 examined the research problem area and the domain addressed in this thesis by highlighting traffic flow interference problem exists when there is an increase in the number of requests without an appropriately designed mechanism to address it (Huque et al 2019). This is usually caused by the effect of the spurious increase in the number of devices that access the Internet, especially mobile devices. This research also reiterated from literature that several approaches have been adopted but that the optimization of the solutions with consideration to both the service providers and end-users has not been fully addressed. This, therefore, left the existing solutions relatively biased, favouring either of the stakeholders and not both (Zhang et al 2015). Thus, the requirements for optimal performance of each running application as it relates to the resources provisioning and cost incurred while subscribing, need to be tailored in a proposed fashion that renders the optimal quality of service on demand (Huque et al 2019, Abdallah et al 2018, Akinola et al 2015).

To this effect, the contribution of this chapter is mainly in the formulation of the research design using ART. The chapter explains what the ART entails along with its operational model as used from the computer graphic field. Other sections detail the evolution of the Deterministic ART Multi Constrained Path model that made use of the MOCQ approach in addressing the flow interference problem.

Therefore, this chapter details the mathematical analysis alongside the algorithm that gives birth to the deployed approach in addressing the traffic flow interference within the network. Section 3.2 details the problem description while Section 3.3 describes the mathematical expressions of the rendering approach. Section 3.4 presents the parameters of the optimal DART_MCP model. Section 3.5 details the algorithm of the mathematical expressions and concludes the chapter while the chapter summary is in Section 3.6.

3.2 Problem Description

This section investigates the background network topology as well as the concept behind the proposed topology resolution for network analysis (He et al 2017, Song et al 2017, Yeganeh et al 2013). This helps to lay a strong foundation for the deployment of an adaptive rendering approach for the proposed multi-criteria network user requirement.

3.2.1 Network Description

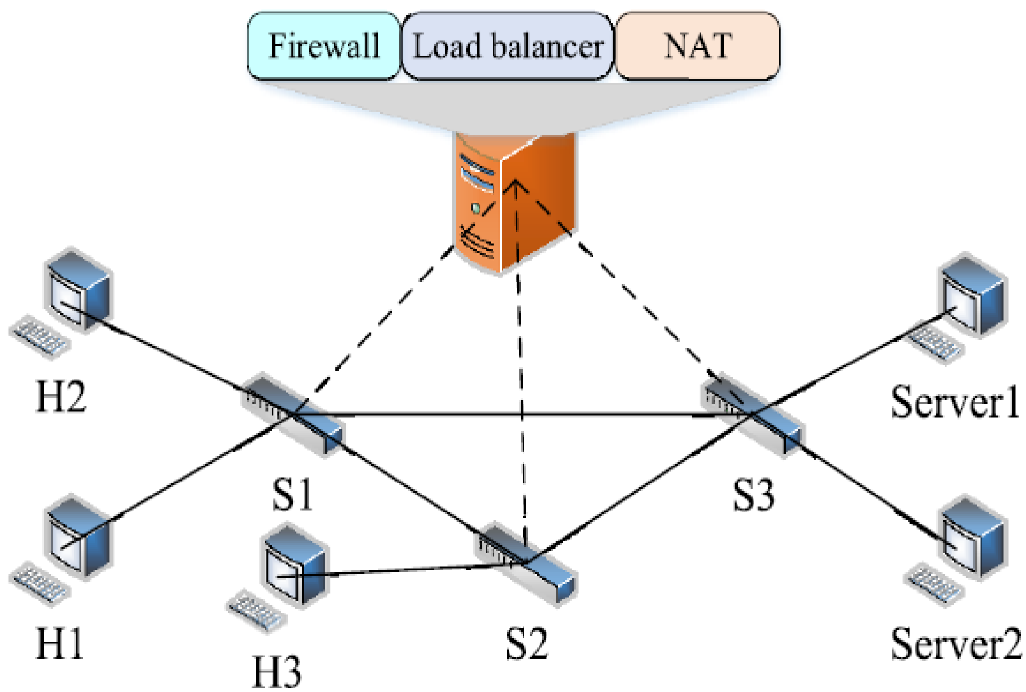


Figure 3.1: A Simple SDN Network Topology (Zhang et al 2015)

Considering a simple SDN topology shown in Figure 3 that contains major components which consist of the Controller, having the Application plane deployments on top of it (Firewall, Load Balancer, and NAT). Below the control plane (Controller) are the Data planes which are majorly the various network elements (Switches and Routers) and several network hosts. The major task within this simple topology is to ensure smooth flow communication from one host to the other while sending information. The proper analysis of this simple network topology could help in addressing larger networks like data centres which consist of numerous interconnected controllers and network elements. With this in view, this section analyses the simple SDN network topology to achieve a simple and easy to comprehend weighted graph in Figure 4. The graph shows the connected network hosts and the magnitude of the network statistics flow requirements that are attainable in each of the connected links. The scenario is to depict the need for a balanced link metrics on each of the interconnected links shown in Figure 4.

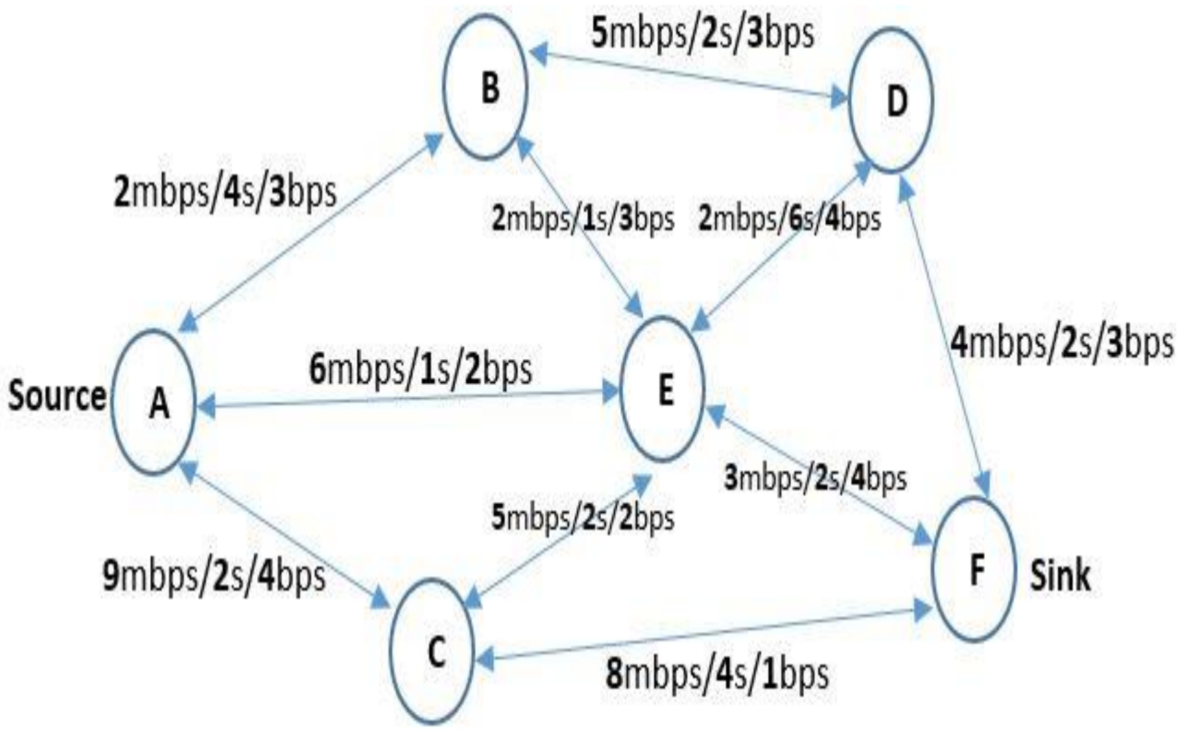


Figure 3.2: Skeletal topological network routes with varying requirements from source to sink. (Space constraints)

Let G = *(V, E)* be a weighted digraph, with weight function w: E → **R** mapping edges to real-valued weights. . If *e* = (*u*, *v*), writing *w* (*u*, *v*) for *w*(*e*). With this, denoting the path length *p* of a network as $p = < v_0, v_1, .., v_k >$ which gives the sum of the weights of its constituent edges using a typical network route traffic which is represented in equation 3.1 (Korkmaz and Krunz, 2001) (Zhang et al 2013)

$$L\ (p) = \sum_{i=1}^{k} w(v_i - 1, \quad v_i) \tag{3.1}$$

The distance from u to v, denoted by $\delta(u, v)$ provides us with the minimum length path if there is a path from u to v, and it is ∞ if there is none. Therefore, taking a reference estimate from $d[v]$ of the length $\delta(s, v)$ of the shortest path for each vertex v, the expressions $d[v] \geq \delta(s, v)$ and $d[v]$ always equal to the length of a known path. This implies that at the initial state, $d[s] = 0$ and all the $d[v]$ values are set to ∞. The validation of the real shortest distance is confirmed on a condition at which $d[v] = \delta(s, v)$. Therefore, combining the path from s to u with the edge (u, v), this research obtain another path from *s* to *v* with length $d[u] + w(u, v)$. If $d[u] + w(u, v) < d[v]$, I, therefore, replace the old path $< s, \ldots, w, v >$ with the latest short path $< s, \ldots, u, v >$. Hence, the path is updated.

Furthermore, this research introduced an energy constraint function that depicts the range of constraints that are combined for the efficient path required to be used. The purpose of these energy constraints is to incorporate the appropriate QoS routing metrics to enhance possible network stability to solve the earlier mentioned network challenge. Considering the path energy function (E) as expressed in equation 3.2:

$$f(E) = max_{k=1}^{K}\left(\frac{w_k(E)}{c_k}\right) \tag{3.2}$$

where $c = (c_1, c_2, \ldots c_k)$ represents the *k* QoS constraints for feasible path requirements. Several constraints like flow overheads, network delay, transmission failure, flow setup time, and many others depending on the network requirement for the particular application in question can be deployed. Therefore, in streamlining the network topology concerning the energy function which inculcates the parameters of the constraint, this research defines the shortest path from a source node to a sink node as depicted in Figure 3.2.

While equation 2 is subject to:

1) $\forall p \in P', W_i(p) \leq C_i \; for \; i = 1, \ldots, m$ and

2) $W_k(P) \; is \; minimized \; over \; all \; feasible \; paths \; with \; respect \; to \; the \; condition \; 1 \; above;$

This gives the expression in equation 3.3 as:

$$w(p) = \sum_{k=1}^{k} w_k(p) / C_k \tag{3.3}$$

Hence, the goal is to first determine a path that considered all the constraints for the network user requirements in such a manner that the minimum requirements are achieved. Secondly, the parameters to address the possibility of flow interference are included so that stability is maintained. In this case, the number of parameters for stability maintenance occurs at any moment when $(p) > k$. The proposed approach to addressing this problem description by analysis is the ART which is further explained in Section 3.2.2.

3.2.2 Adaptive Rendering Technique (ART)

The ART specifies the approach used in a graphic computing system to temper the quality of a picture to the required pixel and aspect ratio that is needed so that it looks the same or similar to the real object. Several steps were involved in the algorithms that are deployed in achieving this technique which involves transformation, rasterization, fragment shading, and frame stability. The technique is termed adaptive because significant noises were removed from the graphics with almost similar fidelity maintained in the course of transformation and reconstruction.

One of the main goals for deploying an adaptive rendering approach is to maintain the speed of frame generation while ensuring the realism of the object on the other hand. This goal, therefore, introduces a measure of trade-off in the algorithm(s). These series of steps (stages) that are seen in the ART that enable it to achieve the maximal speed and realism are unique and typical. The uniqueness is envisaged as useful and deployable in addressing flow interference problems to enhance low latency networks with a high degree of network success rates (throughput). The removal of noise in ART is similar to the clearing of the network buffer of the queue which could eventually impede the stability of the network.

It was with these motivations in view that this thesis addressed the stability of a scalable DCN environment that is caused primarily by traffic flow interference. The ART approach is popular and is a proven approach used in Computer graphics for optimal Image rendering (Wang and Dey 2010). The ART would be implemented on a proven design mechanism (Hierarchical data centre design) to address this problem. To this end, the solution idea in this thesis is referred to as a hybrid of switch-based and Transport-Layer approaches. These are basically among the approaches that were discussed in Chapter 2 under the existing solutions. The idea implemented in this hybrid is not the same as the existing ones. The mode of operation of the ART is summarily depicted in Figure 5 as a continuous flow of loops until the Controller and Switch Stability section has been proven to have attained the required stability or else the loop will continue with the optimization processes.

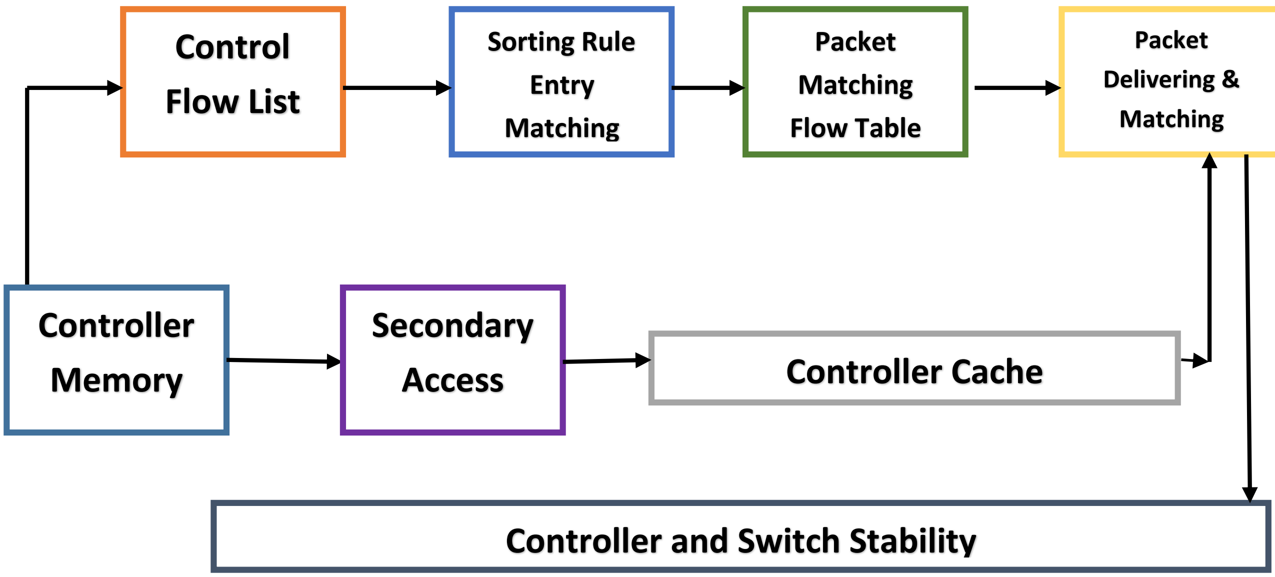


Figure 3.3: Primary stages of control flow rendering adaptation pipeline (Wang and Dey, 2010)

The flow originates from the controller memory section of the loop. The primary stages were initially developed to address the communication and computation constraints in computer graphics according to the publication by Wang and Dey (Wang and Dey 2010). In the case of this thesis, the study would be using it to address the optimization of multi-objective constraints. The figure shown in Figure 5 depicts the reformed primary stages of the flow pipeline in modern controllers. All the statistical data for one flow are first cached in a control flow list (display list).

When the control flow list is executed, the data is sent from the list as if it were sent by an application. The statistical data (bandwidth, delays, response time) are all represented in magnitude and directions as constraints for the rendering solution. Based on the rendering pipeline in computer graphics, the stages that were undergone include display list, vertex shading, rasterization (filling into fragments), and raster operation which eventually resulted in the final stable frame buffer. Similar to this pipeline in computer graphics is the series of pipelining processes listed as the Control flow list, sorting entry matching, matching flow table, and matching operation respectively in our adaptation for rendering usage in computer networks. However, as against a stable frame buffer which is attainable in graphics, our goal in this primary stage pipelining is a Controller and Switch Stability state in networks. Having understood the concept of ART, this study further opened up the various forms of ART and the evolution of the appropriate one that applies to this thesis.

Firstly, various forms of physically-based rendering algorithms are used in solving the global illumination problem in computer graphics (Lafortune and Willems 1994). The algorithm determines the complex inter-reflection of light through a particular scene which is similar to the situation that is seen in a typical network environment where network interference becomes a significant challenge in addressing user network requirements provisioning. The global problem is addressed by the physically-based rendering algorithms which are classified into two major types:

1. Object-Based and Image-Based Algorithms: The object-based algorithms solve the global illumination problem independently of the actual parameters. The key strength has to do with the deployment of the hierarchical algorithm, use of higher-order basis functions as well as some kind of wavelet algorithms for the implementation. On the other aspect, the image-based algorithms considered the actual parameters of the image like the stochastic ray tracing algorithm which considered each pixel for radiance value computation. However, based on performance, the speed of the generation of the interactive frame is a bit slow in this classification. For this reason, its deployment in our solution approach will not be favourable.

2. Deterministic and Monte Carlo Algorithms: The Deterministic approach uses classical numerical techniques to address the global illumination problem. It uses the combination of solution methods for large sets of linear equations and some cubature rules for integrating high-dimensional functions (Lafortune, 1996). On the other hand, the Monte Carlo rendering algorithms deploy stochastic techniques to work out the high-dimensional integrals of the global illumination problem (Lafortune, 1996). The deterministic approach is more suitable

for our kind of network challenge because, in addition to its use for numerical analysis, it can carry out a larger set of high-dimensional functions at a faster rate. To this effect, this research proposed the Deterministic ART Multi-Constrained Path (DART_MCP) solution for our network challenge.

Generally, the rendering algorithms primarily use three steps to resolve any rendering problem. However, this research modified the steps with the inclusion of two additional steps to suit the purpose of our study. Therefore, the proposal listed five major steps in addressing our problem through the rendering equation approach which is referred to as the Multi-Objective Optimization of the Crosspoint Queue management (MOCQ). These steps include:

- Identification of the physical (network) topology
- Formalizing the topology problem in a mathematical model
- Developing the algorithm to solve the designed model.
- Solving the mathematical model, conducting its simulations (as well as experimenting).
- Comparing the results from all these approaches to ascertain if Adaptive Rendering is a novel solution to the problem of flow Interference within the DCN environment.

The derivation of the MOCQ approach from the ART is diagrammatically depicted in Figure 3.4.

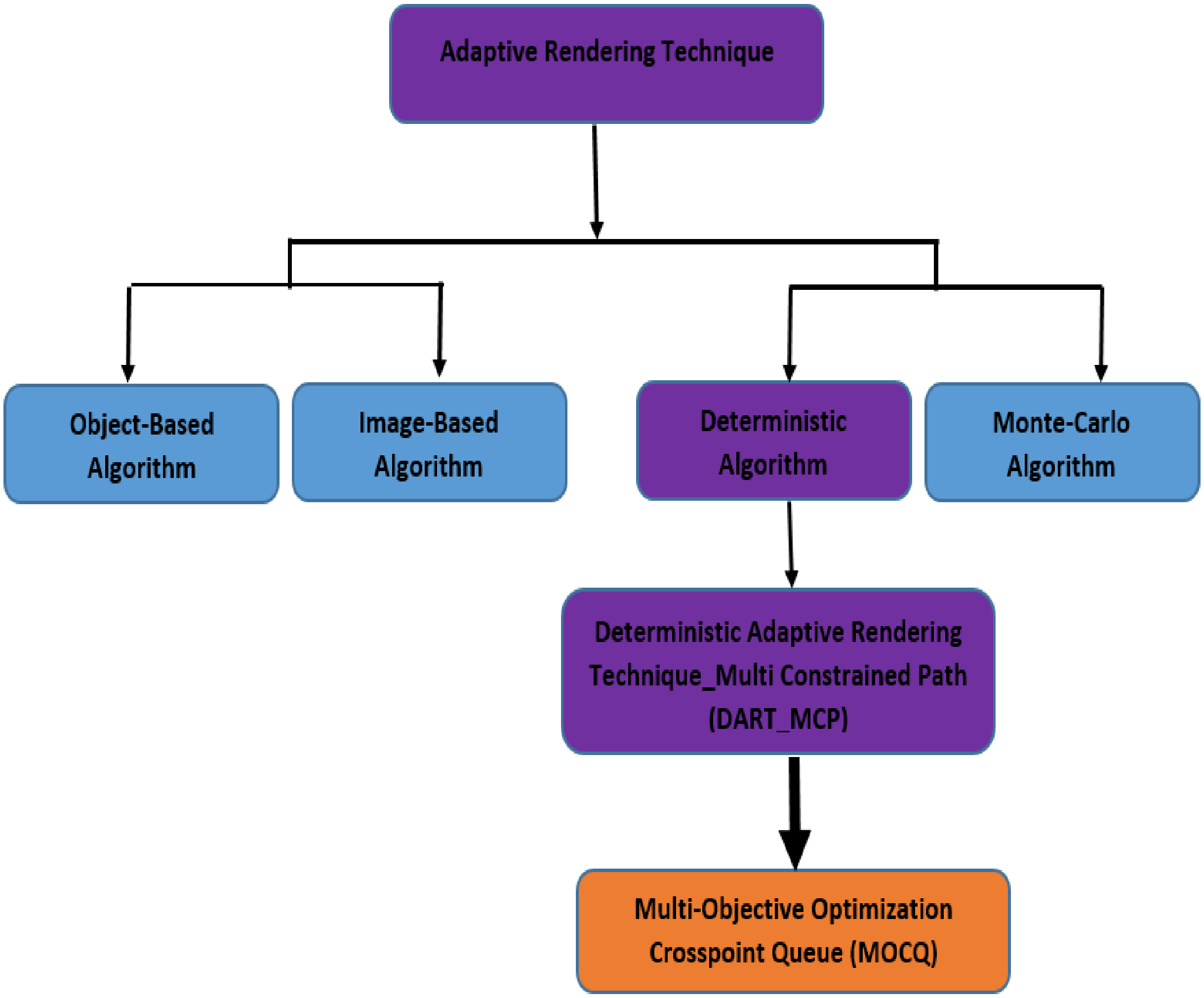


Figure 3.4: Derivation of Multi-Objective Optimization Crosspoint Queue (MOCQ)

The highlighted stages above form the philosophy underlying the adoption of the ART in solving the challenge of network interference. This philosophy is hinged on the ART mathematical model of solving problems that need adaptive mechanisms deployed through MOCQ. The ART model uses two major procedures which include the Scan Conversion stage as well as the Rendering stage. Considering the nature of the technique and the requirement for the DCN, the need for dynamics is important. This research, therefore, modified the ART approach to reflect the dynamic nature of using the DART model for solving the network interference. The next section below gives the analyses surrounding the ART approach as the solution to the concerns of this thesis.

3.3 DART-Model Analysis (Mathematical Model Expression)

There are three main types of messages within SDN network flows within the DCN environment which assists in conveying data flows between the data plane and the network controllers. These are majorly symmetric messages, asynchronous messages, and controller-to-switch messages (Karakus and Durresi, 2016). These messages were used in developing the DART_MCP algorithm which shall be explained sequentially later. However, as their names imply, the symmetric messages are typically the echo and hello messages which do not need any solicitation from a controller. The asynchronous messages are typical messages that are sent to the controller in response to the reception of packets by the switches such as flow removed and packet_in messages.

The last and commonest type of message is the controller-to-switch message which is responsible for delivering messages to the switches without any acknowledgement. The analyses of the mathematical model supposedly deploy the buffer Monte-Carlo which assists in establishing the stability of the network performance in connection with the effects that are generated from the network interferences. The buffer approaches the versatility of distributed networks as it can render a wide influx of transferred packets while maintaining the QoS of varying network users. This is carried out in two major phases which are scan conversion and the rendering phases *(*Lafortu*ne, 1996).* These are highlighted in the sections below:

3.3.1 Scan Conversion

The phase of the analyses at this stage estimates the range of packets that are sent over the networks to harness the scheduling of the paths for preventing interference. Thus, considering the general deterministic Software-Defined enabled Network, which is measured from the source in the presence of *n* interferes at distanced $r_i > 0$ that are active with the probability *p* independently of each other. The path loss law is the standard power law (considering normalized paths – edges) which is expressed as shown in equation 3.4:

$$((r)) \text{ i.e } e(r) = r^{-\alpha}. \tag{3.4}$$

Furthermore, among other things to be considered are the location of the controllers being predefined and fixed, this research assumed that there is a path available for a switch to send packet-in messages when paths are computed by using the Dijkstra algorithm. The aim is to alleviate the sharing of flow paths and network channels and make the network channels subject to Rayleigh fading which is exponential in power fading rate (Bjornson and Sangunetti, 2020). Therefore, the power is represented by P_{ri} that is caused by the rate of packet traffic i, with the distributed exponential mean $r^{-\alpha}$, then this approach express the probability of density function of P_{ri} as in equation 3.5 thus:

$$fP_{ri}(x) = \; r_i^{\alpha}\exp(-r_i^{\alpha}x), \quad x \geq 0. \tag{3.5}$$

Therefore, the scanning conversion phase determines the overall interference by the inter and intra flow packets within the network as the summation represented below in equation 3.6 as:

$$I_n = \sum_{i=1}^{n} B_i \, P_{ri} \tag{3.6}$$

where the Bernoulli random variables *Bi* is with the varying parameter *p*. Hence, the Laplace transforms this exponential random variable of the transferred packets with *mean* 1/y is $\int(s) = \frac{y}{y+s}, s \; \geq 0,$ resulting into,

$$\int_{In}(s) = \; \prod_{i=1}^{n}\left(\frac{pr_i^{\alpha}}{r_i^{\alpha}+s} + 1 - p\right) \tag{3.7}$$

$$= \prod_{i=1}^{n}\left(1 + \frac{p}{1+\frac{r_i^{\alpha}}{s}}\right), s \; \geq 0 \tag{3.8}$$

in the above expressions. There is a need to maintain a proper network *packet distribution* especially when the number of network nodes *n* seems to be relatively *infinite.* To this effect, equation 3.8 portrays a uniform convergence as it converges to some *positive limit* such that for any $s > 0$ if and only if

$$\sum_{i=1}^{\infty} \frac{p}{1+\frac{r_i^{\alpha}}{s}} < \infty \; (\forall s > 0) \; <=> \; \sum_{i=1}^{\infty} \frac{p}{r_i^{\alpha}} < \infty. \qquad (3.9)$$

Thus, the equation in 3.10 gives an even distribution for the network packets but the interference of the network is infinite, then

$$\sum_{i=1}^{\infty} \frac{p}{r_i^{\alpha}} = \infty, \qquad (3.10)$$

Hence for a one-dimensional network setup, the scan conversation is determined while $r_i = i, \; i \in$ N, and $p > 0,$. I achieved an interference of $\alpha > 1$ for a finite network and $\alpha \leq 1$ for an infinite network being a multi-dimensional network setup.

3.3.2 Rendering Phase

For the rendering phase, considering an SDN based DCN whose Rayleigh (amplitude) of fading due to interference varies toward infinite one-sided one-dimensional network path $r_i = i, \; i \in$ N, allowing a close expression for $\int_I(s)$ Laplace expression. Hence, for $\alpha = 2$ and more, the expression is thus:

$$\int_I(s) = \frac{\prod_{i=1}^{\infty}\left(1 + {}^{(1-p)\,s}/_{i^2}\right)}{\prod_{i=1}^{\infty}\left(1 + {}^{s}/_{i^2}\right)}, \qquad s \geq 0. \qquad (3.11)$$

Recalling the Euler's product formula, this can be applied here to simplify further the Laplace expression in equation 3.12 thus:

$$\sin(\pi z) \equiv \pi z \prod_{i=1}^{\infty}\left(1 - {}^{z^2}/_{i^2}\right), \; z \in \mathrm{C}, \qquad (3.12)$$

While the denominator carries z = $j\sqrt{s}$ and the numerator that is equal to z = $j\sqrt{(1-p)s}$, having to rewrite the denominator as:

$$\prod_{i=1}^{\infty}(1 + s/i^2) = \frac{\sin(\pi j\sqrt{s})}{\pi j\sqrt{s}} = \frac{\sinh(\pi\sqrt{s})}{\pi\sqrt{s}}, s \geq 0, \qquad (3.13)$$

thus, deriving the expression,

$$\mathcal{L}_I(s) = \frac{1}{\sqrt{1-p}} \cdot \frac{\sinh\left(\pi\sqrt{s(1-p)}\right)}{\sinh\left(\pi\sqrt{s}\right)}, \qquad s \geq 0. \tag{3.14}$$

Hence, for a higher degree of interference amplitude, this research render a more reliable equation to enable the listening nodes for packet scheduling accordingly with the expression:

$$\mathcal{L}_I(s) = \frac{\prod_{i=1}^{\infty}\left(1 + {}^{(1-p)\,s}/_{i^4}\right)}{\prod_{i=1}^{\infty}\left(1 + {}^{s}/_{i^4}\right)} \tag{3.15}$$

and the value of α, in this case, is 4. By factorization, using Euler's product formula again since the expression in equation 15, states $(1 - z^4/i^4) = (1 - z^2/i^2)\,(1 + z^2/i^2)$. The factorization derived these expressions $z = \sqrt{\pm j s^{1/4}}$ with $z = \sqrt{\pm j\left((1-p)s\right)^{1/4}}$ for the denominator and numerator respectively. Therefore, the resulting expressions give complex conjugates, resulting in:

$$\mathcal{L}_I(s) = \frac{cosh^2\left(\sigma(1-p)^{1/4}\right) - cos^2\left(\sigma(1-p)^{1/4}\right)}{\sqrt{1-p}\,(cosh^2\,\sigma - cos^2\,\sigma)} \tag{3.16}$$

Thus, equation 3.16 gives the resultant rendering expression in any network scenario where the impact of interference is experienced. This derived equation gives the solution model that DART_MCP algorithm analyses could use in the network environment to ensure its stability. The next section discusses various expressions used in determining the optimality of a typical SDN-based network and well as its stability through the network stability function (stability rendering equation function for network flow interference).

3.4 Deriving Optimality with DART_MCP Model

One major goal of proposing the DART_MCP analysis among the various available ones in the data centre networking environment is to enhance an optimal performance in terms of the rate of network stability when tailored user requirements are to be harnessed. This research observed that little or no reports have been carried out on the challenge of network stability regarding its routing algorithm. Therefore, the need for a model to reduce the number of packet failures and path redundancy in the course of packet transmission. The process of addressing this challenge brought about the conception of implementing a stable network via the routing algorithm model.

In this section, this research deploy a mechanism similar to the Ticket Based Probing TBP used in (Chen and Nahrstedt, 1999). However, deploy a network interference alleviation scheme to enhance network stability while meeting up users' requirements is needed. The network interference alleviation was built on the rendering equation to input network stability which was already established on a proven rendering equation in the published works of (Kajiya, 1986) and (Ng et al 2012). The equation holds for the test of optimality in the picture rendering graphic technique according to the published work and it was found useful as a standard to build on in the present research. The equation for this is expressed in equation 17 and, written thus:

$$L_o\,(x, w_o, \lambda, t) = L_e(X, w_o, \lambda, t) + \int_{\Omega}^{1} f_r\,(x, w_{i,} w_{o,} \lambda, t) L_i\big(x, w_{i,} \lambda, t\big)(w_i . n)\, dw_i \quad (3.17)$$

Where $\boldsymbol{L_o\,(x, w_o, \lambda, t)}$ equals to the total outgoing packets from various network hosts with bandwidth **"λ"** directed in a Poisson distribution manner of "w_o" through time **"t"** on a path distance of **"x"** away.

λ represents the bandwidth

$\boldsymbol{w_o}$ represents the Poisson distribution value

t represents the time for packet delivery

$\boldsymbol{L_e(X, w_o, \lambda, t)}$ represents outward packet distribution.

Ω represents units of packets transmitted through mean network *n* containing all possible values of w_o.

$\int_{\Omega}^{1} \dots .\, dw_o$ is an integral value over Ω

$\boldsymbol{f_r\big(x, w_{i,} w_{o,} \lambda, t\big)}$ is the bidirectional Poisson distribution of packets whose proportion varied from w_i to w_o over distance

x, time t, and bandwidth λ.

$\boldsymbol{w_i}$ is the inverse packet flow from controllers to hosts

n is the mean controller equidistance apart

$\boldsymbol{w_i . n}$ is the weakening interference factor of packets as they transverse the network.

Recall in equation 16 which is the general rendering equation that, having $L_e(X, w_o, \lambda, t)$ as the outward packet/flow distribution, for a case where the bandwidth is unity, through a time frame of 1 second. Therefore, the expression $L_e(X, w_o)$ is left.

Taking the conjugate surd of equation 3.15, the resulting expression is in equation 3.18

$$\textstyle\int_I(s) \quad = \quad V(x,y)\, \frac{\cos\sigma_i \cos\sigma_0}{\|x-y\|^2}\, dw_{y,} \tag{3.18}$$

where x and y represent the inverse cosine of σ from distances of the source to sink.
Substituting in equation 3.17, deriving the expression in equation 3.19 as:

$$L_{out}\,(x, w_o) = L_e(X, w_o) + \int_{\Omega}^{1} f_r\,(x, w_{i,} w_o) L_i(y, -w_i) V(x,y) \frac{\cos\sigma_i \cos\sigma_0}{\|x-y\|^2}\, dw_i \tag{3.19}$$

Which is the optimal expression to determine the optimal performance while deploying an adaptive rendering approach to address the Crosspoint queueing solution.

Hence, equation 3.19 expresses the solution to the DART_MCP algorithm model whose analysis enhances network stability through the elimination of network interference. Our idea is that once the solution to a single interference is determined, it is sufficient to address similar multi-objective optimization problems efficiently, then the summation of such a singular solution gives the aggregate of the larger network interference problem. In this way, the stability of such a network can be guaranteed based on the aggregate solution.

This research formulates the users' requests as a Poisson distribution requesting for network resources and also considered this as an optimization problem that needed to find a balance between the network resources and traffic flow requirement. Looking at a Poisson cluster process which is a motion-invariant request from a single user point process.

3.5 DART_MCP Algorithm.

Both the integrity and the transparency of the local area network user requirements become important in the network system. This necessitates that the overall network maintenance is inevitable through having a global view of packet routing in SDN. This global view, therefore, enhances the maintenance of the network system via the network controller. The significance of the global view of the network in alleviating flow interference in an SDN network forms the bedrock for the deployment of the ART algorithm for optimal network packet routing. The ART that was used in this work is primarily the deterministic approach which was built on a classical numerical value technique. The development of this approach first analysed the network topology by using Dijkstra's algorithm as well as the multi-criteria energy function which introduced the constraints that needed to be met for the user requirements. Thereafter, this research applied the DART_MCP algorithm to address various network user requirements. This algorithm is as illustrated in Algorithm I on the Appendix page.

The details about the variables used in the course of the algorithm development were all explained in the table that is shown in Table 3.1. The major flows that were considered were the asynchronous messages, stat-request messages, symmetric messages, and other flows within the network environment as stated in Table 3.1. The goal of the algorithm is to ensure that no two flows experience flow interference (either inter or intra) which is caused by the sharing of the same flow path and (or) network channel. The details of this implementation are further explained in Chapter 4.

Table 3.1: Meaning of the Notations in DART_MCP Algorithm

Symbol	Meanings
α_k	Number of *Controller-to-Switch* messages sent from a controller to S_k
β_k	Number of *Asynchronous* messages sent from S_k to a controller
γ_k	Number of *Symmetric* messages sent between a controller and S_k
σ_x	Number of *stat-requests* messages periodically sent from a controller to S_k
σ_y	Number of *flow_mod* messages sent from a controller to S_k
α_p	Number of *flow_removed* messages sent from S_k to a controller

β_p	Number of the packet in messages sent from S_k to a controller for local flows
γ'_k	Number of *packet_in* messages sent from S_k to a controller for global flows
$\in$	Number of *echo* messages sent between a controller and S_k
$E(u)$	Energy of the node
Z	The normal factor
$g*$	The local minimal energy
g(u)	Positive linear mark of node u
$g_k(u)$	The *k* constant parameter of the path from M and Z controllers
$d_k(u)$	The *k* constant parameter of the path from Z and L controllers
r(u)	The *k* constant parameter of the path between the intra and inter switches

3.6 Chapter Summary

This chapter discussed the type of problem addressed in the data centre which has to do with the network instability as a result of flow interference. The analysis explained that network flow interference occurs as a result of at least two network flows demanding or contending for the use of the same resources at the same time. These resources could be the memory buffer, CPU, or bandwidth. This chapter provided an analysis of the mathematical model for the defined problem which resulted in the development of the DART_MCP algorithm as a product of the standard rendering equation. This algorithm is designed to enhance an optimal performance in the rate of network stability using the user tailored requirement(s) as well as when no network quality is specified. In addition, it has the adaptive mechanism of using the solution in a single interference to adapt multiple subsequent interferences thereby adapting the traffic requests to their new state. This makes use of the active and reactive features which shall be tested in chapter 5. The algorithm, deployed as a mechanism operating as MOCQ approach would be found suitable in solving the identified problem as will be revealed in the subsequent chapters.

Chapter 4

SPECIFIED QUALITY APPROACH USING SINGLE PARAMETER

4.1 Introduction

The development of the Multi-Objective Optimisation Crosspoint Queue algorithm in Chapter 3 is one of the significant achievements of this thesis as it has found its unique usefulness in addressing the Network instability in DCN. This chapter addressed the challenge of prioritising network resources from traffic-flow interference. The network resources (e.g., bandwidth, Buffer) could warrant prioritising based on the application requirement at any instance of time. One of the significant capital outlay required for building an efficient and smooth operating data centre is to be able to manage the resources efficiently, especially the network bandwidth (Mahimkar 2011). In specified quality management of network resources, one of the network parameters could be a very important feature of the network that serves as the key enabling component, which facilitates an efficient and reliable node-to-node communication capability amidst huge generated volume of traffic (Cheng et al 2018, Liebeherr et al 2010).

In a DCN setup, several flows could compete for similar resources. For example, on the one hand, when multiple video streaming applications run concurrently, the resultant effect is a poor performance which could impair the quality of experience that would be received. On the other hand, just as users of mobile devices have been increasing over the years, so is the demand for better performance of these devices from the reservoir that is serving them. Therefore, developers and network operators need to ensure that the relevant network resources are stabilised at all times. Furthermore, while this stability is in place, the relative effect of the achieved stability on both the provider and the end user should be balanced in the same manner in terms of the cost at both ends. The goal of this chapter, therefore, is to ensure that high-quality service is available for various end-users while not jeopardizing the cost on the path of the service provider. This solution is very significant, especially to the business-oriented service providers with cost management in mind to ensure a guaranteed overall cost balance. The solution in this chapter utilised the Transmission

Control Protocol (TCP) to optimise the requirements for the network setup so that user fairness alongside a guaranteed degree of stability is established in the network (Abuteir et al 2016, Hammadi & Mhamdi, 2014).

In this chapter, Section 4.2 describes the bandwidth maximizing stability framework based on the established SDN framework. Section 4.3 presents the various categorization of the traffic flows in a typical home-based network for consideration purposes, while Section 4.4 describes a typical SDN-based network analysis for stability requests. The experimental evaluation is presented in Section 4.5 and provides various performance analysis measures while section 4.6 contains the discussion of the performance results. The chapter concludes with a summary in Section 4.7.
NOTE: SQA: Specified Quality Approach

4.2 Bandwidth Maximizing Stability Framework

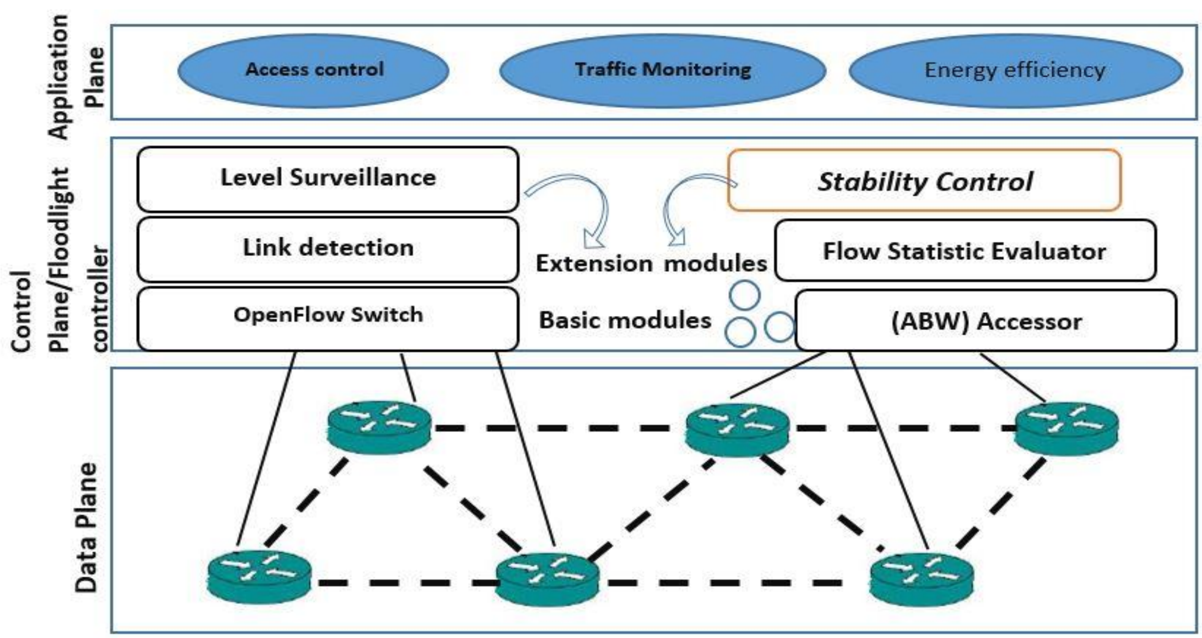


Figure 4.1: Framework for Network Stability based on maximizing ABW.

The diagram in Figure 4.1 shows the layers of a typical SDN paradigm, having the control plane separated from the data plane with the application plane placed on top to allow for the deployment of varying applications. The major modification on the framework was carried out in the control section to include features like stability control mechanism, flow statistic evaluator, Available BandWidth (ABW) Accessor, level surveillance, link detection as well as manipulative modules (Extension and Basic modules) for network control. The data plane consists of various network elements such as switches and routers which are meant to forward packets within the network. The application plane also enhances the deployment of new applications in the network, thereby creating an opportunity for network upgrades. The flow statistics of the network were collected by the level surveillance which was processed and fed into the control section for informed decisions. The link detection section compares the various multiple links that are available to determine the ones with less congestion possibility based on bandwidth availability. Some counters determine the number of bytes being transferred on the links so that the flow statistic evaluator takes the timestamps of replies to messages that are generated.

Moreover, the flow statistics data that are generated from level surveillance are fed into the stability control module to enhance network stability through the control plane. Let us consider a typical OpenFlow switch specification that supports both per-port counters and per-flow counters. The per-port counter enables the transfer of flow statistics to the controller (floodlight) when the switch sends a request. This typically describes the level of packet counts that is being transferred, aided by flow statistic evaluator, although this does not yield an accurate result of the network (Takano et al 2016). The OpenFlow specification cannot detect accurately the timestamp for measuring the response to the messages, thus, before the messages arrive, a new level would have been attained. Hence, the level surveillance determines the link level by finding the difference between the sending rate of the packets and the receiving rate at the destination and quickly re-run the differences to maintain the least possible result that yield the most recent solution.

Mathematically, considering a network whose parameters were as follows: Let DY_i denotes the number of headers packet size currently flowing for the *ith* data flows. EY_i represents the number of header size of the *ith* data flow yet unsent while the P_i denotes the payload size. Also, if *h* is the number of total data flow in and out of the network and *j* represents the number of unsent data flow. Thus, representing the bandwidth available (BA) in the network as the summation of the total from the used bandwidth as stated in equation 4.1 thus:

$$BA = \frac{\sum_{i}^{h}(DY_i - EY_i)}{\sum_{i=1}^{j}(DY_i + Pa_i)} \quad (4.1)$$

For similar data flows running using the same payload size and network protocols on other links, re-expressing the function in equation 4.2 as:

$$BA = \frac{\sum_{i=1}^{h}(DY - EY)}{\sum_{i=1}^{j}(DY + Pa)}$$

$$= \frac{h(DY - EY)}{j(DY + Pa)} \quad (4.2)$$

Therefore, the ratio that is derived from *h* to *j* has a significant influence on the available bandwidth of the system each time. When the number of data flowing is more voluminous than the unsent data, there is a high probability that the number of bandwidth available is very meagre and vice versa. Utilizing the correlation coefficient expression to determine a relative expression for fast bandwidth availability derivation. This approach considers both the sending and the receiving rates of packets and linearly calculates a faster projection based on the Meta-data from the improved calibrated Pathload Algorithm (PA) (Takano et al 2016). Hence, this study determines the transition of ***correlation coefficient*** *Cor(k)* between the sending rate and packet loss rate for the network. Thus, taking the packet loss rate for *Cli_1* through to *Cli_2* to be measured for sending node *Ns* and the receiving node *Nr* as shown in equation 4.3, having:

$$Cor(k) = \frac{\sum_{i=k}^{n}(N_s - \bar{N})\left(Cli_{\_1} - \overline{Cli}\right)}{\sqrt{\sum_{i=k}^{n}(N_s - \bar{N})^2}\ \sqrt{\sum_{i=k}^{n}\left(Cli_{\_1} - \overline{Cli}\right)^2}} \quad (4.3)$$

The mean of the sending rate is given by $\bar{N}$ when packets were sent from *Ns to Nr,* while $\overline{Cli}$ referred to the mean of the packet loss experienced between *Cli_1* to *Cli_2.* A strong value of the $Cor(k)$ results in a value that is very close to unity or the negative unity value which signifies that the stability of the network is high. Any value that is close to 0 depicts a weak correlation hence a low stability network system. In a situation where the network *path/link* is considered, the usual trade-off for ABW measurements is dependent on achieving a more accurate ABW value at the expense of the timestamp needed to complete the process. Using a typical 1000MB full duplex capacity, the bandwidth utilization is calculated as (TxLoad + RxLoad)/510 where Tx is the rate on sending the packets and Rx is that at the receiving end. Assuming there is communication from an end-to-end path along which there are *i* numbers of sequential links, the maximum capacity of the path is always the minimum capacity of all the available links. Hence, we describe the minimum path for the links with equation 4.4 as:

$$P_{link} = min(p_1, p_2, p_3, \dots p_i) \quad (4.4)$$

where $p_1$represents the available bandwidth of link 1 and the P_{link} refers to the end-to-end aggregate bandwidth of a given path. Since the pathload induces a fleet of probe trains that flood the network, the use of iterations to enhance the accuracy of this solution is needed. However, the calibrated PA stipulates the use of only one train of probe samples and the bit rate for each of the subsequent iterations were shown in equation 4.5 as

$$Q_n = \bar{Q} + \frac{|\bar{R} - \bar{Q}|}{2} \quad (4.5)$$

where Q represent the mean of the inter-packet interval from the sender of the recent packet on the iteration; R refers to the mean of inter-packet interval on the reception of the packet while Q_n represents the inter-packet on the sender node for the subsequent iteration. The relationship between these values is significant in that the mechanism compares and find the average inter-packet interval time differences between the destination node and the sending node. Our hybrid concept (improved calibrated pathload and correlation coefficient approach) comes in here by using the method of correlation coefficient to provide a faster solution in comparison to the network specified threshold. Based on the existing approach, the measurement of the traffic is faster when the cross-traffic measurement is in a static state. However, as soon as the situation becomes unstable, the accuracy of the approach usually downgrades, or else there is a need for the deployment of the faster correlation coefficient.

The correlation coefficient approach stipulates that when measuring the ABW in a network that is affected by the network channel and flow path interferences, the narration from chapter 3 proposed that deploying an approach that quickly considers the packet losses within the path/link. For example, given that the available bandwidth in a network is greater than the packet sending rate, there are still some arbitrary packets that are lost although could be very infinitesimal. Conversely, the rate of packet sending could be greater than the available bandwidth that the network provides, yet the relationship is that there is a linear increase in the rate of packet losses as the rate of packet sending increases. Therefore, with the use of the hybrid approach here, the study determined the amount of ABW present within a network at every instance of timeframe for the optimization of the traffic flows, considering the impact of this optimization on both the service provider and end-users in a DCN. This is also contained in the flow statistics which forms part of the module information to be injected into the floodlight controller.

4.3 Categorization of Flows

Based on the various kinds of flows in a data centre, this research categorised the flows into three major groups. These groups are the Giant flows, Larger flows, and Smaller flows. Most of the giant and larger flows are often the real-time flows that draw more bandwidth than the smaller flows which are majorly the text-based flows. In this study, the flows were divided among the ranges by using $l < 100kb$ as smaller flows, $l > 100kb$ as larger flows and $l > 1Mb$ as giant flows which were similar to the approach in Chen et al (2015). However, the challenge of flow interference set in when by the demands for the bandwidth, the smaller flows find it difficult to get the request through due to the interference of the giant and larger flows. Such smaller flows usually have to wait in the queue until the giant flow completes the traffic flow transmission. The task addressed here is to ensure that the end-users are not dissatisfied, and the providers do not incur an excessive cost in the network environment.

4.4 SDN-based Network Analysis for Stability Request.

The subsections below highlighted the analysis of traffic flow from the provider to the user.

4.4.1 Infrastructure.

This research considered a service provider with C numbers of a set of controllers to be used for a DCN which are $C = \{a_1, a_2, \ldots, a_c\}$. Each of the controllers was an element of the total number $a_j \in C$ is assigned with a fixed bandwidth capacity of B_j. If the number of users within the network is denoted by $U = \{s_1, s_2, \ldots, s_u\}$ and the variable N represents the set of application instances currently offered by the service providers on the network being $\{p_1, p_2, \ldots, p_n\}$. Let the variable b_i be the value of the bandwidth that will be required to address a service request $p_1 \in N$. Varying the setup of the service providers applications in the data centre are usually considered to provide several samples of each application such that when an application is not available, it can be fetched elsewhere within the network. Hence, each copy of these applications that was hosted by a_j is defined by $q_j = (q_{1j}, q_{2j}, \ldots, q_{Nj})$, in which $q_{i,j}$ represents the binary pair variable that indicates if p_i is found on a_j. Then, having the expression in equation 4.6 thus:

$$q_{i,j} = \begin{cases} 1 & if\ p_i\ is\ found\ on\ a_j \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

Having all $p_1 \in N$, assuming that $s_u \in U$ can make a single request at one particular time. Therefore, the end user's request can be represented in a kind of matrix form by $t_N = [t_{k,i}]_{K \times N}$, the value $t_{k,i}$ representing the s_u which represents the end-user requesting the service p_1. The expression for the analysis is given in equation 4.7 as:

$$t_{k,i} = \begin{cases} 1 & if\ s_u\ is\ found\ on\ p_i \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

If $r_{i,j}^k$ denote that a_j receives the request that s_u made to p_i. Thus, representing this with the expression in equation 4.8 as:

$$r_{i,j}^k = \begin{cases} 1 & if\ t_{k,i} q_{i,j} \neq 0\ and\ s_u\ receives \\ & the\ request\ from\ s_u\ to\ p_i, \\ 0 & \text{otherwise.} \end{cases} \tag{4.8}$$

If the available bandwidth that is derived from the previous section is denoted by ABW, then $r_{i,j}^k$ which happens to be the main task of this analysis should be enabled to achieve a successful and optimal traffic flow through the controller in such a manner that both the end-user and the provider of the service have their interests guaranteed. This research deployed the use of the Kalai-Smorodinsky (KS) solution for N numbers of players in a win-win solution to a utility-demanding network scenario such that the benefits are maximized at both ends of the provider and users (Kalai

1985, Zehavi and Leshem 2009). Solving the derived optimization would result in achieving a higher level of satisfaction as well as utility and service fairness for both players.

4.4.2 Service Provider and End-Users

This research argued that the stability of an SDN enabled networking is such that it is beneficial to both the provider of the infrastructure as well as the end-user. The analysis below depicted the terms for defining the benefits for both the service providers as well as the end-users.

4.4.2.1 Benefits of Service Provider

This research considered a typical scalable data centre with several *C* network controllers. The network owner usually invests some amount of costs in installing the controllers hence the need to balance the load among the controllers as well as higher utilization of the network bandwidth at each instance. Proper load balancing and efficient utilization would not only result in a stable network but also enhance higher throughput gain by the traffic flows. During a heavy traffic flow especially in a larger network setup, overloading the controller at any instance would have a drastic and significant effect on the amount of throughput that is achieved, and the response time will be drastically impaired. This research proposed an expression S_j in equation 4.9 to represent the consumption of bandwidth by the traffic flows based on its utilization of each controller in the data centre a_j as:

$$S_j = \sum_{k=1}^{U}\sum_{i=1}^{N}\frac{t_{k,i}q_{i,j}r_{i,j}^{k}b_i}{B_j} \qquad (4.9)$$

The KS bargaining solution expresses the provider as a social utility benefit. The KS describes the parameters as having a *C* number of controllers as the players which receive several numbers of requests as *U* from the various end-users, regarded as the commodities. The KS enables every player to maximize the utilization of the controller bandwidth S_j within the utility function in equation 4.9. The maximization process further enhances the maintenance of traffic flow balance amidst the available controllers even at higher bandwidth utilization in a win-win approach. Therefore, expressing the benefit under this approach in equation 4.10 as:

$$\prod_{j=1}^{C} \sum_{k=1}^{U} \sum_{i=1}^{N} \frac{t_{k,i} q_{i,j} r_{i,j}^{k} b_i}{B_j} = \prod_{j=1}^{C} S_j \qquad (4.10)$$

4.4.2.2 Benefits of End-User

The stability of the network also covers the perception of the end-user to ensure that optimal satisfaction is achieved. The pivotal concept for the end-user satisfaction is a measure of fairness attained and the quality of user' experience that is achieved, while the user experience is undoubtedly reflected mostly by the response time for the flows to reach their destinations; some other factors are integral in defining the user experience which could also be a function of the cost of the service, power cost, etc.

This research represented the end users' request as a series of traffic flows in a range of queues arriving at the same time being $Que_{i,j}$. If the average arrival rate is represented by $\theta_{i,j}$ for each of the service request instances of a_j, then, the inter-arrival time of the flows is represented by $\alpha_{i,j}$ with the average service time being denoted by $\emptyset_{i,j}$. This research assume that the current traffic flow is attended to by the same controller, the traffic offered concerning the fraction of the time its flow processing is represented as $\beta_{i,j} = \theta_{i,j} \emptyset_{i,j}$. By analysis, we intend to determine an expression for the mean response time, in terms of the Average Waiting Time (AWT) and Average Service Time (AST). The fairness is always fully expressed with the use of the Coefficient of Variance (COV) thus, we represented the squared COV of the quantities (AWT and AST) by $\sigma^2_{\emptyset_{i,j}}$and $\sigma^2_{\theta_{i,j}}$ respectively. Modelling the controller as M/M/1 queue (Cheng et al 2015, Wang et al 2016c), thus, the response time expression is written in equation 4.11as:

$$\delta_{i,j} = \emptyset_{i,j} + \emptyset_{i,j} \frac{\beta_{i,j}}{1-\beta_{i,j}} \left(\frac{\sigma^2_{\theta_{i,j}} + \sigma^2_{\emptyset_{i,j}}}{2} \right) \quad (4.11)$$

The expression for the cost of the end-users is represented by $\Delta^k_{i,j}$ and the inherent cost of S_u requesting p_i from a_j. Using a user experience ranging from between 0 and 1; the degree of experience of S_u through requesting for p_i from a_j given by $Z^k_{i,j}$ is expressed in equation 4.12 as:

$$Z^k_{i,j} = \exp\left(-\delta_{i,j} - \Delta^k_{i,j}\right) \quad (4.12)$$

Therefore, the mean user experience of the end-user S_u is given by N_k and is derived as expressed in equation 4.13 thus:

$$N_k = \sum_{j=1}^{C} \sum_{i=1}^{N} t_{k,i} q_{i,j} r^k_{i,j} Z^k_{i,j} \ / \ \sum_{i=1}^{N} t_{k,i} \quad (4.13)$$

Summarily, the rational players will enable the maximal use of the controllers for optimal satisfaction or utility which will be revealed by the user experience N_k. Deriving the end-user benefits from the KS product in the KS bargaining solution, this research achieves a fairness result among the participating end users along with the optimal user experience which can be defined as expressed in equation 4.14 as:

$$\prod_{k=1}^{U} \sum_{j=1}^{C} \sum_{i=1}^{N} t_{k,i} q_{i,j} r^k_{i,j} Z^k_{i,j} \ / \ \sum_{i=1}^{N} t_{k,i} = \prod_{k=1}^{U} N_k \quad (4.14)$$

4.4.3 Formulating the Stability Problem

To formulate the stability problem as an optimization problem (MOCQ) in bandwidth demand critical flow that took into cognizance the network flow interference, it is compulsory, as earlier highlighted as one of our objectives, to monitor the effects on the service provider and the end-users. This research, therefore, desire to maximize the expressions in equation 4.15 thus:

$$Maximize: \quad \begin{pmatrix} \prod_{j=1}^{C} S_j \\ \prod_{k=1}^{U} N_k \end{pmatrix} \tag{4.15}$$

Subject to:

$$\sum_{j=1}^{C} r_{i,j}^{k} = 1, \forall_k, \forall_{i,} \tag{4.16}$$

$$\sum_{k=1}^{C} \sum_{i=1}^{N} t_{k,i} q_{i,j} r_{i,j}^{k} b_i \leq B_j, \forall_{j,} \tag{4.17}$$

Where the expression in equation 4.16 ensures that only a single flow can be assigned to a controller at once within the data centre, the second constraint in equation 4.17 restricts the possibility of exceeding the controller capacity constraint for the bandwidth. Summarily, since our goal is to ensure that while alleviating the possibility of flow interference occurring in the course of network traffic flow, this research ensure that the process do not just optimize the problem at the expense of either the producer or the end-user. The product of several instances of applications made available from the service provider and the various requests that were made were not considered because they have no impact that could be seen on the objectives that are considered. Therefore, equation 4.16 stands to give a constraint to benefit the end-user while equation 4.17 represents the constraint for service provider benefit. This research tries to derive the multi-objective optimization problem for the two objectives (which takes care of the stability of the concerned two parties in terms of service provider and end user' benefits) that was stated in equations 4.15 and 4.17 through taking the logarithm of the highlighted objectives as shown in equation 4.18 thus:

$$Maximize \quad \begin{pmatrix} \sum_{j=1}^{C} In \sum_{k=1}^{U} \sum_{i=1}^{N} \frac{t_{k,i} q_{i,j} r_{i,j}^{k} b_i}{B_j} \\ \sum_{k=1}^{U} In \sum_{j=1}^{C} \sum_{i=1}^{N} t_{k,i} q_{i,j} r_{i,j}^{k} Z_{i,j}^{k} \ / \ \sum_{i=1}^{N} t_{k,i} \end{pmatrix} \tag{4.18}$$

By deploying the new weight sum model (Miljković et al 2017) to resolve the multi-objective problem in equation 4.18. It results in a simpler single objective optimization problem such that the summation of any weighted factor of any of the objectives would ensure that the required trade-off is achieved either to the end-user or the service provider. In this case, this research represented the weighted benefits of both parties by γ_1 and γ_2 such that the sum of both equals 1. Therefore, arriving at a single objective problem in equation 4.19 thus:

$$Maximize \quad Q(u), \tag{4.19}$$

Where γ_1 and γ_2 represent the benefits of the end user and service providers respectively. Using the scalarisation method of the rank-sum weight, the multi-objective functions are made into single solutions by weights. This research derived a theorem that efficiently showed that the solution to the single optimization would effectively address the multi-objective optimization as well as it is shown in the theorem.

Theorem 1. The scalarisation of a multi-objective optimization problem derives its part solution by effectively proffering solution to its single-objective optimization derivative(s).

Given that the import from our earlier expressions to have equation 4.20 thus:

$$\aleph_1 (u) = \sum_{k=1}^{U} In \sum_{j=1}^{C} \sum_{i=1}^{N} t_{k,i} q_{i,j} r_{i,j}^{k} Z_{i,j}^{k} \ / \ \sum_{i=1}^{N} t_{k,i} \tag{4.20}$$

Likewise equating the expression in equation 4.21 as:

$$\aleph_2 (u) = \sum_{j=1}^{C} In \sum_{k=1}^{U} \sum_{i=1}^{N} \frac{t_{k,i} q_{i,j} r_{i,j}^{k} b_i}{B_j} \quad (4.21)$$

This research assigned that *u* represent a matrix $r_{i,j}^{k}$ which covers a region of space that is occupied by u and is denoted by a variable Y. This research assigned Y to be a set of *u* which takes into considerations all the objective constraints that are covered by equations 4.16 and 4.17 above. Therefore, there exists an optimal solution for a single-objective optimization problem in equation 4.19 that is represented by *u** which is not effectively solving the existing multi-objective optimization problem that was derived in equation 36. This can be otherwise expressed in mathematical notation as *u** ∈ Y such that the following expression unfolds in equation 4.22 as:

$$\begin{aligned} \aleph_1 (u^*) &> \aleph_1 (u^*), \\ \aleph_2 (u^*) &> \aleph_2 (u^*). \end{aligned} \quad (4.22)$$

Summarily, arriving at an expression in equation 4.23 as

$$\gamma_1 \aleph_1 (u^*) + \gamma_2 \aleph_2 (u^*) > \gamma_1 \aleph_1 (u^*) + \gamma_2 \aleph_2 (u^*) \quad (4.23)$$

The expression in equation 41 which is expressed in terms of *u** depicted that the solution arrived at for the single objective optimization problem is not an efficient solution although theorem 1 is proven. After the analysis of these setups, deploying the MOCQ to implement the constraints and evaluate the performances of both the service provider and end users.

4.5 Experimental Evaluation

This study deployed an Objective Modular Network Testbed in C++ (OMNeT++) network simulator aided by real word traces to ascertain the performance of the ART algorithm. This section simulated a provider which enabled the operation of four data centres with each having 50 controllers. Four different instances of application were hosted on each data centre and the

Wikipedia request traces which happened to be a real word trace was used to represent the network traffic arising from the requests(Gebrehiwot 2017).

This research first tested the rate of congestions for the traffic flows through the request for the running applications. This was ranged for 50 hours which approximately ran for two days. Although the data sets have no end-user details, dividing the whole traffic flow among the users such that a kind of normal distribution is attained. This research set the controllers to have a fixed bandwidth capacity with the running applications instances consuming relatively the same amount of bandwidth. This research first determined the benefit of the end-users and service providers in a situation where both are set to be equal. The initial parameters for this experiment included four controllers with just four users while keeping all other parameters at default state. The cost can be varied in practical response to the changing demands of the end-users. The predictions of the requests for over 50 hours (~2 days) are depicted in Figure 8.

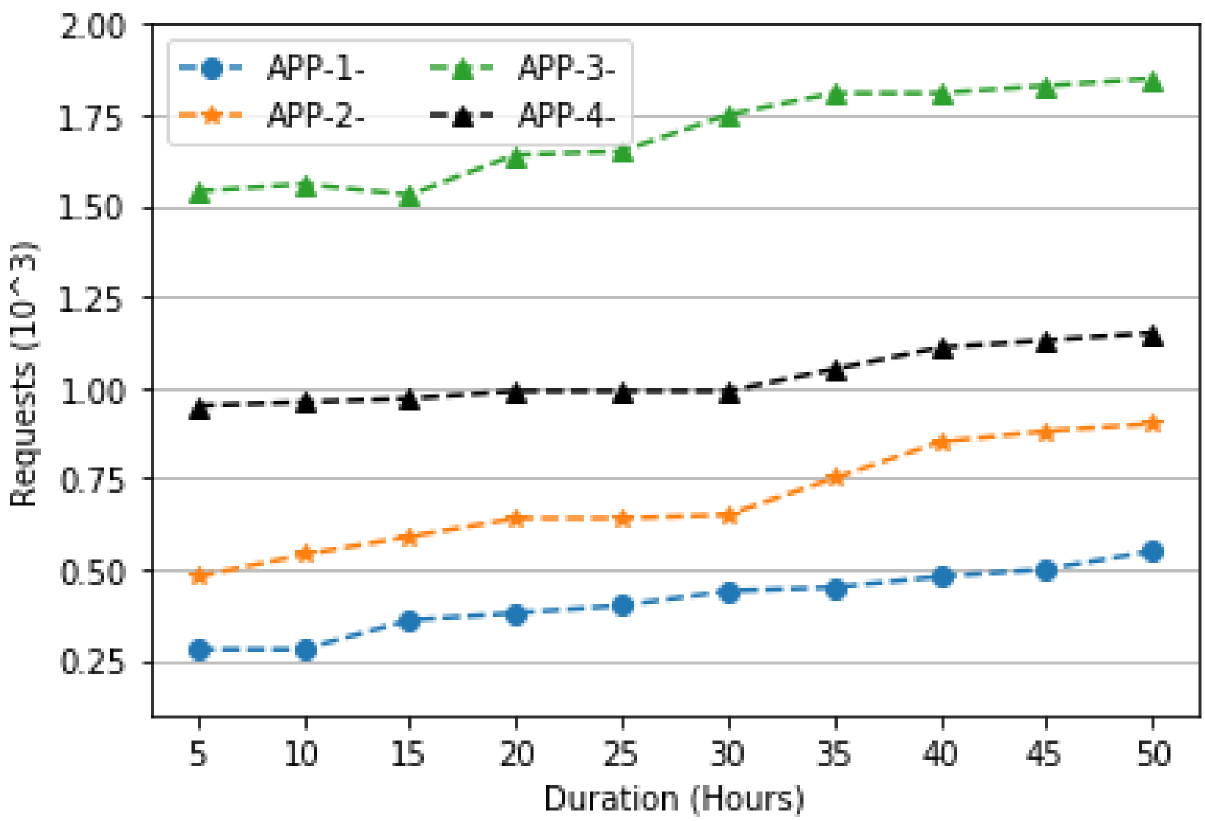


Figure 4.2: Request traces for the applications on the Wiki platform

The requests that were made on the Wikipedia platform vary from application to application and a simple check-up showed that the platform was working efficiently enough to test the performance of the adaptive rendering algorithm that was proposed earlier. Figure 4.2 depicted that the requests were evenly responded to, except in some locations along the line where they dropped a little. The droppings at these locations can be accounted for by several issues which could range from congestion as a result of flow interference, bad network supply, unavailability of the applications itself, or even fewer demands for the respective users. Whatever the case, this research need to run some experiments that will give us appropriate explanations of Figure 8 as well as tell us more of the effect in terms of benefits to the service provider and the end users.

4.5.1 Performance Analysis of the Algorithm

The metrics in the following subsections were used to test the performance of the proposed algorithm.

4.5.1.1 Bandwidth Utilization

Testing the performance of our proposed MOCQ algorithm. The performance test intend to find out the average bandwidth utilization over 50 hours when the capacity of each controller was set to 1000 units (meaning that the set B_j to be equal to 1000 for all the controller j's). Assuming the interface has a capacity of 100MB full duplex, the bandwidth utilization is calculated as (TxLoad + RxLoad)/510. And for this experiment, with 1000MB, similar calculation was made to determine the utilization of the bandwidth utilization. The first inference was to determine the rate of consumption of the bandwidth on the arrival of several requests. The diagram in Figure 4.3 depicted the performance of the algorithm in addressing the traffic flows in the network. Some similarities exist in the behaviour of the average requests sent out which has been earlier plotted and the bandwidth utilization behaviour. One important deduction from the experiment is the maximization of the bandwidth cost. This inference from the experiment is that the rate of average bandwidth consumption barely exceeds 1.80, thus proving beneficial to the service provider in terms of bandwidth maximization.

The service providers do not unnecessarily incur more expenses and costs on extending insufficient bandwidth. The experiment in this section is therefore very useful for the service provider in identifying the maximum technical know-how of maintaining a fairly stable network provisioning, considering the limited size of bandwidth at hand. In the same vein, the second experiment gives more information regarding the corresponding behaviour of the end user with the current provision of the service provider. It must be noted, however, that before the service provider's relative stability is achieved, the end-user demands have already been considered for optimal satisfaction of their requirements. Furthermore, Figure 4.3 showed us more information about the optimization of the bandwidth for network stability on the part of user experience.

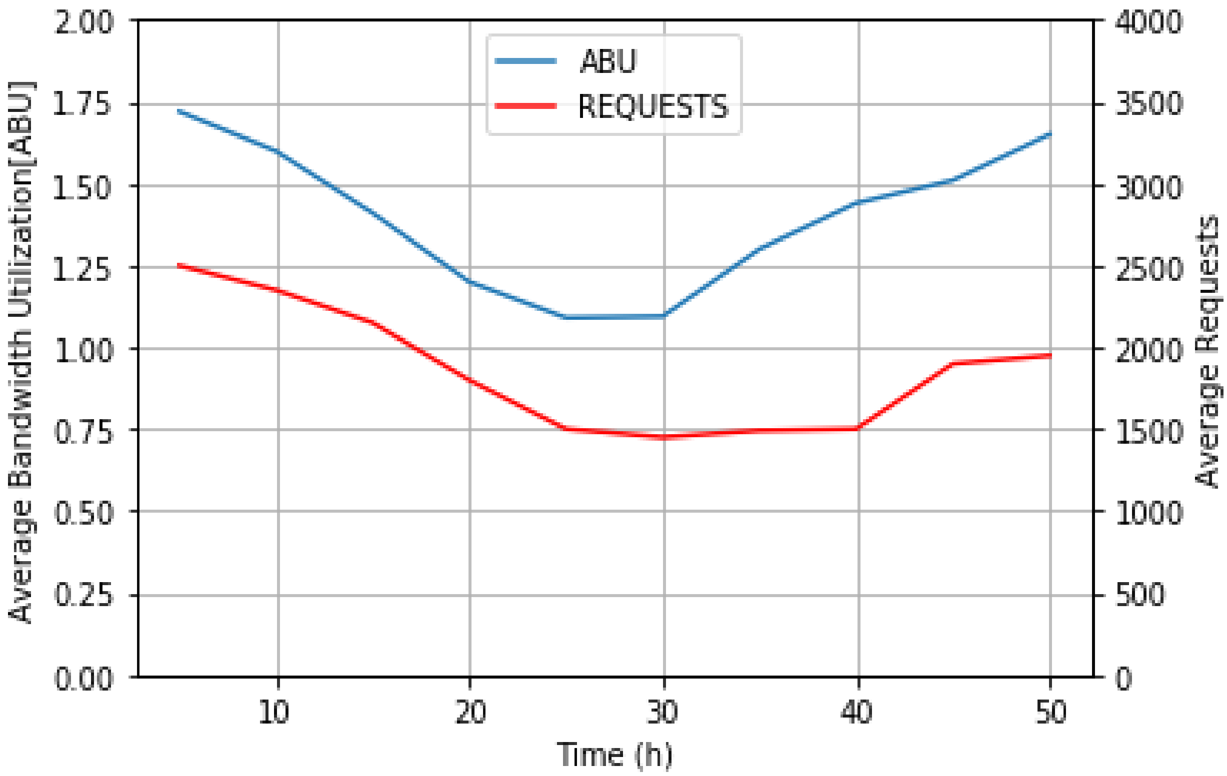


Figure 4.3: Network Average Bandwidth Utilization

4.5.1.2 Average User Experience (AUE)

Figure 4.4 depicted a relatively stable experience over a range of 0.50 irrespective of the fluctuations in the average requests that were incurred. This is expressed as the time on the task request that the user experienced. Time on task indicates how long it takes a particular user to complete the specific task from start to finish. Thus expressing user experience as (user1 + user2

+ userN time ….) /total number of users. The red line which showed the least fluctuating requests at around 2200 and the highest at almost 3500 requests was optimized to maintain a stable average user experience of 0.50. The figure also depicted the impact of time function on the network when it was almost tending toward 50 hours. A tilt was experienced and this could be attributed to the accumulated network flows which were probably meant to initiate the attainment of a new stability level for the network user experience. Thus, the AUE is maintained under a stable rate below 0.75 stability level.

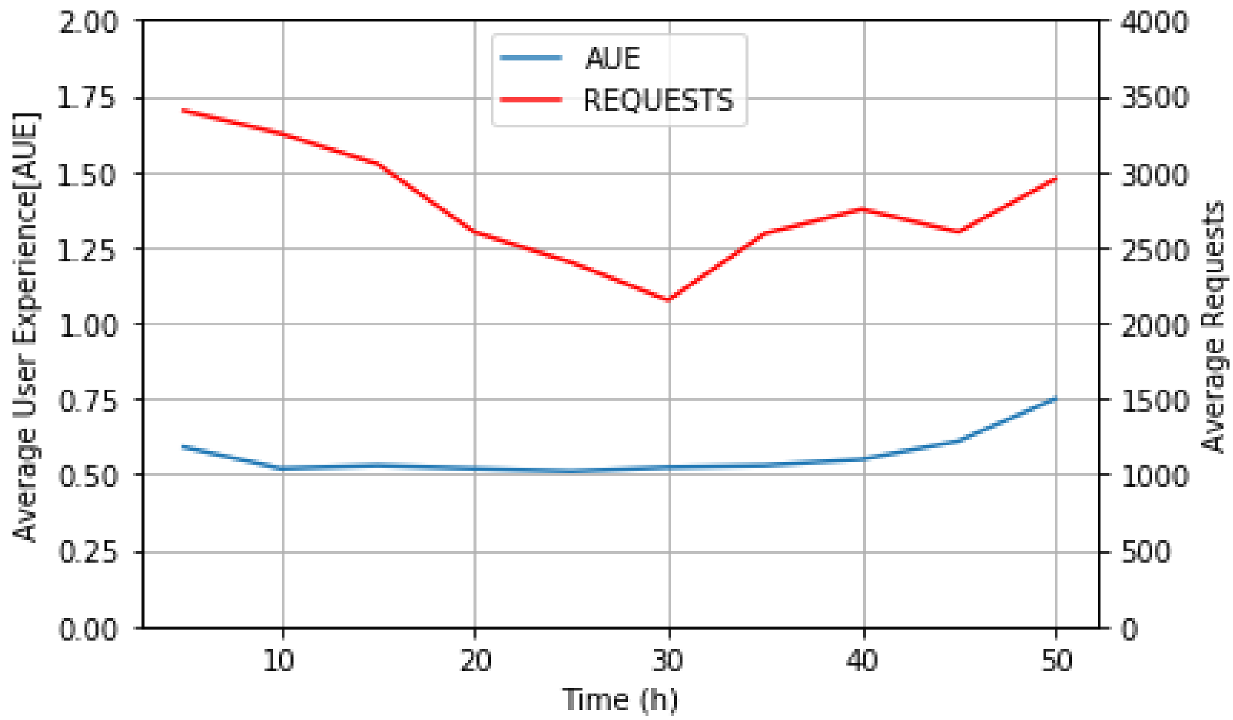


Figure 4.4: Network Average User Experience

4.5.1.3 Cumulative Distribution Function (CDF)

This research desired to derive the cumulative distribution function per number of request responses granted to go through the network and the result is shown in Figure 4.5. The OMNeT++ has an in-built function that determines the CDF of a parameter over a variable being processed as *get CDF (x)* where in the case of this experiment, the x is the response time. The function returns the cumulative distribution function for the one-parameter distribution function. The processing capacity of the controllers was maintained at B_j = 1000 and this research examined the rate of response time that was derived in the course of the experiment. This experiment critically analyse that the rate of request responses was approximately 98% below 1000 ms in performance, depicting that just a little around 2% took more than 1000 ms to have feedback. The extracts from Figure 4.5 showed that the bulk of the traffic flows was responded to at an average of 600 ms in terms of response time. One of the goals of our proposed work is to optimised the network systems in such a manner that a guaranteed level of stability is attained in the network which favours both the service provider as well as the end user. Therefore, for industrial use, the network stability that is attained on average for this particular network setup is maintained at 600 ms when fixed at a controller bandwidth capacity of 1000 unit setting.

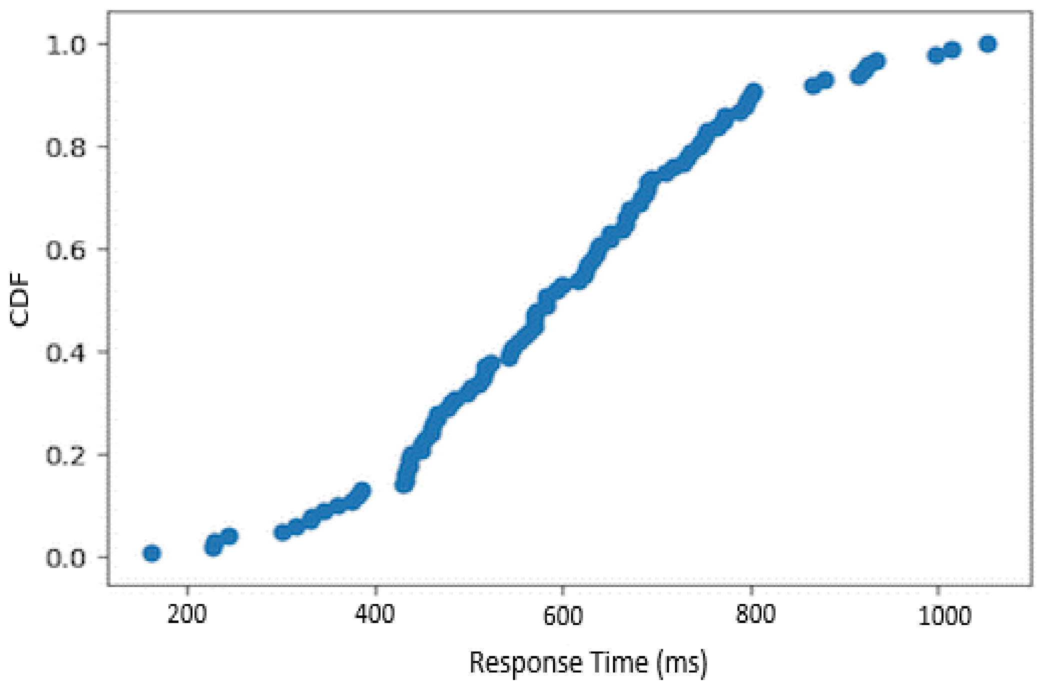


Figure 4.5: The Cumulative Distribution Function at Bj equals 1000 units

4.5.1.4 Reduced Controller Bandwidth Capacity Comparison

This study intended to see the effect of the reduction of the bandwidth capacity B_j to 800 while comparing it with the performance of the initial 1000 units capacity which was used in the previous sections. The initial intention was to see the impact of the reduction of the bandwidth capacity on the amount of bandwidth utilization and thereafter try to determine if there was the likelihood of having a fluctuation or probable changes in the level of stability that the end-user experienced. While keeping the capacity at 800, the recorded performance was achieved as shown in Figure 4.6 for the Average Bandwidth Utilization (ABU). This study discovered that the consumption was increased a bit above the earlier experiment even though the pattern of the utilization that was recorded was similar to earlier ones. These results prove to be correct in that when there is a reduction in bandwidth size, the utilization might need to shoot up for the number of flows that have been deprived of the network resources. This confirms the stability of the network more and can be used to make several probabilities regarding any specific SDN based network setup environment.

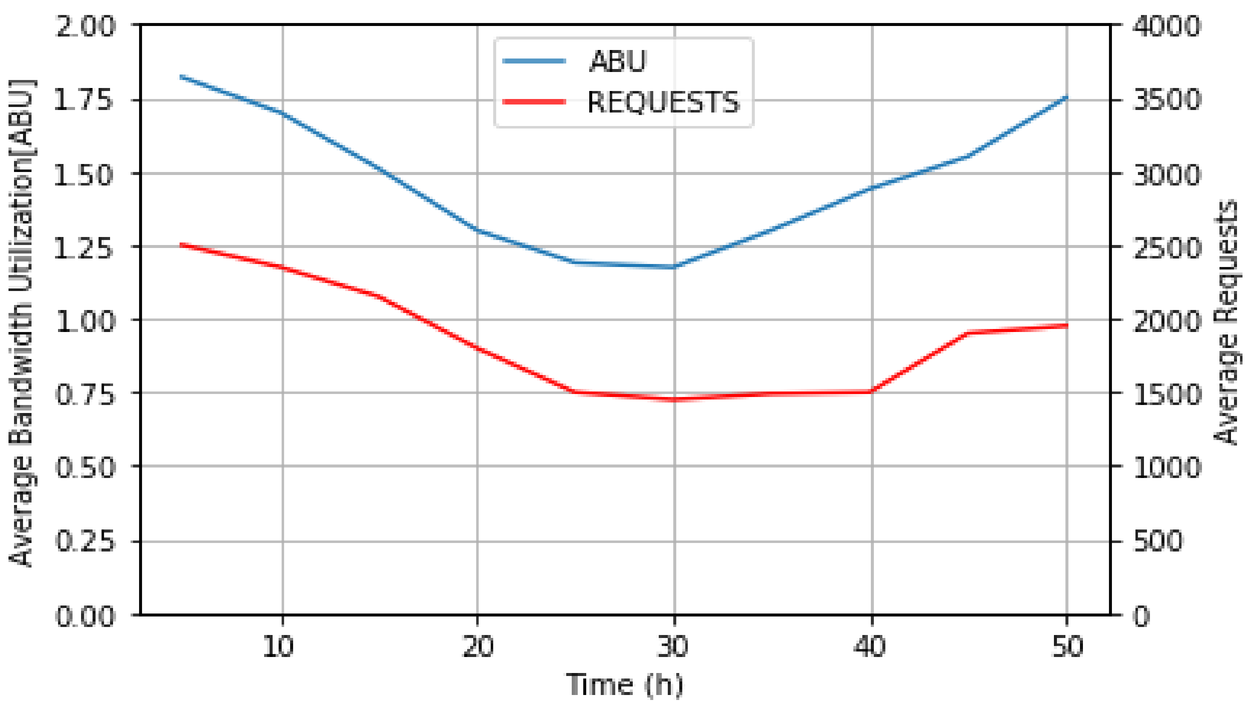


Figure 4.6: Network Average Bandwidth Utilization for Bj equals 800

In the same vein, this research was interested in the performance, in terms of the experience of the end user in the course of maximizing the available bandwidth consumption from the service provider. The result proved that there was relative stability in the end-user experience despite the fluctuating average request accrued. This implied that the service provisioning was not affected in any way by the reduction in the bandwidth that is made available to the traffic flow requests which were sent. This further proved the stability of the network provisioning system that is optimized using the MOCQ approach. The implications and the significance of the result at this point is that the service provider could earn some more profits by not incurring more expenses in acquiring more bandwidth on each of the controllers. This is depicted in the diagram in Figure 4.7. The maximization of the bandwidth that was set on the controllers enhances profit realization with relatively no negative effect on the output end-user experience.

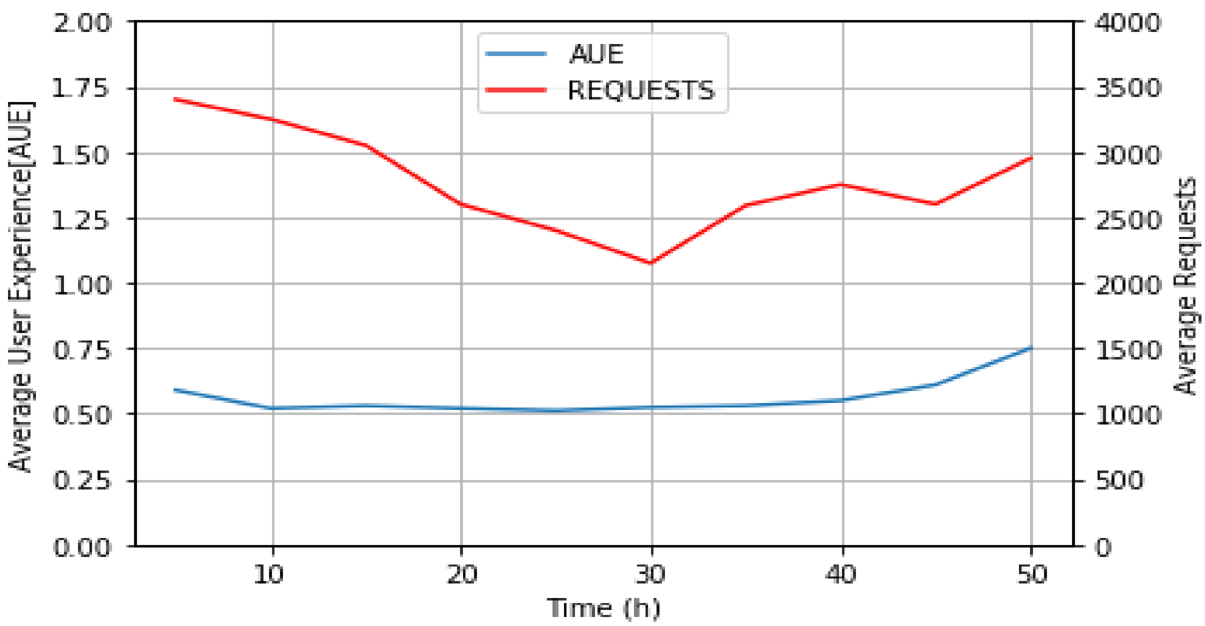


Figure 4: Network Average User Experience for Bj equals 800.

4.5.1.5 Varying Controller Capacities: The CDF performance

The processing capacity of the controllers was dropped from the initial value of 1000 to 800 and subsequently to 600 to evaluate the effect of constant traffic flows on the average response time taken for the flow setup completion. The Cumulative Distribution Function (CDF) usually depicts the level of distribution of the results especially in places with the highest frequency and in our case, the number of responses was seen to intersect and disperse along an almost similar axis though not the same. Three different validations were determined from this simple experiment. These are:

Firstly, the average overall response time (ms) for traffic flow in the system is determined at around 600 ms, thus the service provider and the end user could derive an optimal benefit.

Secondly, Figure 4.8 showed the possibility of the service provider to vary his cost of rendering the services without a drastic effect on the output on the end user part thereby maximizing the providers' cost.

Thirdly, both parties benefit from the service being rendered with the major interest protected.

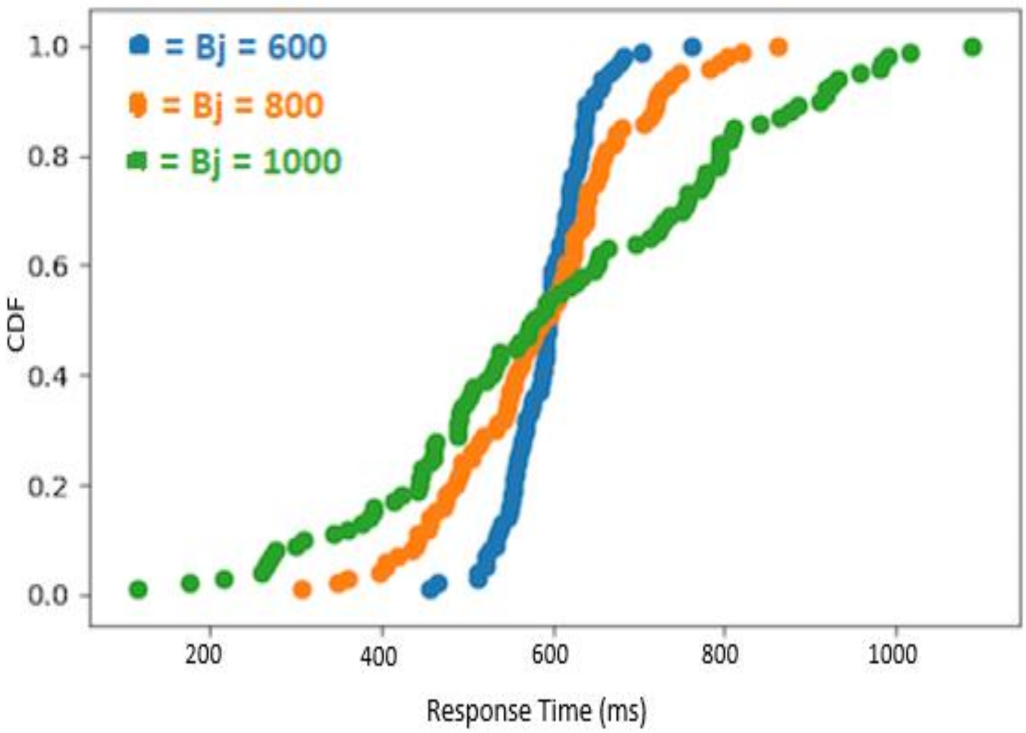


Figure 4.8: The Cumulative Distribution Functions at various Bjs' units

4.6 Discussion of Results

The specified quality approach in this chapter addressed the network bandwidth as one of the vital network resources that must be maximized on any network platform (data centre networks). The goal is that the flows from the three highlighted categories (giant, long and small flows) must be optimized in such a way that one of the flows is not incurring deterrent delay for the other within the network. Issues of increasing latency in the course of managing the traffic flow in critical flows were addressed via optimisation technique that deployed the MOCQ approach (a model for network management based on ART system). Available Bandwidth Utilization, Average User Experience, and Cumulative Distributive Function were used to determine the level of satisfaction of both the provider and end-user, thus attaining a stability point for both ends. The usefulness of this approach found its relevance mostly in a typical network provider environment (e.g., network virtualization) where the need for a particular network quality happens to be paramount and must be sliced at that level of importance to the end-users.

This solution to manage the bandwidth discussed in this chapter is relevant to the network organisations to use the agility of prioritizing bandwidth to help adapt to the other changing business requirements and new markets for prioritized quality. This usefulness is similar to one of the objectives of Network Function Virtualization (NFV) where time-to-market period is shortened because the infrastructure can be changed to adequately support the organisation's new products. The cumulative distribution frequency depicted that the network can adjust quickly and easily to changes in resource demands as appropriate such that the traffic coming to the data centre is regulated either as it increases or decreases. The adaptive rendering mathematical model that was deployed in prioritizing network bandwidth is one of the contributions of this thesis to the body of knowledge in the field. The contribution is significant with the additional advantage of SDN software which provided the controlled program capability feature to scale up or down the network demand(s).

4.7 Chapter Summary

This chapter identified the challenges of bandwidth utilization within a DCN environment. This research addressed this problem in two dimensions. Firstly, the chapter proposed a stability framework platform to measure appropriately the traffic flow which was fed into the stability analysis processes. To decipher the amount of bandwidth on the network, the chapter used the improved calibrated pathload which is hybridized with the correlation coefficient technique to promptly determine the available bandwidth. Moreover, one of the goals of the chapter is to achieve an optimal and stable network condition that guarantees the interest of both the service provider and the end user. This research analysed the traffic flow into three main flow categories which depicted the kind of flows that are found in data centres. Using various evaluation metrics such as Bandwidth utilization, Average User Experience, Cumulative distribution function, Reduced controller bandwidth capacity comparison as well as varying controller capacities, this research devised appropriate solutions to improve the stability of the critical flows within a network environment.

Chapter 5

UNSPECIFIED QUALITY APPROACH USING PROXY-CONTROLLER

5.1 Introduction

The previous chapter unveiled the importance of ensuring that critical flows are provided with an adequate mechanism that will enable guaranteed flow sustenance within the network environment. The performance of these network flows has become the dominant factor that determines to a large extent the user's experience thereby enforcing the need for high-quality network services. To keep the users in continuous demand, there needs to be a guaranteed level of user experience as well as continuous performance improvement. Several factors contribute to the quality of experience that users encounter in the course of network usage. As mentioned earlier, a long-standing challenge is an unrestricted increase in the number of users' subscriptions which has led to network degradation often referred to as network instability. This is usually due to the interference caused by the numerous devices of varying capacities connected especially when none of the network qualities has preference over the other (Abdallah et al 2018).

This chapter's contribution lies in investigating the background of SDN to address a network situation where no specified quality of service was provided within the network, thus implying that on average, the qualities of the network were relatively important for optimal performance (Caria and Jukan, 2016, Guan et al 2013, Hu et al 2014, Wang et al 2016). The solution to the unspecified quality approach in this chapter budded its innovation from the hybrid of Switch-Based and Transport Layer (the evolution of Proxy-Controller (P-C)). These two approaches were earlier used as explained in Chapter 2 by different authors though none of them tried to incorporate or take advantage of both in their solutions. The transport layer approach introduced the mechanism of a distributed system into the proposed solution while the switch-based mechanism was deployed in the form of proxy devices (Song et al 2017, Karakus and Durresi 2017, Akinola et al 2019).

This study tested the performance of the (P-C) approach to determine the level of path control and workload optimisation that can be achieved, thereby determining the level of network stability that DART_MCP can guarantee through MOCQ to the network system. Several authors have opened up the need to scale SDN based networks to evaluate the impact on network performance but our reference point in this chapter is on the interference problem incurred. The proposed solution at this point also profits from the research of Song and others in Song et al (2017) to affirm the need for determining the network stability evaluation of any approach that would be proposed in the field. This research, therefore, intends to fill the gap by answering the question of how the performance of a typical scalable network can be tested efficiently, to ascertain the level of network stability it guarantees when none of the network quality has preference over the other (Teo et al 2016).

In this chapter, Section 5.2 presents the proxy-controller mechanism and its valuable components for different functions to be carried by the solution modes (Active and Reactive modes). Section 5.3 presents the series of performance analyses with results obtained from the simulation and numerical analysis experiments. Section 5.4 presents the discussion of the results that were derived while the chapter concludes with a chapter summary in Section 5.5.

5.2 The Proxy-Controller Mechanism

The diagram in Figure 5.1 indicates the elements in the Proxy-Controller (P-C) deployment approach to network control management. It illustrates that the (P-C) deployment works between the data plane and the control plane (controller) to provide some updated information that is implementable on an SDN-based network through the application interface. The control plane in the network usually obtains the network states from switches and enhances proper management through the global network view.

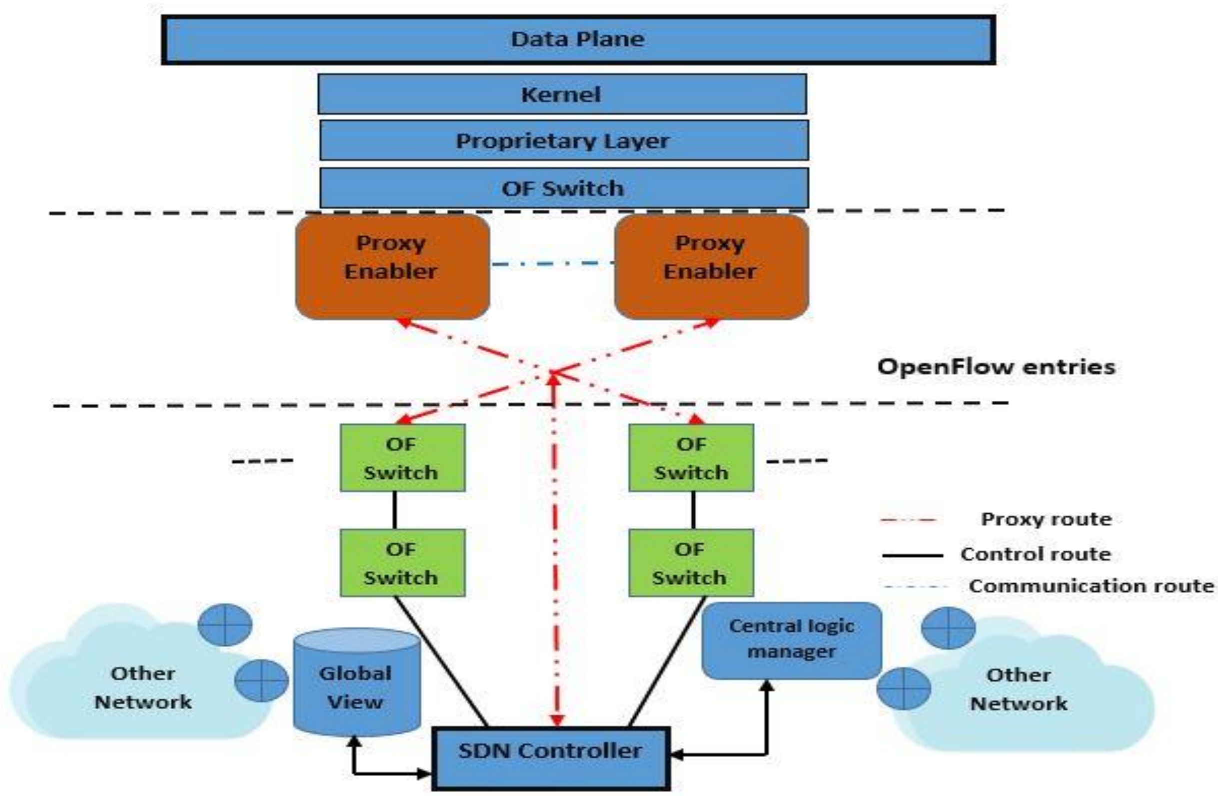


Figure 5.1: Proxy-Controller enabled SDN for Network Stability

Moreover, the diagram in Figure 5.1 depicts the OpenFlow switches from which the flow messages come via the proxy enabler to the controller. The proxy enabler has features that determine the OpenFlow messages dynamically and instruct the appropriate action on the number of filtered messages accordingly. The filtering process helps to reduce the number of messages that flood the controller to reduce the workload. The proxy enabler also coordinates all the local network states as well as handles the configurations of the switches. There is also an added provision for the proxy enabler to enhance the SDN controller in its connection to other external networks based on the scheduled capacity of controls that can be tolerated. This whole process is managed by the central logic manager by sending control messages via the proxy enabler. The central logic manager delegates packet-in messages to the targeted switch when there is a sudden upsurge in the number of the packets transmitted thereby avoiding the bottleneck of the network control path.

5.2.1 P-C Mode of Operations

SDN controller issues an assist request destined for the proxy enabler to initiate the need to compute or carry out a particular task such as resolving all the flow paths for a particular IP address. The proxy enabler amends the control logic manager to harmonise the forwarding rules updates and network states. When the OpenFlow switches send the packet_in messages, the messages are filtered appropriately, and corresponding matches cause the triggering effect to be realized by the proxy enabler. The unmatched messages are routed back through the control path to the SDN controller. However, by way of analysing the effect of the proxy enabler in the mode of operation, the proxy enabler contains some logic instructions which resolve the occurrence of network path sharing which results in instability.

The operations work in a dual mode of Active mode and Reactive mode. The active mode only makes use of the network adjustment module to stabilize the network and achieve the desired stability. The reactive mode uses the countermeasures feature to stabilize the network thereby attaining the desired stability as required. The diagram in Figure 5.2 depicts the content of the Proxy-Enabler wherein lies the dual modes discussed above. The key element in the enabler is the Network Adjustment (NA) module which resides at the centre of the enabler. The NA module contains the optimizer section of the ART algorithm, and it is referred to as the ART Optimizer

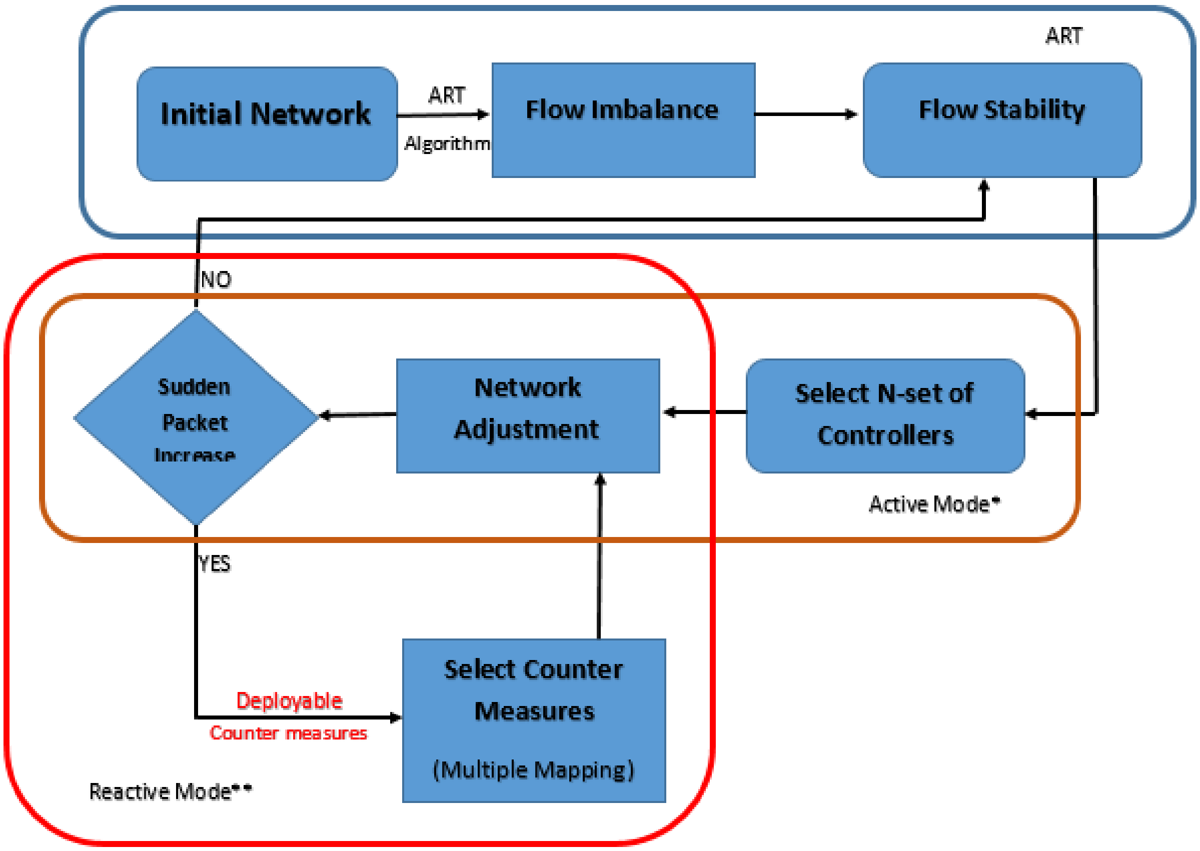


Figure 5.2: Proxy-Enabler contents depicting dual Mode Mechanisms for Network Stability

module. When the ART Optimizer module requests for the need of the countermeasure procedures, it has switched to the Reactive mode which is mainly to address the sudden upsurge in network flows.

5.2.1.1 Active Mode (MOCQ)

The active mode deals with the need to monitor the number of controllers that need to attend to the volume of traffic that is currently running. This is controlled through the ART optimizer of the Network Adjustment module. The major objective of this mode is to find an optimal flow switching for the network switches to the controllers in a manner that the network flow setup time is minimized while inhibiting the flow interference. This research assumes that a path is available for a switch to send a packet-in message to a controller and the path can be easily computed using Dijkstra's algorithm. If several packets in messages *P* were to arise from a switch *i* towards a path through to the controller j of many *C* controllers, if the maximum fraction of the traffic flows for the path is denoted by the resistance constraint ***d*** with the overhead load as $\gamma_j = \sum_{i=1}^{S} T_i P_{i,j}$ If the processing capacity of the controller is P_c and the propagation delay between the switch *i* and controller *j* is given by *Q*. thus representing these parameters in an optimizer (MOCQ) as shown:

Minimize:

$$\sum_{j=1}^{C} \gamma_j \tau_j + 2 \sum_{j=1}^{C} \sum_{i=1}^{S} T_i P_{i,j} Q_{i,j} \qquad (5.1)$$

Subject to:

$$\gamma_j \quad < \quad P_c \qquad (5.2)$$

$$\sum_{j=1}^{C} P_{i,j} = 1, i = 1 \dots \dots .. S, \qquad (5.3)$$

$$0 \leq P_{i,j} \leq \boldsymbol{d} \text{ , } i = 1 \ldots\ldots S, j = 1 \ \ldots\ldots C \qquad (5.4)$$

The optimizer ensures fairness in the traffic flow thereby preventing the occurrence of interference. One of the significant effects of the occurrence of interference is the undue time spent in flow setup time as well as the rate of failure in data transmission over the network which is referred to as the network failure rate. Therefore, determining the average sum of response time by the to and fro time delay between a switch and a controller often gives a direct relationship with the amount of stability that a typical network has to offer. Hence, the objective function in the above optimizer expressed in equation (5.1) represents the total time taken for the setup requests to be carried out within the network. The constraint (5.2) is to guarantee that the overhead of the controller(s) will not be exceeded. The constraint in expression (5.3) ensures that there is a unidirectional movement of packets-in messages by coming from a Poisson distribution to be directed to the controllers. Constraints in equation (5.4) give the resistance constraint which restricts path sharing that results in interference in the packet-in message sent between switches and controllers.

5.2.1.2 Reactive Mode (Multiple Mapping)

The reactive mode which is shown in Figure 5.2 is practically a countermeasure process that is designed to use multiple mapping mechanisms. The option is deployed to ensure distributed flow setup requests among the controllers and switches. This approach is referred to as the multiple mapping scheme according to (Hu et al 2012) and (Selvi et al 2016) which uses a heuristic proposition to ensure that flows are evenly balanced. Multiple mapping ensures that a coarse-grained directional flow from source to destination pairs per flow. The distribution is considered at the least level of individual flow as against the aggregation of flows coordination to the destination. This research modelled the shadow mapping approach used in Guennebaud et al (2007) to bring about the multiple mapping solution to the individual flow in an OpenFlow network. One of the problems with the single mapping that is often deployed in the OpenFlow network is that it often requires needless re-mapping of the switches in the network to the controllers during traffic variations thus causing delayed flow setup and various interferences. The single mapping approach reported in (Wang et al 2016c, Cheng et al 2015, Gao et al 2016) did not flexibly distribute the flows at each instance, preventing the needed switch resilience for stability.

The reactive mode considers mapping each flow from the switch to more than one controller in the case of high traffic fluctuations. Considering the available bandwidth for controller operating normally as B_k with the required flow bandwidth of B_r, given that the flow distance from the switch to the controller is Z_n, where the resilience constraint is given as *r*, this research represented the maximum bandwidth available for any flow by B_n, and the rate of the feasibility factor of the optimal flow as Z_p . Therefore, writing an iterative flow optimal projection as Z_{min} about the expression given in equation (5.5) as:

$$B_k = B_r \frac{Z_n r}{B_n} \left(\frac{1}{Z_{min}} - \frac{1}{Z_p} \right) \tag{5.5}$$

The first estimate of the above gives the least *min* flow value which is Z'_{min} directly obtained from running the initial sudden flow at a level of $[\log_2(B_k)]$. This least *min* flow value allows the traffic flow size to be optimised iteratively until the optimal convergence is attained.

5.2.2 Working Scenario of MOCQ

The ART approach proposed in the research is believed to address the flow management challenges within the network flow by reducing the probability of DCN flow interference via Multi-Objective Optimization of the Crosspoint-Queue Management (MOCQ). This approach deployed the hierarchical setup of network controllers with switches to favour MULTIPLE mapping of controllers that MOCQ took advantage of in our study. Multiple controllers on the same hierarchy are mapped together to suddenly address smaller/short flows in the reactive mode whenever interference is likely to occur in the network. The main desire is to achieve stability between the possibility of high flow management concerning the network overhead accrued. This research understood that the assignment of dedicated memory and procurement of larger bandwidth will help in addressing the possibility of network flow interference, but it is usually at a higher cost to the service providers. Moreover, such surplus is not practical or exists in the reality. Such assignment can also not be achieved without obstructing the protocol stack of the network element. The solution in this chapter, therefore, addresses the running problem in two mechanisms approaches through the following steps:

- Categorising flows in a fashion similar to light rays in an adaptive rendering prototype such that appropriate priorities are given to bandwidth demanding flows and less delay time are utilized by the light/time-sensitive flows adaptively.
- Provisioning of a reactive traffic variation intervention mode by assigning switch flows to more than one controller through adaptive multiple mapping mechanism thereby correcting the flow interference within the same switch flow path.

The ART assigns bandwidths to both the input flow and output flows separately. All incoming flows are scheduled for appropriate memory buffer and on appropriate execution are ejected through output memory buffer. The ART uses the queuing theory according to the M/M/1 queue thus ensuring that the flows are fairly assigned on arrivals. The separate memory buffers accounting for each switch path are appropriately normalized not to compete for the path with one another even though the flows could be going out through a similar output memory buffer. The M/M/1 theory is similarly applied in the output section to ensure that the output paths were well separated to avert the possibility of output path interference.

During traffic flow increase, the input memory buffer could suddenly be full and according to the approach in coarse output queue (OQ), the tail simply losses its flow. However, the ART ensures that each flow is scheduled for pipeline using its adaptive feature (Primary control pipeline stages) to give light-sensitive flow (Small flows) its needed time priority without interruption to the larger flows thus keeping interference at check amidst the flows. The light-sensitive flows are given priority over the larger flows because they can be completed in the shortest time frame. The multiple mapping mechanism also enables the traffic to be shed off to

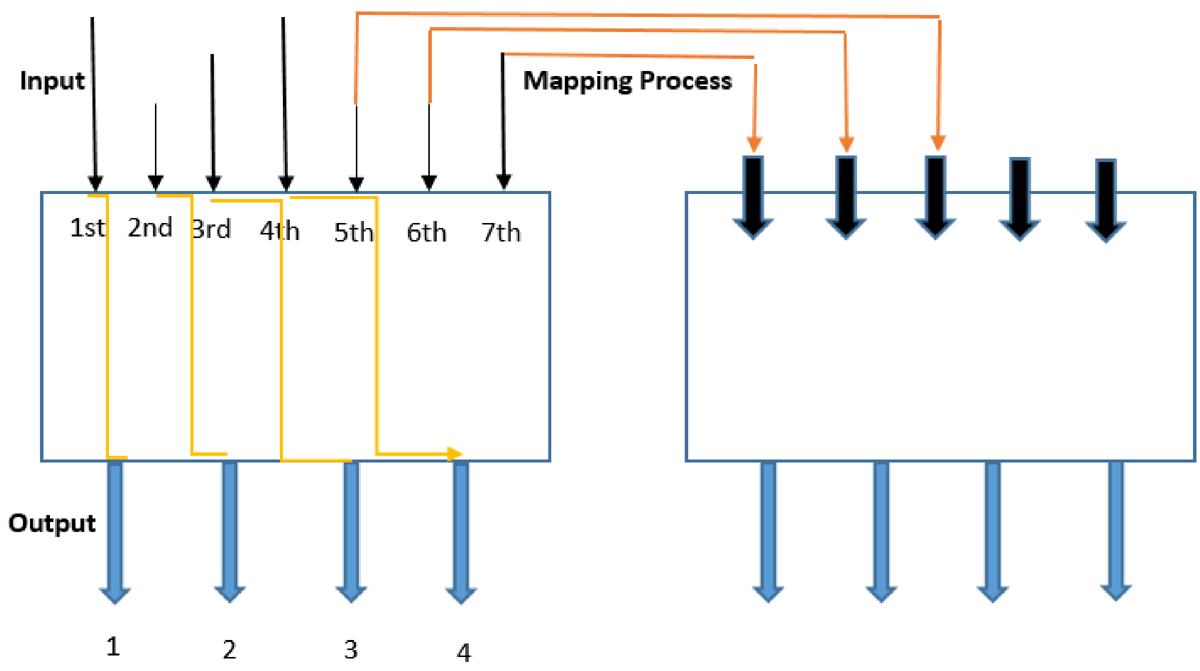


Figure 5.3: Switch Mapping process of MOCQ

adjacently mapped controller upon buffer fill-up which accounts for the second stage of the P-C process called reactive mode according to Figure 5.3.

At the input stage the ART access the length of the flow to know its type of flow using l < 100kb as smaller flows, l > 100kb as larger flows, and l > 1Mb as giant flows (Chen et al 2015). The flows are assigned the paths based on the M/M/1 scheduling algorithm and light-sensitive flows are assigned immediately to any available mapped controllers instead of waiting to experience dropping or incur extra processing delay.

The output stage ensures that the output empty buffers were assigned to the incoming output flows according to the scheduling approach. These scheduling reoccur iteratively and subsequently disannul the possibility of contending paths.

5.3 The Performance Study of the Mechanism

An exhaustive test of the novel ART approach is carried out in three phases. The significance of these tests is to ascertain the interference impact of the technique on the network flow setup that causes instability in the network. These tests are:

5.3.1 Network Fairness and Relative Stability

This section describes further the performance of the numerical analysis explained in both Chapters 3 and 4. The performance is compared with a simulation setup in the latter part of subsequent sections with a similar numerical analysis. Two first sets of results from the numerical analysis were derived to address the average flow set-up time that is achieved in the network, about stability as well as the level of fairness attained. These were achieved by solving the optimization problem that was derived in averting the flow setup interference in the network.

The resistance constraint introduced ensures that the level of flows on each of the switches does not affect the failure of the connected controller. The analysis confirmed that frequent re-assignment of the switches accrued to a disadvantage to the network in terms of the stability that is experienced as well as the time wasted which is directly proportional to the delay time experienced in flow setup time. This necessitates that there is a need to study the impact of the variation in packet-in message rates on the stability of the network when the Proxy-Controller (P-C) is deployed as against when it is not deployed. The study determined the effect of the reactive mode during a sudden influx of flows by accessing the impact of the Multiple Mapping Mechanism as against when it is not deployed.

This research consider by numerical analysis, the network size of 250 switches and 60 controllers arranged under the Hierarchical topology, setting the propagation delay between the controllers and switches to range from [0.1 to 1] ms. This experiment uses the controller capacity of 1500 packet-in messages per second, with the rate of the packet-in message set randomly at the range between [100 to 500] packet-in messages per second which are expressed usually in the traffic matrix. The resistance constraint ***d*,** which reflects the peak proportion of packet-in messages that a controller could receive from the switch ranging also between [0.4 to 1]. Hence, in the Reactive Mode, the resistance constraint is represented by ***d***RM.

5.3.1.1 The Fairness Performance

The traffic matrix enables generating several hundreds of packet-in messages randomly for the test of network fairness. By fairness, this research refers to a measure of attaining an optimal flow setup time within each available network switch as well as a guaranteed resistance constraint provided to the controllers in use. The associated metrics for accessing the level of fairness under this section include relative standard deviation and max-mini ratio of the reactive mode. The relative standard deviation of the switch mapping derived concerning the measure of fairness expresses the ratio of standard deviation to the mean flow setup time that is achieved while the max-mini ratio expresses the ratio of the switches with maximum flow setup time to the minimum flow setup time for the mapping solution achieved. The result for the fairness performance in terms of preventing rapid variation during flow setup times is shown in Figure 5.4.

The average ratio of maximum to minimum reactive mode is expected to be relatively high for a fair network output as this gives the expression of the relative performance of the network mapping solutions during network variations. As the number of traffic matrices increase, the figure accounts for a 40 percent improvement over the cases where there is no reactive mode (nRM) in place as the ratio drops from 1.9 to 1.1. This result implies that this discovery can enable or guarantee the stability of any scalable network through the implementation of the reactive mode analysis of the ART approach.

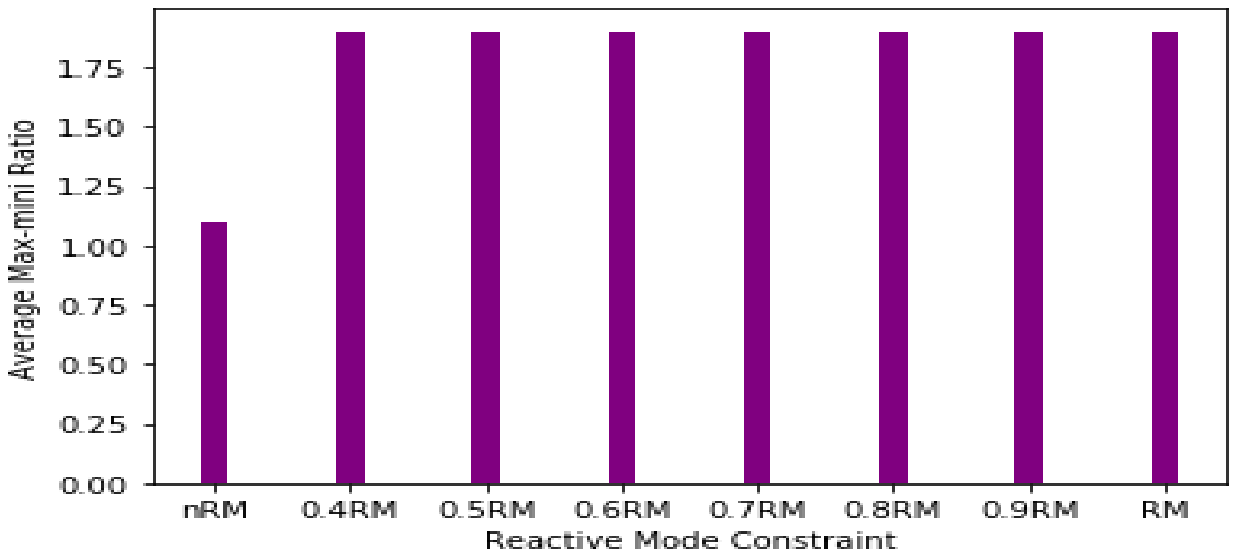


Figure 5: The rate of average Max-mini switching ratio

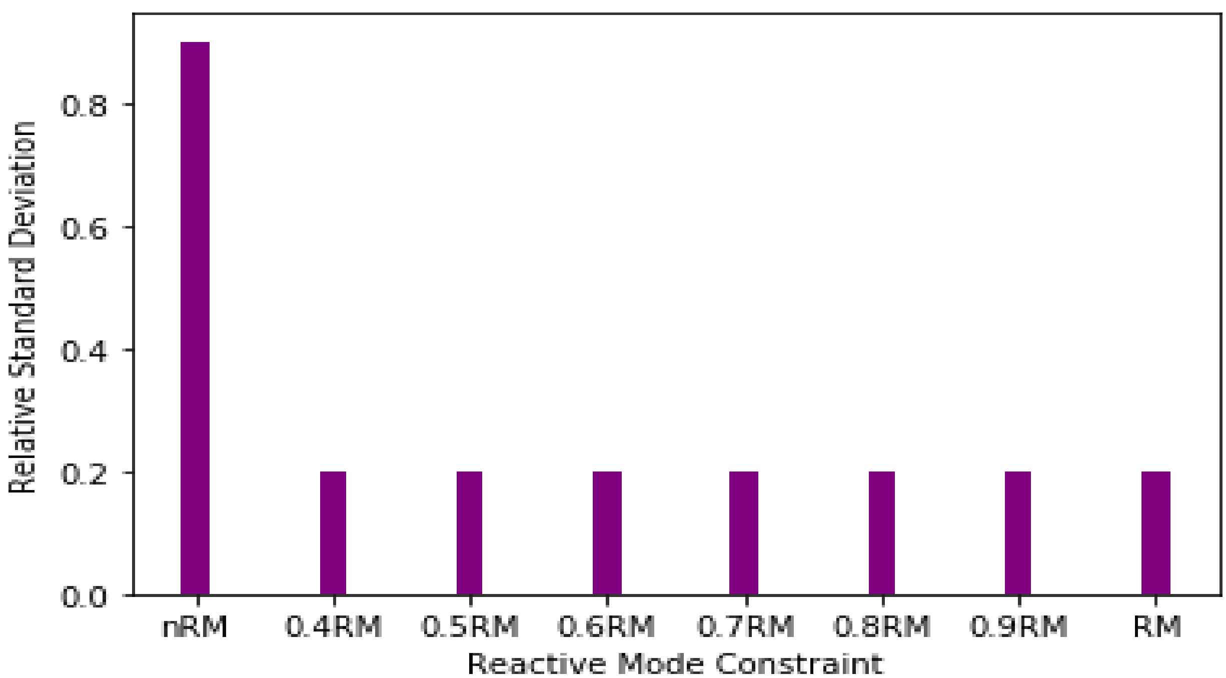


Figure 6: The Relative Standard Variation output (re-mapping fluctuations)

Furthermore, the relative standard variation of the P-C approach result is depicted in Figure 5.5. The rate of variation is very high with nRM in place which implies that the deviation from the mean flow setup time is virtually unstable and almost a multiple of five times the reactive mode approach. This further implies that the rate of switching is very high with the reactive mode mechanism which enables the network to be highly prone to instability. Hence, both tests that were conducted under this section confirmed the relative fairness in flow setup distribution within the network with the reactive mode approach in place. This test measures the momentum of mapping and re-mapping of switches to controllers in which a high level of re-mapping confirms the level of instability within the network environment, as the reactive mode approach requires lower setup time according to Figure 5.5.

5.3.1.2 The Relative Stability

The test in section 5.3.1.1 above presents the expected behaviour of the ART algorithm in controlling the mapping of the switch to the controller. The test for stability itself is very significant to the contribution of this study. This section evaluates the impact of the approach on the stability of the network specifically concerning the flow setup management within the network using the simulation approach. This study aimed to check the approach proposed on a network simulator for its efficiency in affirming a similar concept. This research conducted trace-driven OMNeT++ based simulations to evaluate the impact of the introduction of P-C on stability. The primary goal is to access the level of network stability that is achieved in the course of the deployment. This research conducted our simulations with the generally implemented fat-tree and Virtual Layer 2 (VL2) topologies. Using 24 pods, with 3465 hosts and 720 switches. Considering the VL2 topology setup, the degree of the aggregate switch (DA) and the degree of the intermediate switch (DI) is set to 48 with 576 racks each hosting 10 hosts. This research set the O(V2) as 105 with the number of N controllers to be 50, given that the capacity of each of the controllers is 1600K flows/s.

The inter-arrival time of the packet-in message is followed closely by the request arrival rate, just as seen in the real-world typical fat-tree data centre. Thus, given that the assignment that exists between controllers and switches was represented by *N by M* with matrix Q which satisfies that a switch is directly connected to at least one or more controllers that serve based on the multiple mapping principle earlier discussed. Using the value of *k* as a load factor, evaluating the performance of the P-C approach under different load conditions and the absence of P-C (Akinola et al 2019). The study determined the mean number of swapping in mappings over several traffic matrices that were generated and the values over a range of network variations were shown as represented in Figure 5.6

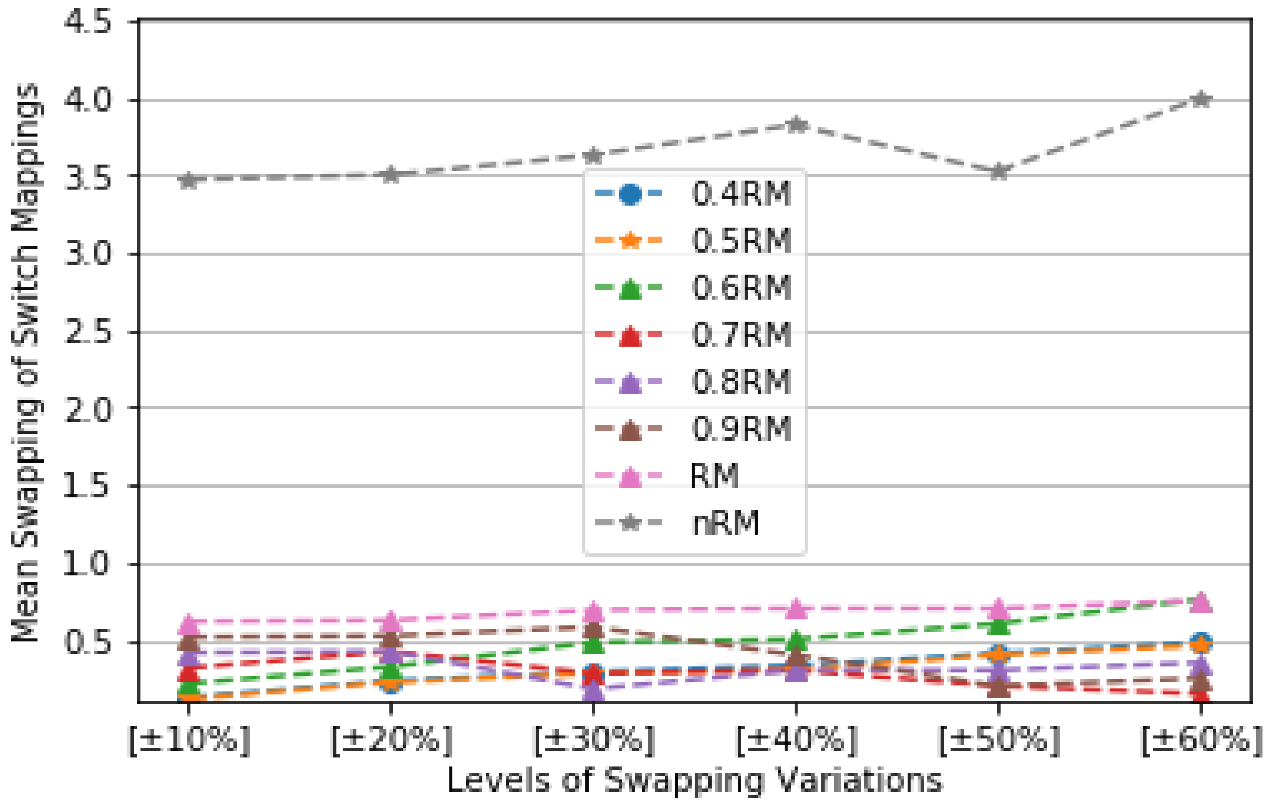


Figure 5.6: Mean swapping of switches during reactive mode mapping

The chat in Figure 5.6 depicted the result of the frequency of swapping of switches to controllers within each number of traffic matrixes that were introduced into the network flow. The occurrence of swapping in the situation where P-C was not deployed was noticed to go as far as four times when it is not used. Most of the time, as the system tried to adapt through the algorithm used, there is a need to utilise all available options from the controllers especially those that are not operating at maximum capacity. Hence, there is no need for newer mapping to take place as in the case of *nRM* which is constantly looking for a new controller to enable the completion of the flow task. This research also noticed that the resistance constraints were maintained within a reasonable limit for optimal performance in this situation with variation in traffic flow. Thus, both the ART optimizer of the Network Adjustment module and the multiple mappings of the reactive mode ensure the stability of the traffic flow as experienced in the diagram in Figure 5.6. Similar results could be seen as well from Figure 5.7. The diagram depicted the average maximum to minimum fraction at each packet-in message that was sent. The figure revealed that the relative resistance level is maintained at a higher state in each constraint introduced than when the P-C approach is not used. The resistance constraint of 0.7 which is represented in the black legend was able to maintain both the peak and least of the fluctuations appropriately across the boundary depicting that this level of constraint at that point is the most stable of them all.

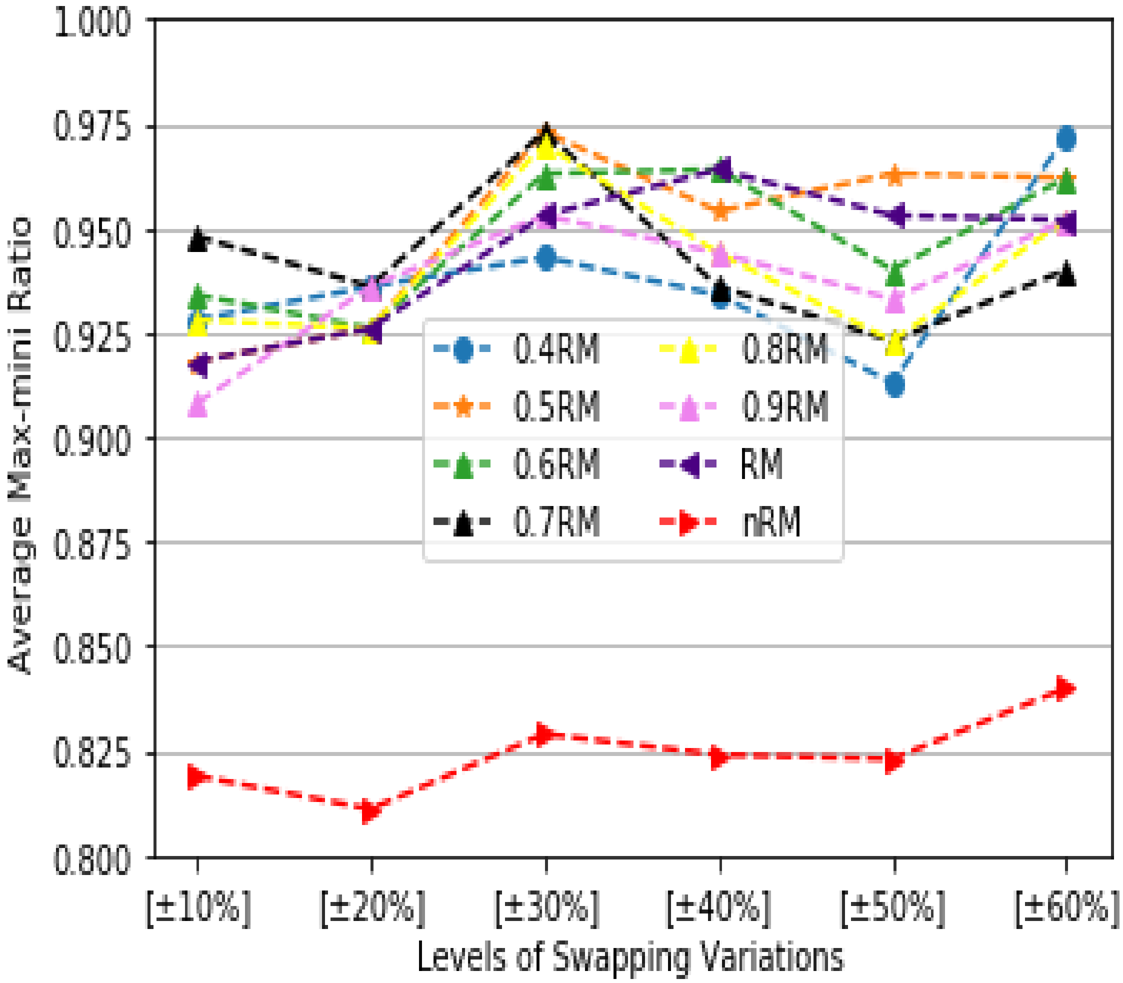


Figure 5.7: Average Max-Mini Ratio

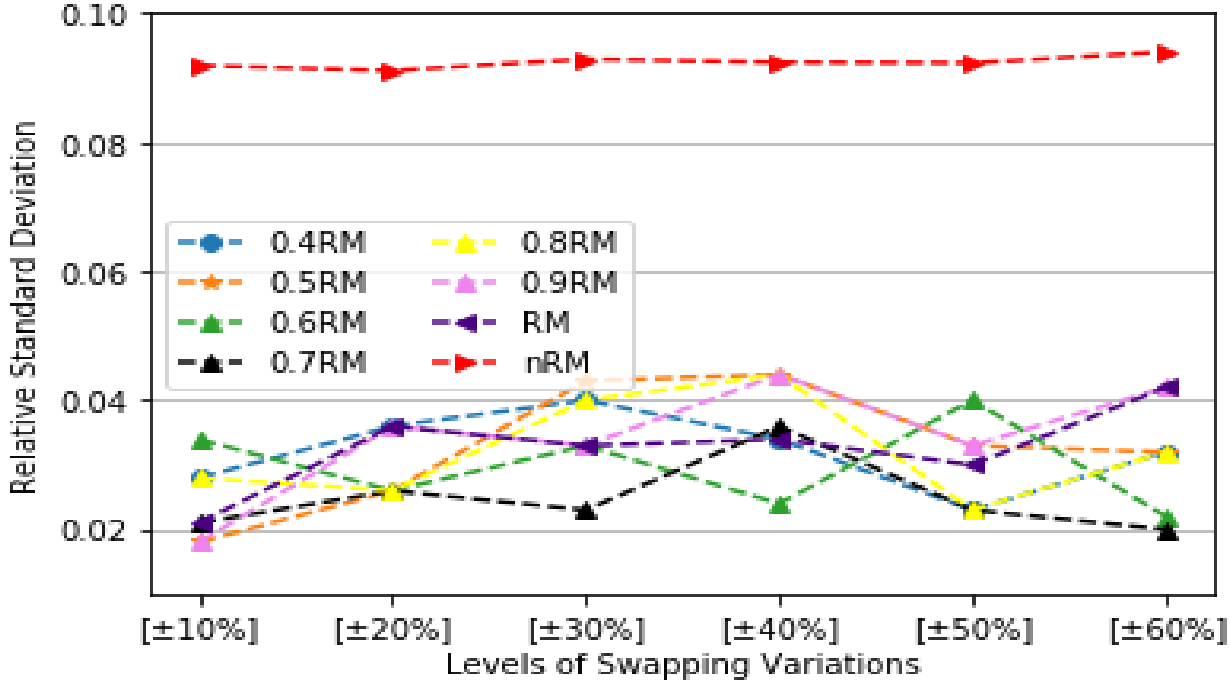


Figure 7: Evaluation of Relative Standard Deviation.

Figure 5.8 depicted the result of the relative degree of deviation that accrued at varying traffic matrixes. The deployment of the P-C approach gave less deviations as seen in Figure 5.8. However, the nRM depicted a higher degree of deviation at almost 0.1 thus resulting in an unstable network situation. Again, the resistance constraint of 0.7 gives the best level of stability in comparison to others in this simulation. The possible reason for 0.7 constraints having the best stability response is that the point of equilibrium of network topology and traffic flow manner is often attained through this way and over above which the performance is bad and under which the stability is not so desirable. Therefore, in designing a particular stability mechanism for a typical network for a company or organisation, there is a need to carefully determine the optimal amount of resistance constraint that is to be set in the network ART optimizer module to guarantee optimal network performance. Due to the importance of the resistance constraint that was determined in this network, this research decided to use the resistance as a yardstick to test the performance of the numerical analytical model of our mathematical approach as well as the simulation through a network simulator OMNET++ that this research has used in this section. The result is discussed in Section 5.3.2.

5.3.2 The Significance of Resistance Constraint

Having understood the need for a resistance constraint in the network which helps to instil network stability through a guaranteed measure of the level of stability, determining the range of stability that each of the constraints introduced could achieve for the same typical network setup under this section was very vital. This test is important majorly for two reasons: one, it helps us to determine relatively at which stage or level that the introduced constraint gives a maximum or optimal performance; and secondly, accessing through the outputs if the two research methods that was deployed so far are directing us to similar results which invariably confirm the feasibility and correctness of our proposed ART theory.

Strictly, introducing resistance constraints that range from 0.1 through to 0.9 such that as much as the constraint tends to unity, the performance most likely is going to deteriorate. Using the numerical analysis of the mathematical derivations in Chapter 3 and the algorithm explained in the same chapter, this research determined the degree of relative standard deviations that each method could achieve, and these were presented in Table 3.1. The figure in Figure 5.9 also depicted the output shown in Table 3.1 accordingly with the explanations following the figure.

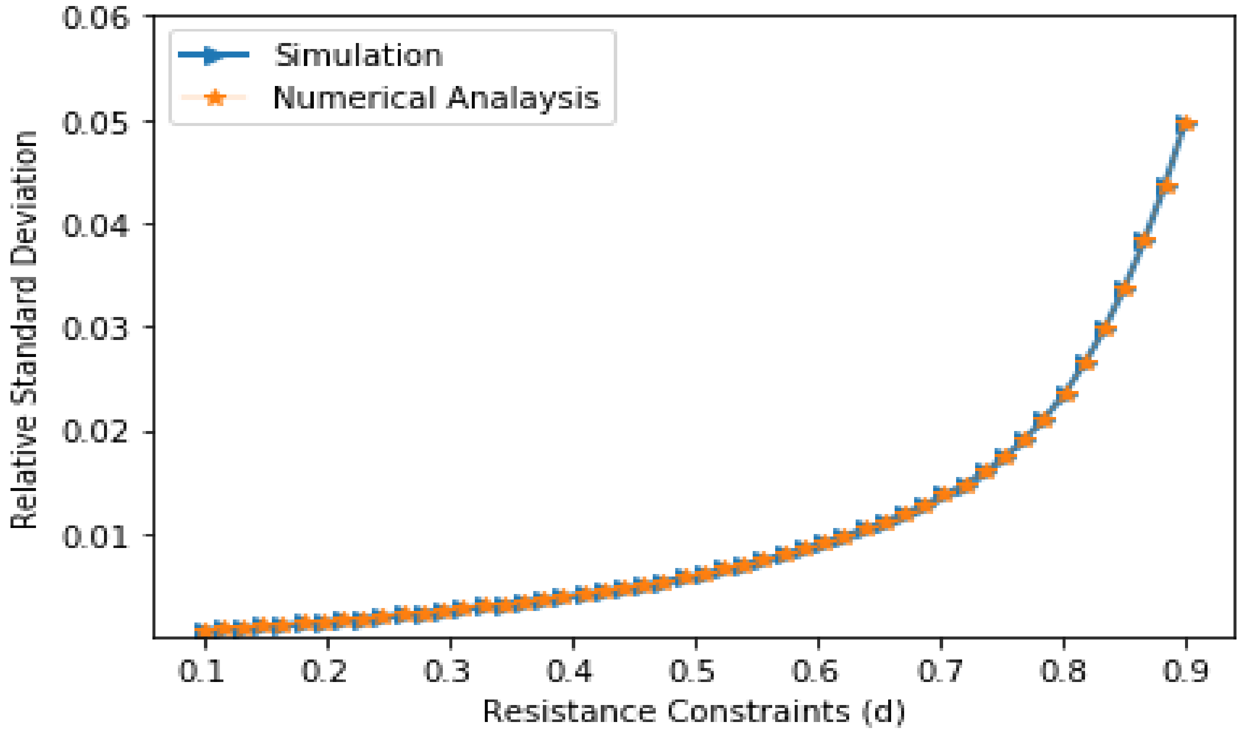


Figure 5.9: Comparison of Relative Standard Deviation from Numerical Analysis and Simulation

Figure 5.9 shows the result of the output when the value of resistance constraints is varied to derive the corresponding values of relative standard deviation for both the numerical analysis and simulation methods. The initial values are similar and tightly close to one another but as the result progresses, the two methods vary very minutely but maintain similar moves or curves as they progressed. Two major conclusions can be deduced from the diagram in Figure 5.9. One, there is a major balance that is common to both methods when the resistance constraints were at 0.7 which is also confirmed by the result from Section 5.3.1.1 and Section 5.3.1.2 of this chapter. With the constraint maintained at 0.7, the network performance is maintained at optimal level and the traffic flow interference is kept off the scene. Two, the uniformity in the direction of the graphical representations showed that both the numerical analysis as well as the simulation methods resulted in similar solutions, thereby confirming that solution worked perfectly for the considered problem.

5.4 Discussion of Results

The stability of a network when there were no preferences amongst the network qualities often proved to be a challenging task. This is because all the variables within the optimal equations are equally relevant at any instance in time. The proxy controller served to be an intermediary to measure the smooth running of traffic flow to the respective destination via the Active and the Reactive mechanism that was proposed. The goal is to be able to determine a resistance constraint for each network under consideration through which it can guarantee a level of stability. The frequent mapping and remapping of network flows often determine how stable or unstable the network environment would be. In this chapter, the specific network under consideration in these experiments attained a resistance constraint of 0.7. The determination of the resistance constraint via the ART approach serves as one of the significant contributions to the body of knowledge in this thesis.

In the same vein, considering a standard real-life DCN, it is highly imperative based on the findings in this chapter to first determine by experimental analysis the resistance constraint of the network. This will go a long way to ensuring that the involved stakeholders enjoy guaranteed stability such that problems like over or underperformance within the network would not be a subject of consideration. It further advises the service providers and the end-users on the assessment of the level of service guarantees to be delivered and accepted at both ends. On the other hand, the level of resistance constraints introduced could also be regulated by the service provider to meet up a specific requirement of the end user. This is further expressed to imply that the resistance constraints could give the level of numbers of end-users that could be accommodated at each instance of time before the 'yield point' is attained. This discovery could, however, be the pointer towards taking an informed decision for maximum benefits at both ends.

5.5 Chapter Summary

This chapter presented the implementation of Proxy-Controller (P-C) in the network environment to eradicate the occurrence of traffic flow interference. The study explained that the P-C enabled approach utilised both the active mode and the reactive mode to address the network challenges. The active mode uses the ART Optimizer module to actively filter out the possibility of flow contending for a similar path. However, with the occurrence of variations that could suddenly overwhelm the network, the reactive mode quickly addresses the instability within the network. The reactive mode uses the multiple mapping of the switch to the controller to provide alternatives which will not only dissolves the path contention problem but also address the delay in flow setup time as well. The approach also inculcates the resistance constraint value(s) which helps to enable a measure of stability within the network. The experiments were able to introduce a varying degree of resistance out of which they all performed relatively well but were later able to determine the unique state of 0.7 as an optimal resistance value for the type of network that was considered in this thesis. Furthermore, the chapter was also able to ascertain the performance of our proposed approach with two research methods used in this chapter which are numerical analysis and simulation. Both methods were seen to deliver almost similar results for the same problems. The subsequent chapters further utilized the established idea to solve similar challenging network problems but in a different real-life context. The MOCQ's performance would be tested in a use-case scenario in Chapters 6 and 7 to confirm how efficiently the algorithm performs relative to others.

Chapter 6

ANALYSIS USING NETWORK PERFORMANCE METRICS

6.1 Introduction

Chapter 5 elaborated on how MOCQ can be used to address an unspecified quality metric of a typical network by improving the management of network metrics. This possibility makes our solution suitable for a real-life challenge such as the management of Public Health through the Public Health Management System (PHMS) which is often reported in literature as having network bottlenecks in recent times (Saul et al 2018, Akinola et al 2020, Cicioğlu and Çalhan, 2019). Although the management of a large number of patients' information has been studied, their networking aspect remains a challenge in the prompt assessment of patients' records. The deployment of MOCQ on the network controllers enables the regulation of the control flow logically through its intelligent control thereby enabling efficient management of the traffic flow interference which lowers the occurrence of network latency and enhances more network stability. The goal of this chapter is to access the level of satisfaction that the service providers can guarantee at the attainment of MOCQ stability which is very important in keeping network users in the market.

Several providers need mechanisms that can improve the performance of their service delivery and it was based on the necessity of such mechanism that MOCQ found its relevance in guaranteeing network stability. Thus, this chapter deployed the DART_MCP algorithm (using the MOCQ approach) on a health-based framework to optimise the performance of a large-scale PHMS environment. This chapter contains Section 6.2 which describes the proposed network design setup for the PHMS while Section 6.3 discusses the simulation pictorial view of the network setup. Section 6.4 describes the design test for miniature testbed setup for PHMS service performance while Section 6.5 describes the network performance metrics used for the system evaluation. The comparative selections of relevant existing works with explanations of the results are discussed in Section 6.6. The discussion of all the results was conclusively carried out in Section 6.7 while Section 6.8 summarises this chapter.

6.2 Proposed Network Design

The proposed design for PHMS was based on the SDN technological approach which advocates that the control plane needs to be abstracted from the data plane for enhanced programmability and proper monitoring of the network system environment. This novel technological approach enhanced network scalability and faster transmissions of data over the network. With the view to making the control of the network flexible through programming, this study devised the network layout as shown in Figure 6.1. The key features in the figure consist of the control plane in the controllers and the data plane where the switches are connected to the network hosts. The Master controller is the feature through which the entire coordination is carried out with the help of some sub-controllers called Zonal and Location controllers.

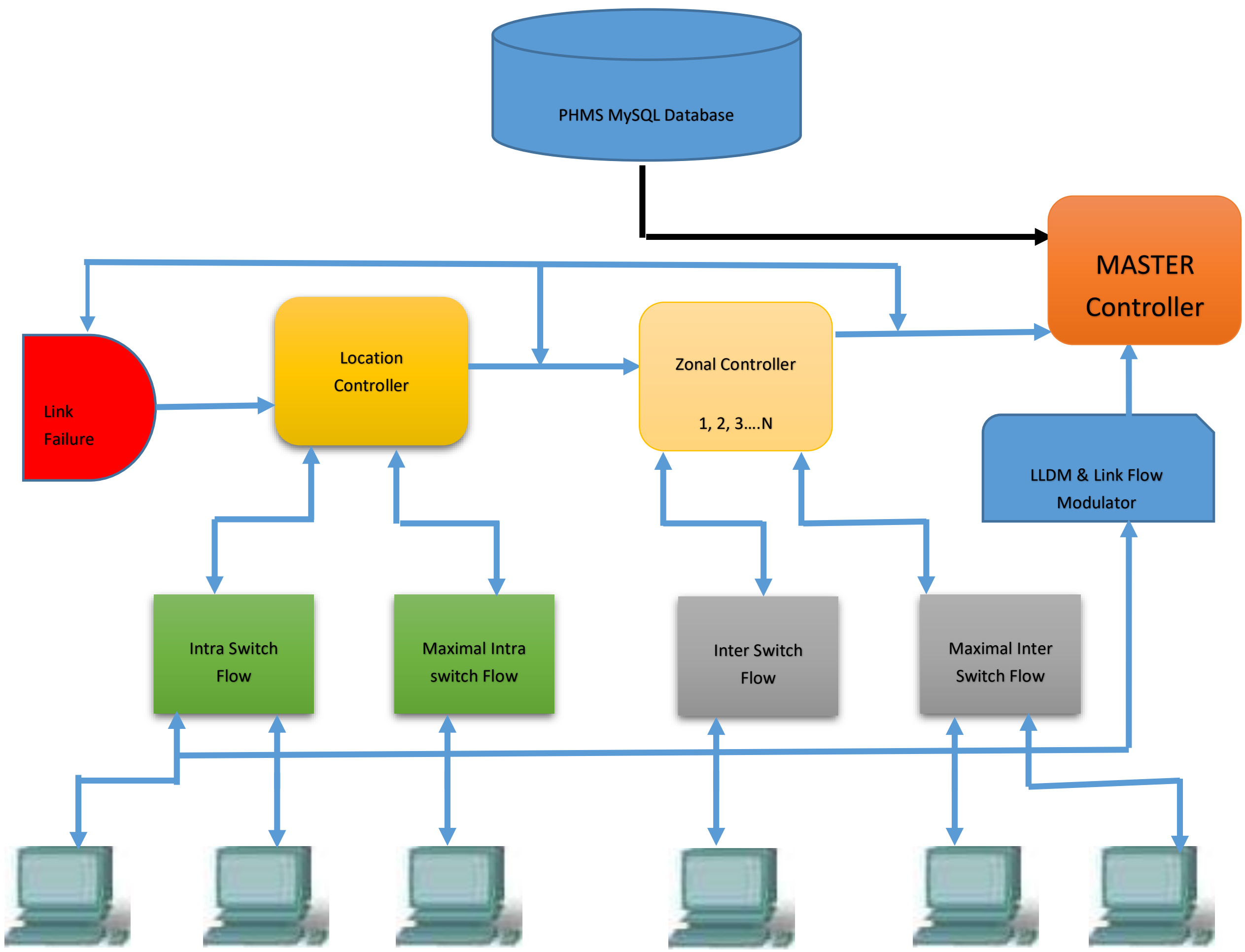


Figure 8: Control Plane Design Model for Robust PHMS Network

Other network parameters include the Database which houses the patients' information and the forwarding elements called switches which help in forwarding network host data to the appropriate direction. The innovation in the network design here is such that the "controllers" enhance the outright modification of the whole network to behave in a desired manner required by the network administrator. In addition, the study innovatively separated SDN design deployment into three sets of controllers for enhanced coordination of network flow and maintenance of its stability. Thus, the control planes were structured as shown in Figure 6.1, containing the location controllers, zonal controllers, and the Master Controller hierarchically. The hierarchical manner of network arrangement was based on our recent findings in our research on controller designs for Data Centre (Akinola et al 2018).

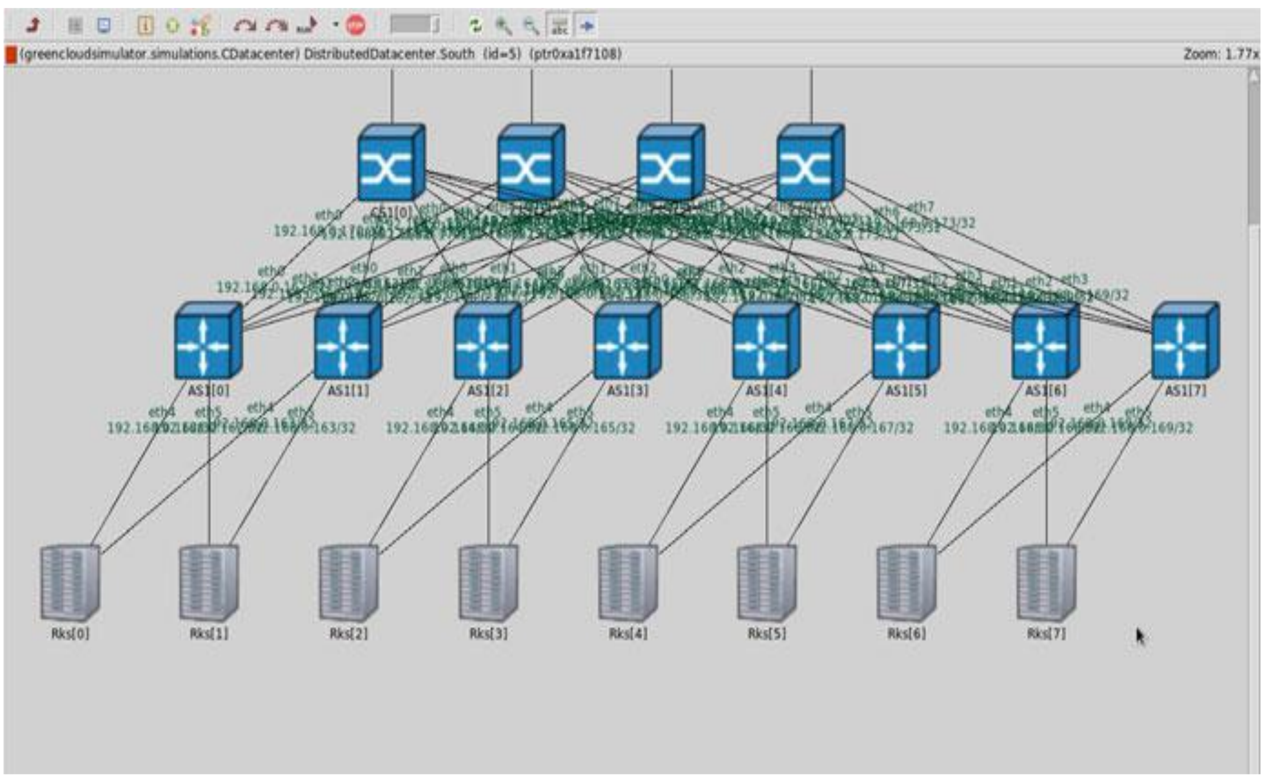


Figure 6.2: Hierarchical Simulation Setup

The controllers were aided by two different types of switches namely Intra and Inter switches which forward flows from various network hosts to the controllers majorly those around its zones and locations respectively. The maximal intra switch flows are controlled by the Location controllers while inter-switch flows are coordinated vis the zonal controllers which are in turn managed by the Master controller. The structured hierarchical arrangements were designed to enable extensible and proper control of the traffic via programming of the control plane interface. The simulation perspective of the diagram in Figure 6.1 is shown in Figure 6.2 depicting the hierarchical arrangement manner for network design. Moreover, the link flow modulator linked the multiple signal flow with the controllers and ensured that appropriate signals were motivated to achieve a balance in the traffic flow.

The link flow counter usually reports the number of possible traffic failures (link and switch failures) that were experienced in the system operation with the link flow modulator, keeping the master controller instructions to restrain the possibility of the traffic failure and as well salvage the situation for quick fault recovery. A Link-Layer Discovery Module (LLDM) protocol is used to discover the link information that interconnects both inter-location and intra-location traffic flows. Thus, multiple switch locations often forestall the need for appropriate link discovery towards network management. The patients' data are finally stored in the database provided at the top of the network design for writing, reading, and deleting purposes. Thus, the need for the database to be updated with the current information as permitted by the administrator via the Master controller. The master control plane can provide the network's logical wide view for better decision making to stabilise the network flows. The design combined the hierarchical and programmability of the control plane by deploying the DART_MCP algorithm for the PHMS system thereby enabling an extendable but stable network environment.

6.3 Design Simulation for Network Setup

The setup and experimental testing were carried out on OMNeT++ simulator with INET framework installed to provide the routing protocol for the experiment on core i7, 8G RAM 1T memory Asus PC. The goal is to assess the performance of our proposed MOCQ approach in comparison with the other proposed and evaluated approaches in literature as was reported in Chapter 2. The switches were arranged in layers to depict the hierarchical format of the design as depicted in Figure 6.2. The simulation selected OpenFlow switches for the simulation which would enable the sending and reception of traffic flows within the network. The capacity of the various controllers varies from the Master to the Zonal as well as the Location controllers. The Master controller is directly accessed by the administrator who through the master controller could update new protocols, run new control flow programmes, and monitor the whole network. This process ensures that the rapid effect is made on the network states thereby ensuring that the performance of the entire network is rapidly improved. The settings of the controllers and switches were designed according to Table 6.1 for the hierarchical network design based on the literature (Step 7. Configuring a hierarchical network — INET 4.3.0 documentation, 2022). For the experiment, the maximum number of host (nodes) $N_p = \frac{d^3}{4}$ hosts where *d* is the radix or degree. Using x = number of local switches, y = number of network links, g = the number of hierarchies in

Table 6.1: Simulation Environmental Settings

SN	Scaling Parameters	Simulator Settings
1	Simulator Name	OMNeT++
2	Numbers of Controllers	8~155
3	Number of Switches	18~20,480
4	Simulation duration	1200 seconds, starting packet sending from 1000th second to last for 100 seconds each
5	Number of Nodes	80~53,248
6	Modules	8~4096
7	Controller degree	2~5
8	Switch degree	4
9	Diameter	4~10
10	Betweenness Centrality	0.06369~0.00005
11	Assortativity coefficient	-1
12	Spectral radius	4~6.32455
13	Algebraic connectivity	0.87689~0.64765
14	Avg. node degree	3.2~6.15385
15	Density	0.04051~0.00012
16	Links	128~163,840
17	Levels	2~5
18	NC = τ (Normalized network criticality)	1.06899~0.30542

the architecture and z as the density. Then, the number of host N_{p*} = g.x.z = (x.y+2) x.g = (2z.z) = $4z^4$ under a stable condition of the network state. This research assumed that under the balanced state of the network, the network has x=2z=2y while the controller degree d = z+x+y-2. Therefore, determining the maximum number of hosts that the network can tolerate with the relation $N_p \approx \frac{d^4}{64} + \frac{d^2}{8}$.

In considering the cost of network setup for software-defined networking, representing the number of the switches by $S_{sdn} = \frac{6\text{Np}}{d}$, having $\frac{d^1}{2}$ master controllers, $\frac{d^2}{2}$ sub-controllers (Zonal and Location) and $\frac{d^2}{4}$ multiple switches in the network.

Similarly, if the number of switches in conventional networks design was given to be S_{con} = g.x $\approx \frac{5\text{Np}}{d}$, determining the cost of networking for the number of nodes per switch, and the time frame for packets delivery (PDT) which is given by the addition of transmission (propagation) time (TT) and propagation delay (PD). This is expressed in equations 6.1, 6.2, and 6.3 as:

$$PDT = TT + PD \quad (6.1)$$

$$\text{TT} = FST + LMD + QD + NPD \quad (6.2)$$

Flow Setup Time (FST) is the Frame Serialization time, given by the ratio of packet size (bits) to link data rate; LMD is the Link Media Delay, given as the ratio of link distance (m) to processing delay; QD is the Queuing Delay, given by the ratio of Queue depth (bits) to the link data rate and NPD refers to the Node Processing Delay which is specified depending on the users' machine.

$$\text{PD} = \frac{Link\ distance}{Propagation\ speed} \quad (6.3)$$

But the speed of packets in copper wires is $2 \times 10^8\ ms^{-1}$ and it is equal to the speed of light in a wireless medium as $3 \times 10^8\ ms^{-1}$ since using a process that mimics the working principle of light rays [17]. The derivations above are useful in setting some constant use in the simulation setup using OMNeT++ along with INET framework deployment.

6.4 Design Setup for PHMS Service Performance (Miniature Test Bed)

This section reports the detailed design for the miniature test-bed performance test. In addition, it details various metrics for measuring the performance of the research methods that this research is deploying. It also provides the setup of the simulation test on network convergence which determines the rate of service delivery to the various clients that are connected to the public health management system. The traffic is mainly from mobile devices via the network to the respective destination for appropriate responses. The diagram in Figure 6.3 shows a typical network setup to issue queries from mobile devices by sending traffic flows to and from the database (DB).

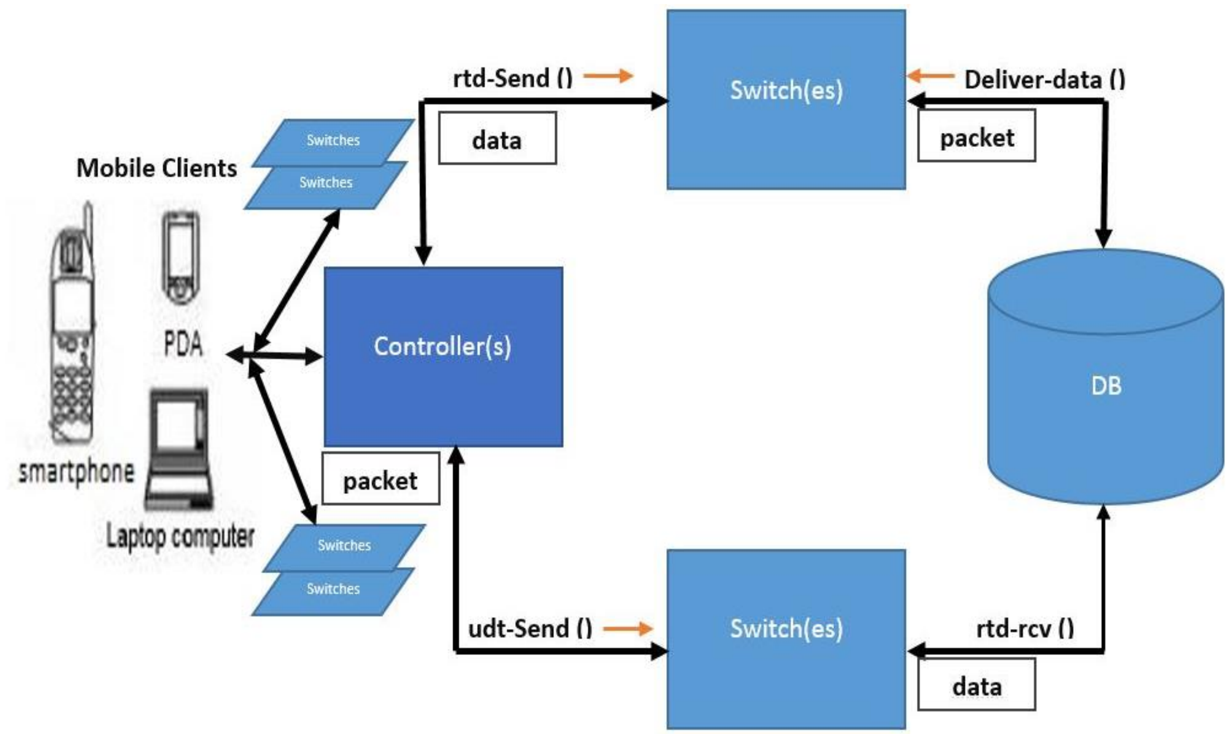


Figure 6.3: Data Performance Network Setup for Simulation and Miniature Test Bed

The simulation environment setup was divided into two sections namely the MySQL database and the mobile host devices that sent packets. The MySQL being an open-source relational database management system is a secure, scalable, and independent platform with easy-to-use features which has a proven guarantee for the safety of patient data and information. The mobile device (host) issues a query that calls a web service that calls the server (Tomcat) wherein the requested information lies. Jersey (JAX-RS) implemented Web services framework invoked by the clients.

The experiment was first conducted using 8 mobile devices (HUAWEI Ascend Y330), a Dual-core processor @ 1.3 GHz and 512M RAM each, running an Android 4.2 Jelly Bean operating system. These devices help to collect real-life data from the locations as specified in Table 6.2 and send it to where the data centre for patients is located within the University. The table gives samples of the data that were gathered from different locations. This study gathered the information in Table 6.2 in a similar fashion as used in the research of (Chotchoungchatchai S, 2020) and the details were used as the test-bed data input. The devices send the data via wireless communication through which uploads are made to the database of the framework as shown in Figure 6.3. The rate of data arrival processes from various users was represented as σ in terms of σ_1, σ_2, σ_3, and σ_4 and so on.

Table 6.2: Samples of Data from Users.

Locations	Distances	Diastole	Systole	B. Sugar
Mtuba_$_1$	45Km	>200	>180	≥110
R. Bays_$_1$	25Km	>180	>160	100-109
Ongoye_$_1$	5Km	≥156	140	90-99
Empangeni_$_1$	15Km	130	120	80-89
Mthetwa_$_1$	45Km	110	115	80
Nongoma_$_1$	40Km	100	110	75-80
Eshiwini_$_1$	10Km	96	100	75
Unizulu_$_1$	500m	95	94	65
…..	……	…..	….	……
……	……	…..	…..	…….

Using the data received, the study determined the degree of variability of the data uploaded. Initially, the issue of latency and attenuation over the distances covered in sending the messages affected the performance of the system. However, the use of hierarchical network design that incorporates the running of developed control algorithms via the SDN platform brought about performance improvement. The data arrives at the server with some given predefined rates, which on the other hand varied the utilisation of the server appreciably due to the approaches used in network designs. Mathematically, the data sent waiting time is the summation of the time it takes from the sending devices towards the receiving servers of the network. Thus, the waiting time was derived with the expressions below in equations (6.4), (6.5), and (6.6) as

$$W_q^{T\ dispout} = \sum_{i=1}^{r} W_q^{itot} \tag{6.4}$$

Hence, the total waiting time for a particular network approach while sending categories of network flows is given by

$$W_q^{itot} = W_{qdin}^{i} + W_{qst}^{i} + W_{qdout}^{i} \tag{6.5}$$

Finally, we have the total waiting summarised based on the hierarchical as:

$$W_q = \sum_{n=1}^{m} W_Q^{itot^n} \tag{5.6}$$

Where these expressions depicted the time it takes to pass a particular data from the source to the destination, the waiting time from the provider to devices is represented by W_{qdin}^{i} , while that from devices to controller is W_{qst}^{i}, and the one between controller and database is given by W_{qdout}^{i}. The mobile interface for querying requests is shown in Figures 6.4, 6.5, and 6.6 respectively. In Figure 6.4, the user tries to access a personal account in the healthcare system through login with the username and password. In Figure 6.5, the user tries to confirm his ID number to have an update on the next healthcare appointment. Figure 6.6 updated the user with the date, time, and respondent on the day of the appointment. Several other users could be sending such information simultaneously, and several different requests will be sent to the healthcare system. These interfaces typically show examples of the type of requests coming from the mobile devices that were used during these experiments. These are the interfaces from HUAWEI Ascend Y330 used in performing these experiments.

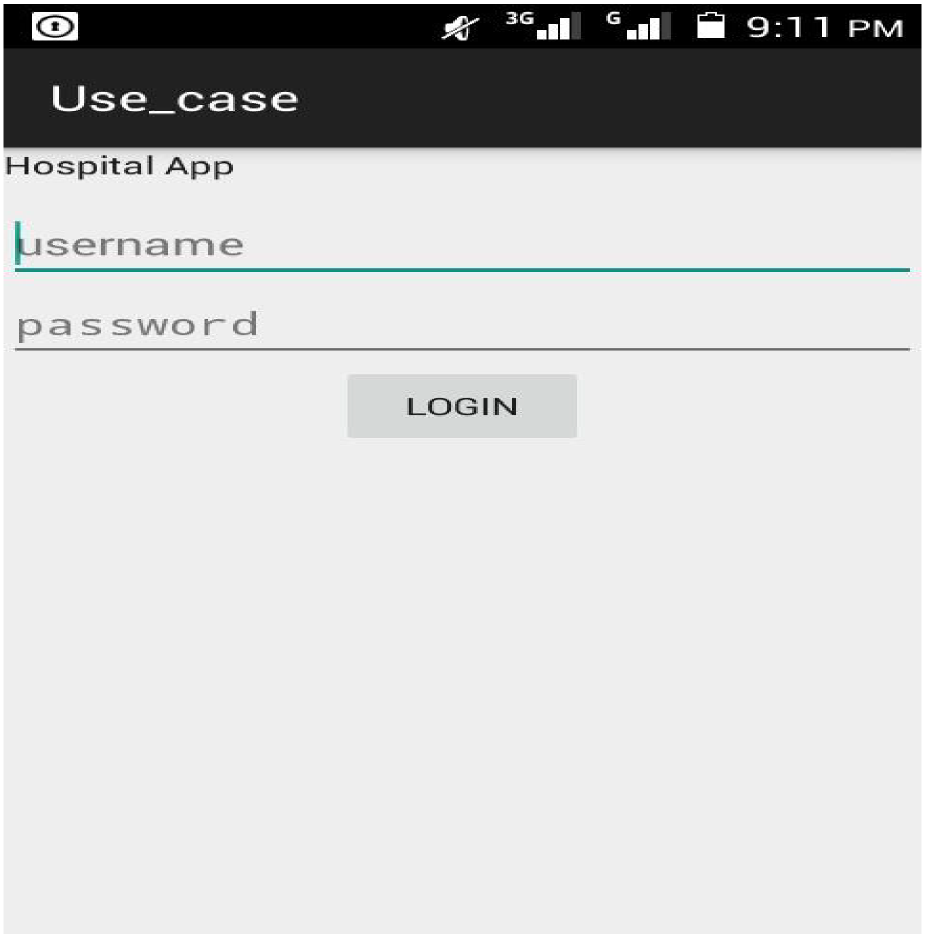


Figure 9: User's Accessing PHMS Account.

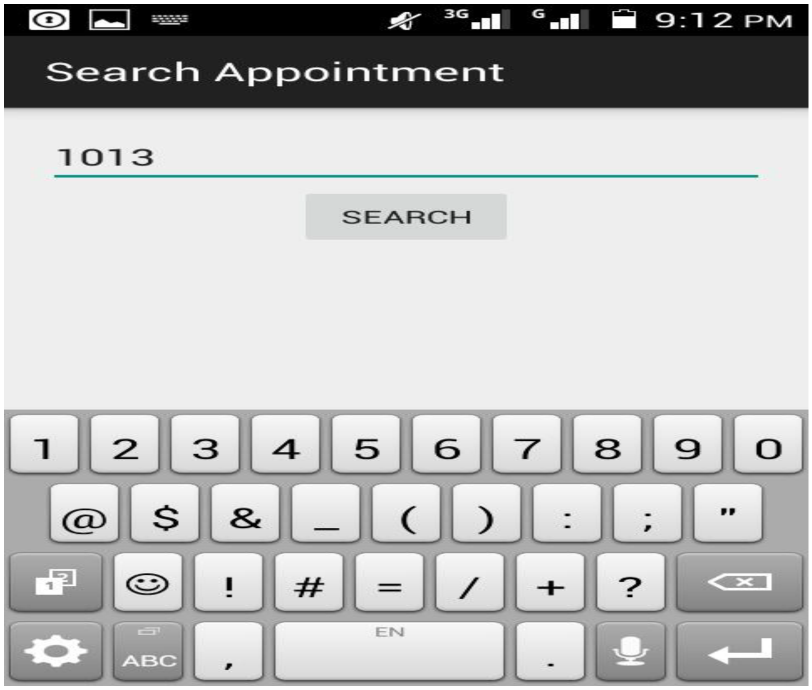


Figure 6.5: User's confirming the ID Number.

Figure 6.6: A user accessing the next health care appointment.

We later used Apache JMeter (Load Generator) for giant flows and larger flows while Wikipedia request traces (Gebrehiwot M.E, 2017) were used for small flows to populate more requests that outnumbered the limited available mobile devices to test the performance of the design for data availability and other performance testing metrics such as response time and throughput of the network. We carried out some initial tests to measure the smooth connections existing when a few requests were issued. We further carried out the test on data available when the number of requests was greatly increased, and we compared the performance of both network design approaches.

6.5 Descriptions of the Network Performance Metrics

Under this section, we itemise and define the performance metrics that are used in evaluating the network performance while deploying the DART_MCP algorithm (which uses the MOCQ approach) in a typical PHMS use-case scenario. The measurement of these metrics enable the network users to see the significance of optimal service delivery and the level of stability that is provided based on deploying DAR_MCP algorithm. In conclusion, the subsequent sections detailed more explanations on the metrics that will be measured while testing our approach with relevant research methods (simulation and miniature testbed).

6.5.1 Network Performance Metrics (Service Provider Interest)

The subsections below contained the list of metrics that are used for testing the network performance.

6.5.1.1 Rate of Bandwidth Utilization

The bandwidth utilization measures the rate at which the current traffic flow in the network utilizes the allocated bandwidth at that moment. The flow setup time serves as a metric that determines the rate of the traffic flow sent is executed successfully. The rate of the bandwidth consumption is, therefore, measured about the degree of the completion of the flow setup time for the number of traffic that was released at that time.

6.5.1.2 Increasing Size of Radix(D)

Normally, the deployment of high radix reduces the hop and the switch count, thereby decreasing the network latency and power consumption. However, the increase in the high radix comes at a cost on the infrastructure, and the test for this is supposed to be reflected on how the algorithm manages this resource such that less cost is incurred and yet high performance in terms of throughput is maintained and there is reduced latency.

6.5.1.3 Increasing Flow on Switch Failure

Several factors are responsible for the occurrence of increasing switch failure within a network environment some of which are security attacks, wrong configuration, failed software and firmware upgrade or patches, and link failure among others. The typical case under this study is the procedure that accounted for the major cause of switch failure. This implied that we are addressing the occurrence of switch failure during traffic flow interference. Ordinarily, the switch would not go down unless there is a bottleneck that overwhelmed it and forced it to malfunction. The study intended to find out the resultant impact of our approach at resolving increasing switch failure that is solely dependent on the traffic flow interference challenge.

6.5.1.4 Increasing Flow on Link Failure

The occurrence of a link failure is commonly observed in a network environment including an SDN platform. It is majorly caused by a sudden increase in the number of flows per second especially when it is above the capacity of the present network resources. The relevance of this among the metrics is based on the fact that the interested rely greatly on determining to what extent that the proposed solution affects the rate of link failure.

6.6 Comparative Selections

Significantly among the researches already discussed in the literature review section are those reported in the article by (Chen et al 2015) which contained works whose abbreviations are OQ, CQ, HCF, and CQRD respectively. The OQ (Output Queue) scheduling switches and CQ (Crosspoint-Queue) switches are architectures that were designed to increase the hardware switch capacity deployed into the data centre with no substantial mechanism to decrease the challenge of traffic flows as a result of interference issues. The state-of-the-art HCF (Hashed Credits Fair) addresses traffic flow challenges through a fine-grained queue management scheme by assigning both low and high-priority queues. Closely related to the HCF is the CQRD (Crosspoint Queue with Random-Drop), which separated a particular queue and assigned it to each pair of input and output ports to favour dropping of a packet when the Crosspoint is full. This study proposed the DART_MCP algorithm which deployed the Multi-Objective Optimization of Crosspoint Queue management approach (MOCQ). The approach manages the buffer Queue system using a fine-grained adaptive rendering approach. For the sake of clear analysis and avoidance of complexity, results from OQ were not included in this performance evaluation since its behaviours are almost similar to CQ. This chapter, therefore, reported the performance of the proposed MOCQ for three other works, which are CQRD, HCF, and CQ in the subsequent experiments below. These similar selections would also be adopted in Chapter 7 while determining the level of stability via the stability performance metrics that are guaranteed using the MOCQ approach. Conclusively, the following experiments were carried out to determine the relative satisfaction derived by service providers in deploying the MOCQ approach that was reported in Chapter 3.

6.6.1 Experiments on Service Provider

The service providers are always cautious of the cost and try to maximize the use of the resources at their disposal within the network. The study carried out the following experiments to determine the satisfaction that the provider enjoyed while making use of the resources to meet the required need of the end-users.

6.6.1.1 Experiment I: FST Vs Bandwidth Utilization

The study tested the degree of variability in the FST using real-life data between SDN-enabled database networks. Based on this result, further experiments were carried out to determine other subsections' results for the performance modelling. The master controller acts as the server (Master Controller) within the network environment whose bandwidth utilization is measured at what rate in which flows set up are completed (ε) as shown in Figure 6.7.

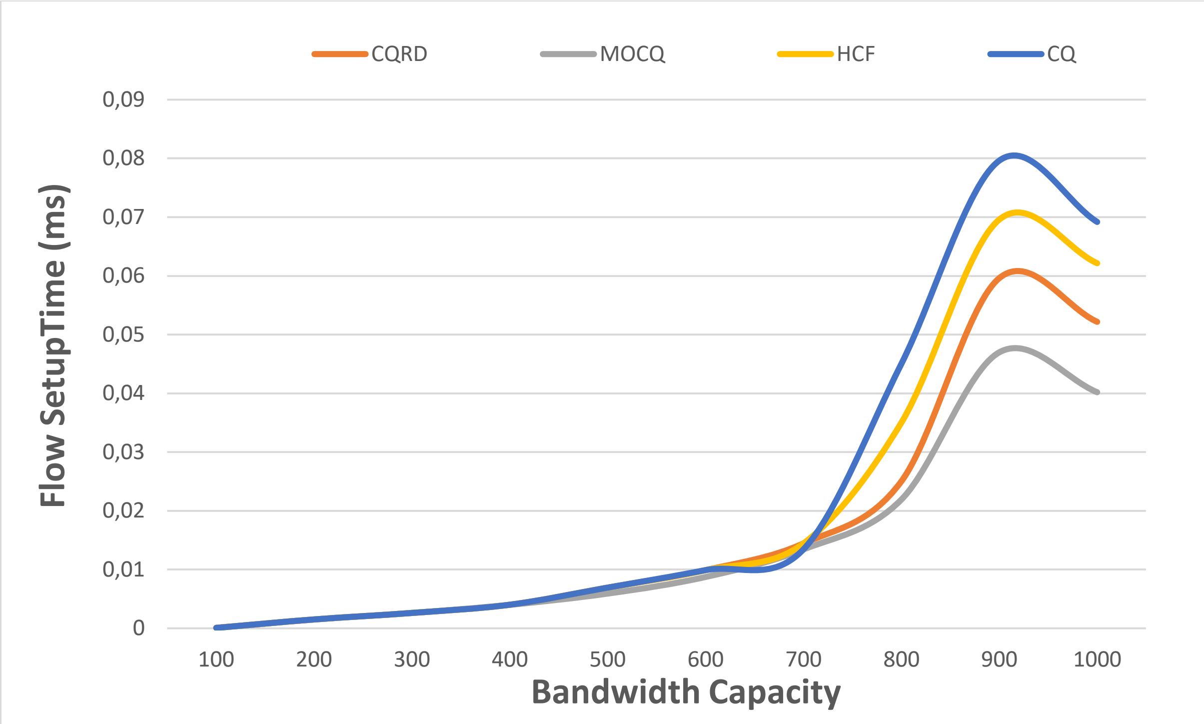


Figure 10: Flow Setup Time against the rate of bandwidth utilization

The FST performance was similar when the initial number of flows was few because the degree of variability is hardly noticed as the rate of utilisation tends towards Zero. However, the coefficient of variation between the four approaches is very small and less than unity at this point. Considering the rate of utilization of the bandwidth, which increases at a linear progression, as against the FST in milliseconds, the MOCQ utilized lesser time in setting up the flows of corresponding bandwidth in comparison to the other deployed approach. Precisely, at the rate of utilization of 900, there is a drastic shift in the FST, which is higher for other approaches because the capacity of the network can no longer bear the increasing load, and this is depicted by the increased FST before which the system begins to drop flows. The sudden divergence experienced was due to the fact that the traffic flow has exceeded the capacity of the network elements.

6.6.1.2 Experiment II: Increasing Size of Radix(D)

This study intended to find out the performance of the various approaches concerning the increasing number of flows and increasing switch hardware size in terms of radix. This is further related to the amount of latency incurred by each of the various approaches to determine their performances. As both radix size and the number of flows increases, the diagram in figure 6.8 depicts the scattered plot of the performance of each approach. Normally, when the radix size of a switch is increased, it provides an avenue to increase the number of switch ports in the hardware thereby opening more opportunities for the flows to pass through without challenges and this is supposed to result in better performance of the system. However, considering the increasing traffic flows correspondingly, it poses some challenges to the system as well in terms of flow interference along the channel. The multi-objective optimization of the Crosspoint queue approach through its analysis helps to manage the available resources adaptively to alleviate the interference problem. The effect of the solution results in reduced network latency in comparison to the other approaches. The MOCQ which is represented by the blue bubbles were more crowded at the least possible latencies than any other with increasing traffic flow and switch radix sizes. The significance of this is the fact that MOCQ enables the service provider to have maximum utilisation of the available radix sizes in the system without investing in the unnecessary cost of the switch hardware. While other approaches were already reporting higher latencies, the proposed approach still maintained lower latencies with the same resources that were made available.

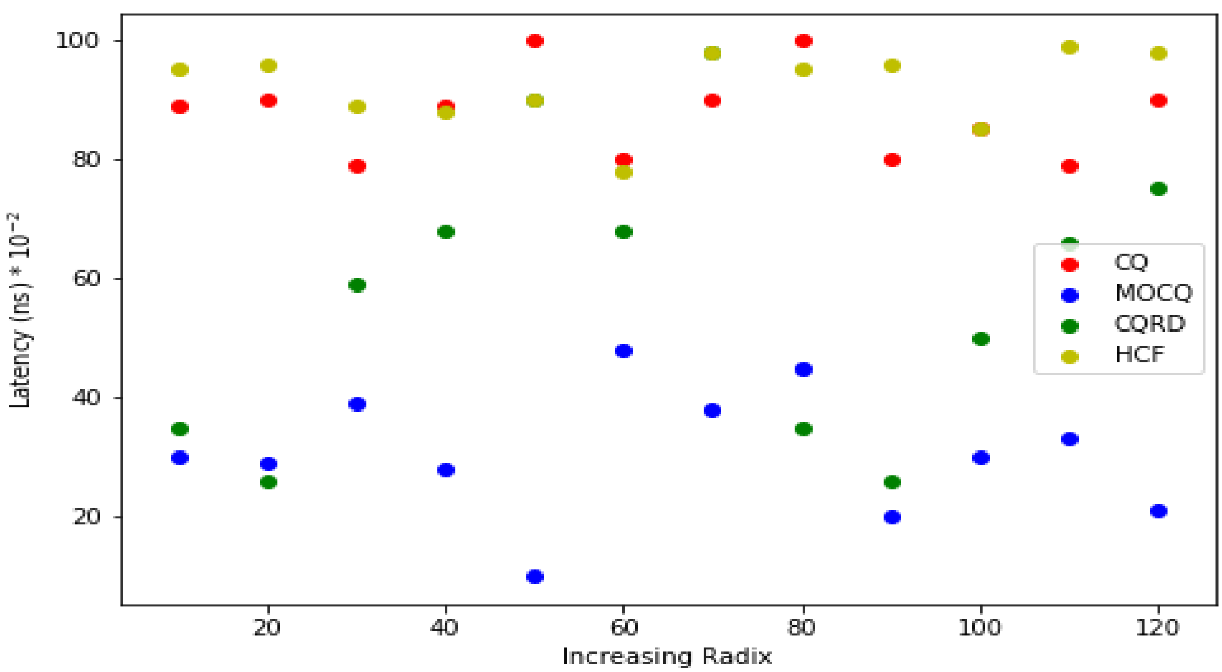


Figure 6.8: Effects Increasing Flow and Radix on Network Latency

6.6.1.3 Experiment III: Increasing Flow on Switch Failure

Considering the hierarchical order of the network setup, we try to decipher the effect of our MOCQ on the network by measuring the number of switch failures incurred in the same time interval. The arrangement of the switches in an abstracted hierarchical routing manner allowed easy resolution of stretching path problems by utilizing intra-switch links and all possible pre-calculated inter-switch links for traffic flow. In comparing our proposed MOCQ approach and the other three, the maximum number of controller flows is kept at around 60k (64,768) due to the various approaches adhering to their complex default structure and design (Chen et al 2015). The performance is noticed to have improved with the MOCQ approach through a reduction in the amount of unsuccessful traffic flow to the destinations which are caused by unresponsive switch devices. In Figure 6.9, the percentage of switch failure is on the average of 0.25% which is equivalent to an over 150% improvement over the average performance of the existing CQ approach. Moreover, our approach ensures the provision of the shortest path through the enabling algorithm, to increase the stretch value of the buffer size to the maximum capacity utilisation and avoid the sub-optimal use of the available resources, thereby predisposing others to more switch failures.

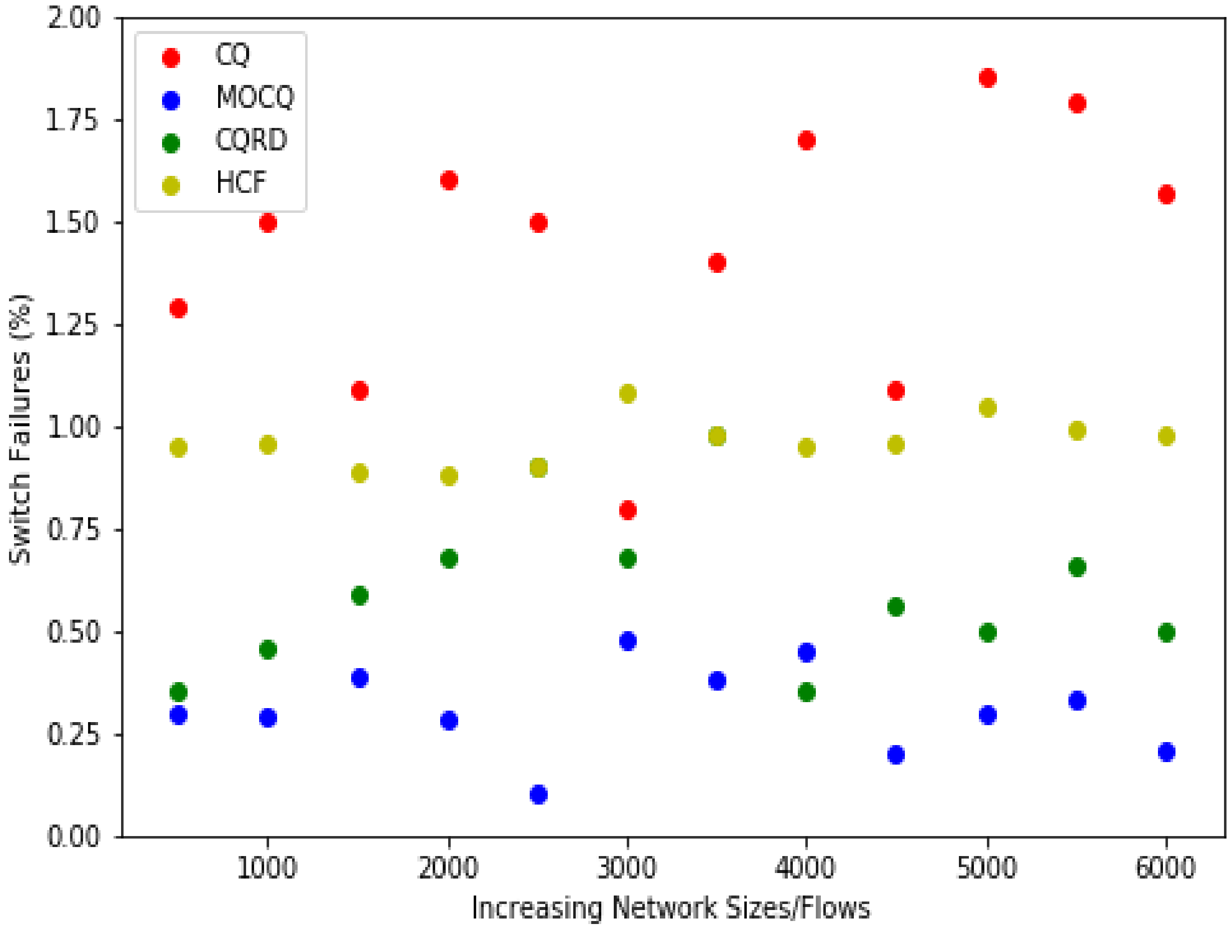


Figure 6.9: Effects of increasing Flow on Switch Failure

6.6.1.4 Experiment IV: Increasing Flow on Link Failure

Lastly, under the section of the service provider performance, we carried out a link failure performance test for the approaches and Figure 6.10 depicts the performance of MOCQ and others in the experiment. It is obvious that as the network size increases, the link failure also increases. However, the MOCQ approach proved to be a better option at managing the packet loss due to link failure. When there are limited values of network sizes that are propagating flows, the CQ and HCF already recorded higher switch and link failures before the corresponding CQRD and MOCQ approaches. This implies that the CQ and HCF are more prone to suffer from network instability due to their inability to effectively handle increased or heavy traffic flow coming from the network hosts. It is reasonable to conclude that even though there is an increase in the switch and link failures as the network sizes increase, the trend of MOCQ based network setup gives a better record of traffic flow performance. One of the major goals of performing the switch and link failure experiments was to determine the stability function of the network approaches whose degree gives the measure of the performance of the various approaches considered during these experiments. The details of these evaluations are analysed in Section 6.7 under the discussions of results.

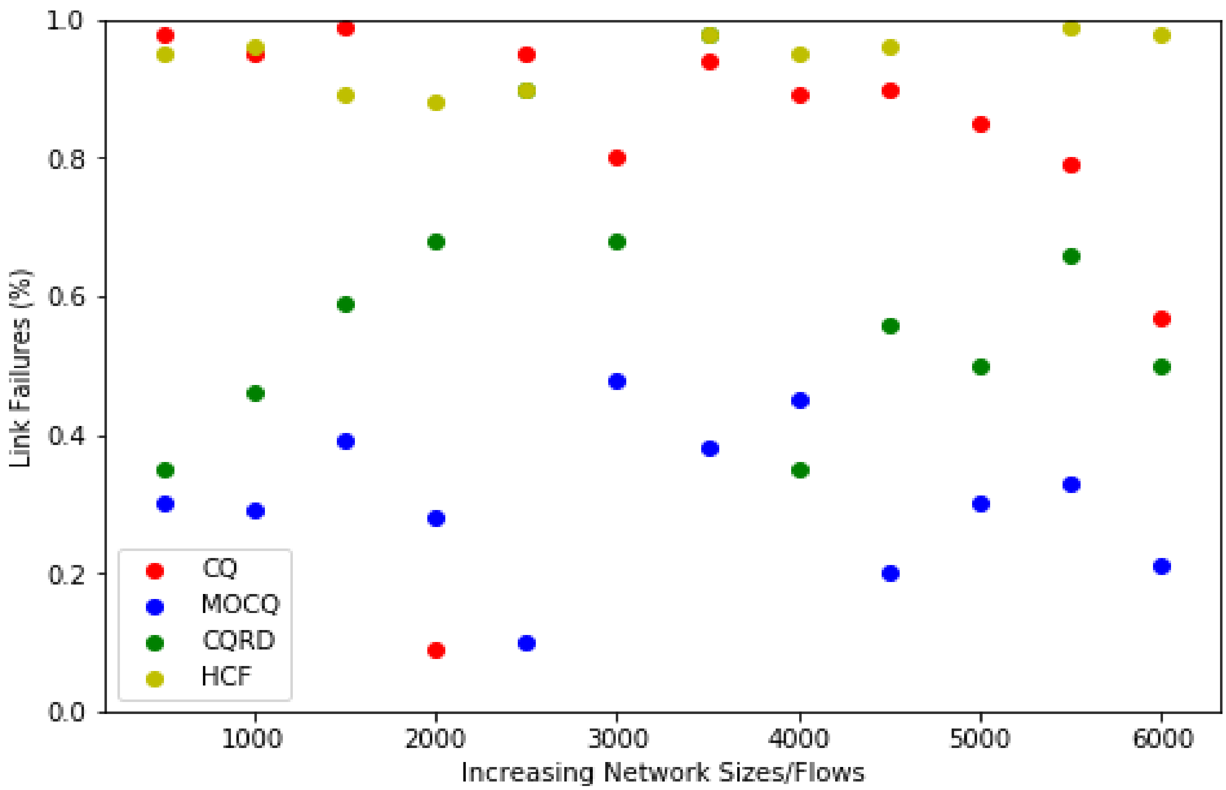


Figure 6.10: Effects of increasing Flow on Link Failure

6.6.1.5 Experiment V: Simulation Against Miniature Test Bed

The stability function attained by each of the research methods is used to validate the adaptive rendering approach that we proposed in this thesis. The validation of the approach used in this research is very important as we are the first to investigate the deployment of ART in solving network flow management problems. Based on this very important reference, we try to determine how the simulation of MOCQ performs in the corresponding comparison with the miniature testbed method as the number of network hosts and clients increases. We are validating this in addition to the resistance constraint determination that we validated on numerical analysis and network simulation earlier on in Chapter 5 under Section 5.3.2. The result of the validation is provided in Figure 6.11 which compares the Simulation with Test-Bed methods.

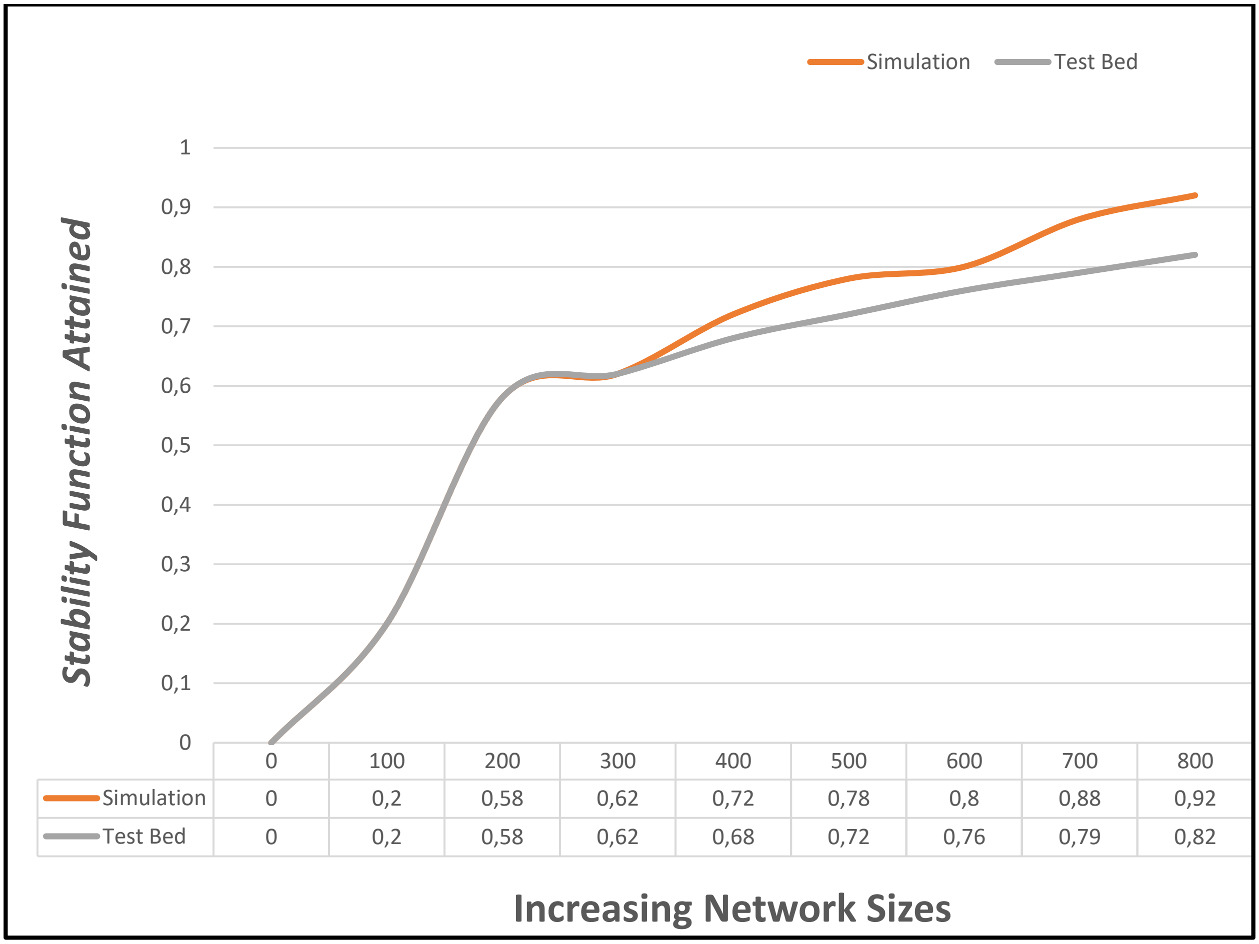


Figure 6.11: Comparison of Simulation and Test Bed Stability Function Results

We called it a miniature testbed because we cannot design many more physical flows to the tunes of 1000s and more. Hence, we deployed the use of small flows from Wikipedia request traces and Apache Jmeter (for larger and giant flows) to increase the influx of network flow into the network. The results in Figure 6.11 depict similar performance for both research methods used. However, as the number of service requests reached around 300 requests, the stability function of both approaches changes slightly with some degree of deviation. Due to the kind of storage used in the test bed, it became challenging for the network to scale proportionately thus leading to increasing stability function.

However, we understood from literature that the lower the stability function, the more unstable the network is and the higher it is, the more stable the network becomes (Akinola et al 2020b). Furthermore, we concluded that in this experiment, there are some assumptions in the testbed and probably some errors which did not lead to similar results. Therefore, from 400 requests upwards, we could see the linearity of the similar performance in the methods and below 400, both were able to validate appropriately the performance of MOCQ in alleviating the effect of network flow interference within a DCN system.

6.7 Discussion of Results

Bandwidth utilization is one of the important metrics for the service provider. This is because of its cost and if it is not well managed, it results in dissatisfaction and loss of profit. The proposed MOCQ approach has found its relevance in these areas in that its mechanism provided the avenue to ensure that an optimized output is derived from its deployment. Every traffic flow needed some level of bandwidth to run smoothly and usually, they often contend together for the use of the resources. Hence, every mechanism that would be in place must be such that it reduces the Flow Setup time thereby enabling the quick release of the resources for other tasks to make use of it.

The lack of an adequate and reliable mechanism will not only affect the performance of the system but also inhibit the sustenance of the end users' subscriptions. Thus, reducing the FST has a significant effect on the duration used in holding down the bandwidth to complete the ongoing task. The multiple mapping releases the failed link as soon as the traffic flow fails thereby releasing the utilised bandwidth for the subsequent tasks as fast as possible. Therefore, maintaining a balance of flow set-up time for the providers' and end users' interest at this junction becomes an integral contribution of the chapter to this thesis. This network performance metrics tested under this chapter showed a measure of improvement over the existing approaches.

The multi-hop routing which was not adequately addressed by the other approaches introduces the occurrence of flow interference on the intermediate switches, and this leads to performance degradation. Several approaches have proposed the increment of Radix sizes which often enables the provision of increasing switch port numbers to address the problem. This is believed to enable the incorporation of more flows in the network and in the process ensuring the network remains stable. However, this comes at some cost to the service provider. Generally, the service provider is always interested in maximizing the use of the resources already procured. The MOCQ approach from the earlier experiment was able to use the same radix size to achieve both high latencies and reduced latency in the system. This implied that the multiple mapping and quick release of bandwidth while a link is down enabled the reduction in the delayed latencies incurred in the system and resulted in some level of satisfaction to the service provider.

Furthermore, whilst keeping the controller capacity constant, the experiments tried to determine the average percentage of switch failures incurred by the various approaches. The failure of the switch while all other factors are relatively available in this kind of setup is usually a result of bottlenecks via flow interruptions. This has some direct relation to the link failure as well although the paramount goal is to ensure that the switch failure is reduced to the minimum by ensuring that the link failure is eradicated, thus enabling the stability of the network. The experiments have proved that although the MOCQ approach does not outrightly eradicate the possibility of switch failure that is caused by link failure, the study was able to maximize the utilization of the available network resources to achieve maximum output.

6.8 Chapter Summary

This chapter presents the applicability of our proposed approach to alleviating the occurrence of traffic flow interference in a typical SDN-based DCN. We designed a hierarchical model network setup for the data centre which typically enhances the performance of the separation of the control plane from the data plane. We expressed a detailed setup for the implementation of the network environment for PHMs on an OMNeT++ network simulator. We also depicted the testbed analysis where students are allowed to physically use devices to test the performance of the network system. We evaluated the performance of the Multi-Objective Optimization Cross-Point Queue MOCQ approach especially in comparison with other contending approaches in the field. We identified the impact of network interference on the combination of user requirements which invariably reduced the success rate and the running time of the network. The failure rate (switch and link) directly relates to the throughput and the latency experienced in the course of the network operations, therefore, necessitating their optimization.

Furthermore, we deployed the network performance metrics majorly in this chapter to test for the relative satisfaction of the service provider. These explanations detailed the various mechanisms and methods that were adopted in capturing the required data for determining the effectiveness of the network environment as well as the level of stability that is guaranteed concerning the network service provider. In the next chapter, however, we conducted the stability performance metrics on the described system to test the end-user satisfaction level. The next chapter determines to which degree of stability level is guaranteed to the end users.

Chapter 7

ANALYSIS USING STABILITY PERFORMANCE METRICS

7.1 Introduction

Several network metrics have been used to test for network performance as highlighted in Chapter 6 to determine the level of satisfaction that the service providers could guarantee while adopting the DART_MCP algorithm that deploys the MOCQ approach. Similarly, the study would be incomplete without determining through the stability performance metrics, the level of stability that is guaranteed to the end users as they expect maximum turnover that is equivalent to the cost of incurring the use of the network platform. The reason for this is that the impact and the importance of an optimized PHMS cannot be over-emphasized in a fast-growing geographical location for dependability and stability purposes. As an inference that has been established in literature, this chapter intends to compare the stability performance derived from the MOCQ approach with some of the works in literature that were proposed on the control of the network flows through algorithms thereby managing the control plane.

Hence, this chapter focuses on the expectations of the end-user in terms of the performance evaluation (as this has a relative significance in the demand and supply chain) on the DART_MCP algorithm which deploys the MOCQ for the alleviation of the traffic flow interference problem. This chapter starts with Section 7.2 which describes the stability performance metrics used in the performance evaluation while Section 7.3 justifies the reasons for the selections made for comparison. Section 7.3.1 describes the experiments to test the benefits of the end users while Section 7.3.2 discusses the experiments on the end users. Section 7.4 discusses the results from this chapter with their significance while Section 7.5 summarises the chapter.

7.2 Comparative Selections

The study deployed similar comparative selections which were used in Section 6.6 of Chapter 6. These were mentioned earlier in an article by Chen et al (2015) in our literature review which contained researches with these abbreviations: OQ, CQ, HCF, and CQRD. On the one hand, closely related to the HCF is the CQRD (Crosspoint Queue with Random-Drop), which separated a particular queue and assigned it to each pair of input and output ports to favour dropping of a packet when the Crosspoint is full. On the other hand, the DART_MCP algorithm deployed the Multi-Objective Optimization of Crosspoint Queue management approach (MOCQ). The approach managed the buffer Queue system using a fine-grained adaptive approach via active and reactive modes of ART. This section continued the evaluation with the end users using the performance metrics in the following section.

7.3 Descriptions of the Stability Performance Metrics

This section itemises and defines the performance metrics that are used in evaluating the level of stability while deploying the DART_MCP algorithm (which uses the MOCQ approach) in a typical PHMS use-case scenario. This is very significant to the end-user to understand precisely the level of stability performance from a particular data centre. This helps to determine the decision either to keep subscribing to or consider other service providers to save costs. It is not enough as an end user to rely solely on the information provided by the service provider alone; these stability tests, therefore, give the confidence needed by the end user to adhere to the services that are rendered by the provider.

7.3.1 Stability Performance Metrics (End-User interest)

The subsections below contained the list of metrics that are used for testing the stability performance in the network setup.

7.3.1.1 Network Availability for Patient's Data Assessment

The preliminary test is to access the rate at which the network is readily available for use in the PHMS when requests are sent via it. The mobile devices send requests via the cascaded switches being controlled via the controller and routed through to the data reservoir and forthwith back to the client. This test aims to determine the rate of network availability (also directly related to network stability function) in the two research design methods (simulation and testbed) that are considered in this chapter. By data availability, we mean the number of successfully completed packets sent via the network and returned to the source of the request (Li 2019). This test deployed a random sampling technique for the number of the completed request while considering a subset S' being an element of a set X which consists of a certain number of services to be accessed, thereby testing the rate of network availability under a randomised data sample.

We consider a given set of patients' data I = $\{S_1, S_2, S_3, \ldots, S_n\}$ having a subset $I' \in S$ being accessed through some queries from mobile devices. The random sampling approach selected 30 attributes related to individual patients in the database and similar requests to these selected attributes were checked to ascertain if the query hit or not. When it was completed, we recorded a hit, and when not, we had a null hit for the query that was sent. Hence, the availability of the network is defined as the ratio of Mean time between failures to that of the sum of mean time between failure (MTBF) and mean time to repair (MTTR) expressed in equations 7.1, 7.2, and 7.3. Thus:

$$Availability = \frac{MTBF}{MTBF + MTTR} \tag{7.1}$$

Where we expressed MTBF and MTTR as:

$$MTBF = \frac{\sum_{i=1}^{n}(start\ of\ downtime - start\ of\ uptime)}{n} \tag{7.2}$$

$$MTTR = \frac{\sum_{i=1}^{n} time\ to\ restore\ to\ normal\ operation}{n} \tag{7.3}$$

where *n* equals the number of failures.

Hence, we calculated the network availability in terms of having access to the patients' data for the network.

7.3.1.2 Traffic Flow Response Time

This metric determines the time it takes for the network to respond to requested data via each of the approaches. It is measured as the time taken in sending the request and the reception of the response between the database and the client. The expression for the Average Response Time used for the evaluation of the network design is presented in equation (7.4) as

$$Average\ response\ time = \frac{1}{n}\sum_{i=1}^{n}\left(T_f - T_o\right) i \qquad (7.4)$$

Where: n represents the number of requests

T_f represents the time when the request was sent

i represents the request's index

Thus, it is possible to determine the time taken for each sample network to complete the task when several requests were issued within the network environment simultaneously.

7.3.1.3 Traffic Flow Throughput

Determining the number of transactions processed per given time provides the throughput of the network under consideration. In this experiment, the throughput was measured to determine how many requests were completely processed per second with an increasing number of clients' requests being issued. The throughput for the network designs was determined by the expression in equation (7.5) as:

$$Average\ throughput = \frac{1}{k}\sum_{i=1}^{k} ET \qquad (7.5)$$

Where *k*, in this case, represents the number of requests at a particular session and *i* represents the request index that occurs over time to perform an execution (E) over many request sessions. Just like the response time and network availability, Apache Jmeter (Load generator) was used to simulate more requests to test the network stability performance.

7.3.1.4 Stability Function

We defined the stability function (SF) of the network design as the ratio of the rate of link failures to that of the switch failures expressed in equation 7.6 as percentage thus:

$$SF\% = \frac{rate\ of\ \Delta\ in\ link\ failures}{rate\ of\ \Delta\ in\ switch\ failures} \times 100 \qquad (7.6)$$

Hence, it is possible to calculate the stability function for our proposed approach to ascertain the level of stability that could be guaranteed in the network as well as other adopted approaches from other authors.

7.3.1.5 Rate of Network Convergence

It is important to understand the fact that the performance concerning the stability of a network is closely related to and directly proportional to the rate at which a particular network scenario attains the network convergence state (Jude 2021). In SDN, the convergence state is reached when the routing table of the controller possesses an updated record of all the shortest routes to all destination nodes (Sudheera 2019). This is different from the conventional network setup where the network convergence is expressed as the state of achieving similar routing information on the routers and as well as possessing all the connecting nodes' shortest accessible routes.

Thus, determining the rate at which some specified controllers were able to attain an updated routing table in comparison to the best approach within our comparative selections would be vital to the end users' satisfaction. However, the process of determining the convergence of a typical network does not have a standard model as a general rule from literature. Different methods were used in Sudheera et al (2019) and Li (2019) but we choose to deploy the approach used in the latter based on the fact that we can easily access the time it takes a network to recover when smooth continuous ping messages are interrupted at one end of the network.

Using the OMNeT++ simulator, it is possible to design a network size that depicts the four types of network approaches we are looking at with nodes ranging from 1 to 200 nodes over several network links from $x-1$ to $\left\lfloor\frac{x \times (x-1)}{2}\right\rfloor$. In addition, the parameters in Table 7.1 were used to configure the system for proper evaluation of the network convergence. It is important to understand that several approaches like the random drop (CQRD) and credits hash have proven to work efficiently. However, due to the complexity, the comparison was made to determine the efficiency of the random drop with the proposed MOCQ approach in this work (Li 2019).

Table 7.1: Simulation Environment for Network Convergence

Parameter	Symbols	Values
Network link range	NLR	10Mbps
Flow table update delay	D_{FLU}	10 ms
Bandwidth of links btw C & S	linkBW	10^8 bps
Controller load factor	CLF	normrnd(1,1)
Forwarding table update delay	d_{FOU}	10ms
Instruction per second	IPS	2^{15} instruction/sec
Congestion factor	c	normrnd(1,1)
Detection delay	d_d	1 ms
Link Delay	D_L	1 ms
Network depth at the fault location	f	$Log_2(n)$

As mentioned earlier, Algorithm 2 in the appendix section was edited to suit the measurement that is to be carried out in this chapter although the idea was mined from the approach adopted in (Li 2019). This proposed convergent algorithm was able to measure the rate at which the network traffic flow was able to reach stability/convergence. This algorithm was deployed to test the performance of the corresponding CQRD that is run on the SDN data centre network. The proposed algorithm was depicted in Algorithm 2 on the appendix page.

7.3.2 Experiments on Stability Performance Metrics

The experiments under this section expressed the evaluations of the end-user benefits and satisfaction on the performance of our proposed approach. These experiments were carried out on

Network Availability, traffic flow response time, traffic flow throughput, the stability function derivations, and the rate of network convergence among the approaches as they were explained in Section 7.3.1. The following section depicts the result analysis of these results.

7.3.2.1 Experiment I: Network Availability

The performance of the PHMS was tested for Network availability while performing some basic clients' request services like login of details, employees' profile and history, inputting patient records, booking appointments among others as depicted in Figure 27-29. The experiments were set up to evaluate how readily available the network connection to the data storage is to carry out basic healthcare services amongst the clients and the network servers. The login service was first accessed to determine the response of the networks and due to the relatively simple matching processes involved, an availability record of 100% was recorded against all the other approaches that were evaluated. This depicted that when a less computational process is carried out, little or no effect is felt concerning the approach deployed. However, when carrying out other health care activities that involve more computational processes and matching of data, it was discovered that the availability of the network to carry out the requested tasks varied with the MOCQ based approach. This is provided that on average, a better network availability option is available among other options as depicted in Figure 7.1.

Under some conditions, we realised that the client requests often involve initiating one or more multiple requests. Unlike the login request which only confirms the authenticity of a username and password that was submitted, it does not initiate more than one other request, else this accounted for why the login service was readily available for all the four approaches that were tested for availability. However, other requests like diagnosis, patients' details with history, and the discharging services initiated many other requests which sometimes rendered the network temporarily unavailable due to the influx of flows within the network environment. This, accounted for the inefficiency that was recorded against some other approaches in these experiments.

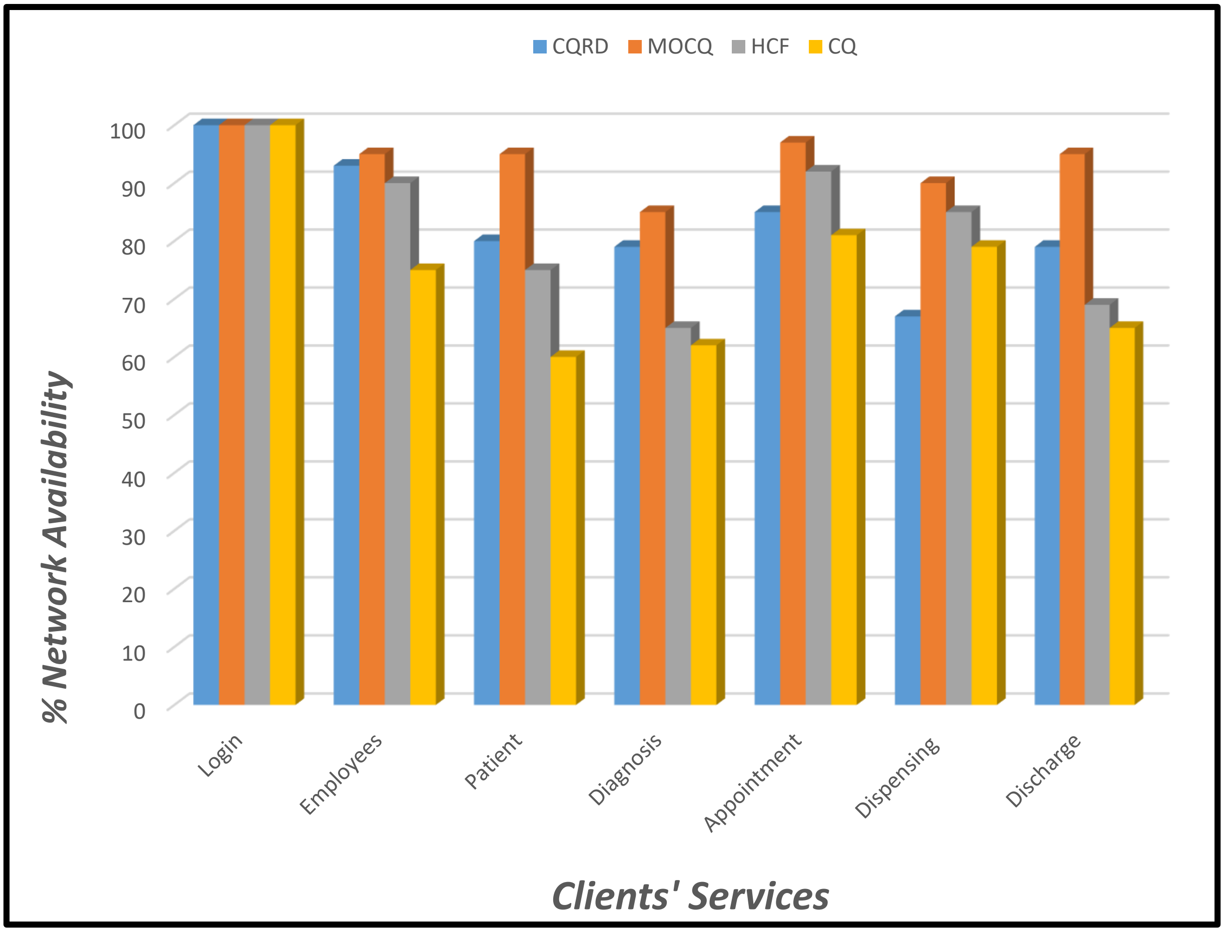


Figure 7.1: Network Availability for the approaches

7.3.2.2 Experiment II: Traffic Flow Response Time

The next experiment showed the response time used to complete each specified task, with an increasing number of requests as depicted in Figure 7.2. The figure showed the performance of the MOCQ with other approaches. At every instance, the MOCQ approach took less time in milliseconds to retrieve the query sent by the devices compared to the other corresponding approaches. This can be explained by the fact that the MOCQ often bypasses the occurrence of network stretch problem that takes a longer route through the system that adapts to the current situational need. The local network routing is restricted only to the premises as controlled by the control plane rather than going via other components when it is supposed to be routed to another network. The other approaches route the packets outside their domain thus taking a long time before the requested feedback is returned. Similarly, the result from the throughput for the approaches also follows a similar output as the number of data that were successfully routed from one node to the other over a given period varies for different approaches. The HCF depicted some fluctuating response time curves because of the harsh credits that were allocated for buffer allocations when it is full. Due to premeditated assignments, it can quickly normalise some flows while others are not easily assigned.

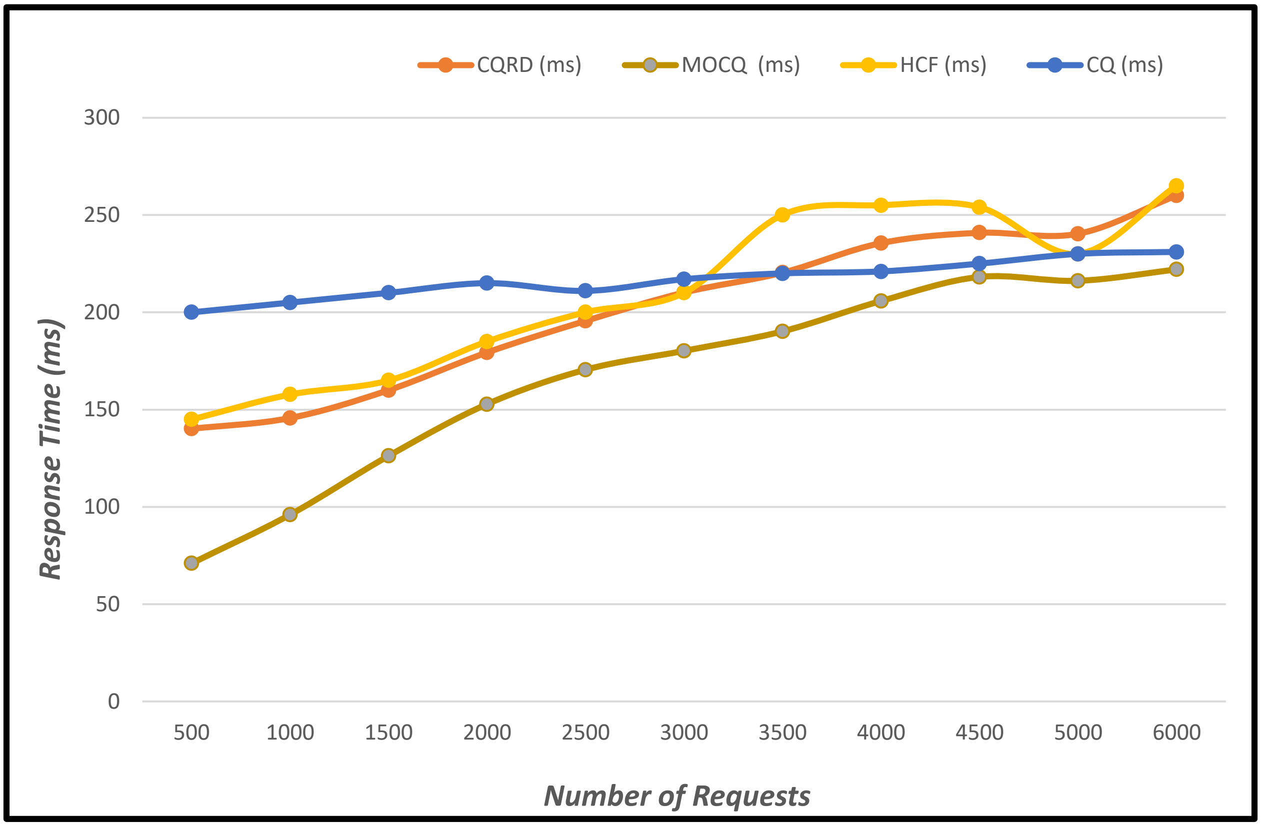


Figure 7.2: Response Time for the approaches

7.3.2.3 Experiment III: Traffic Flow Throughput

Figure 7.3 depicts the graphical representation of the performances for the network throughput for the tested approaches. As the number of service requests increases, there are variations in the performance of the approaches tested. The closeness at the mid of the graph typified the fact that when an average number of the request was made with the approaches, their performances come close to one another, although it is not the same. However, when there are fewer requests which could just be within their domain, their throughput often performs better than when there is a larger request which obviously will be both inside and outside the domain and could result in network flow interference. The adaptive nature of the MOCQ played a vital role over the existing approaches, thus giving relatively higher performance than the others. Moreover, the multiple mapping significantly ensured that service requests were attached to more than one controller to ensure high throughput in the long run.

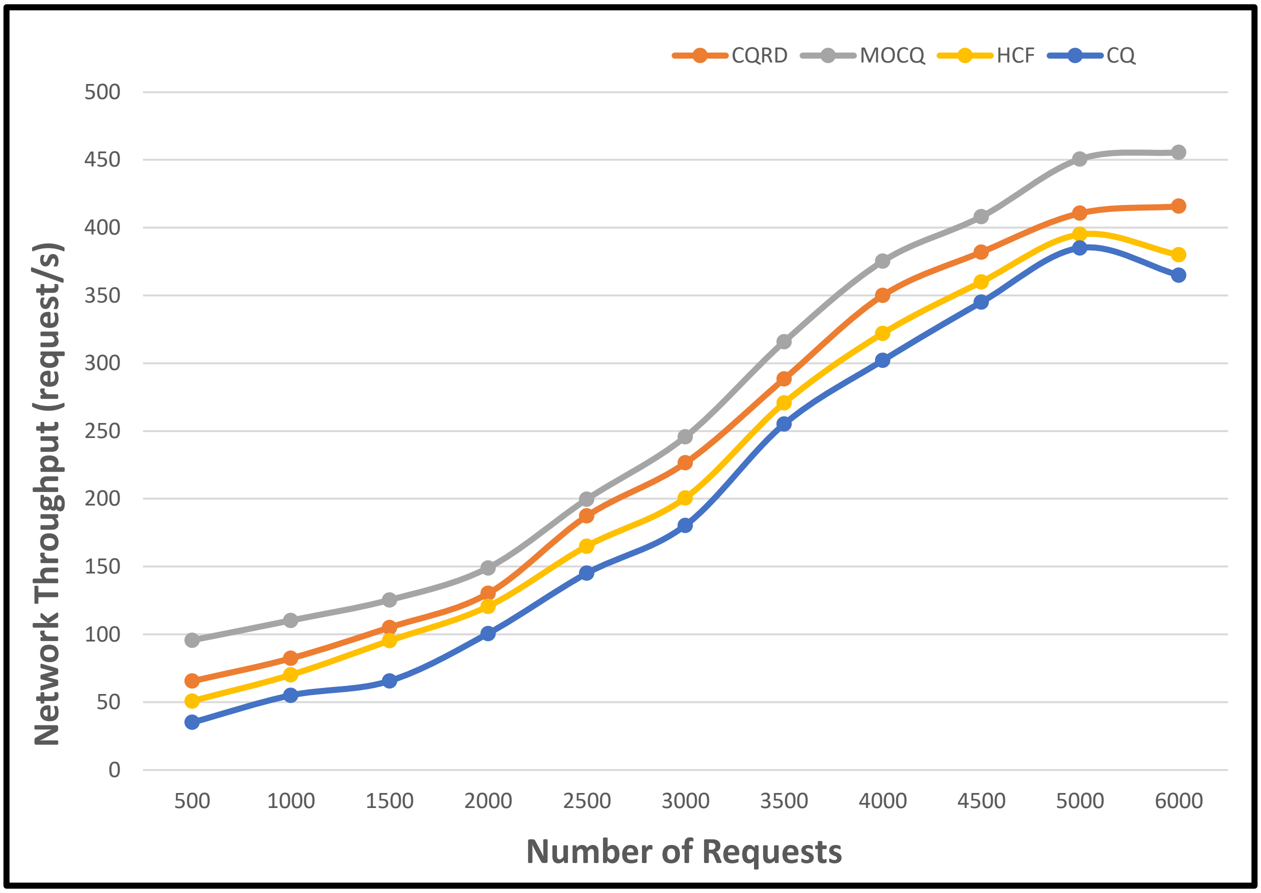


Figure 7.3: The throughput for the approaches

7.3.2.4 Experiment IV: Stability Function

The result of the stability function for all the approaches under consideration is very important especially for end users' decision making. This section compares the performance of all the approaches as we increased the network sizes from 500 to 3000 nodes. At each level of network size, the stability function is a measure of how reliable the network is to avoid the possible or the tendency to be prone to failures. This implied that the higher the value of the stability function of a network, the better is its ability to prevent failure. From Figure 7.4, the MOCQ was able to attain a higher stability level than any of the other approaches that were compared. Significantly in sizes 1500 and 2000, the network flows are more of the light-sensitive flows/small flows, hence the convincing higher stability level that was attained for MOCQ. However, on the other approaches that were shown, the performance was a bit lower as compared to the proposed approach in this study. The hierarchical setup and multiple mapping of MOCQ were responsible for its better performance.

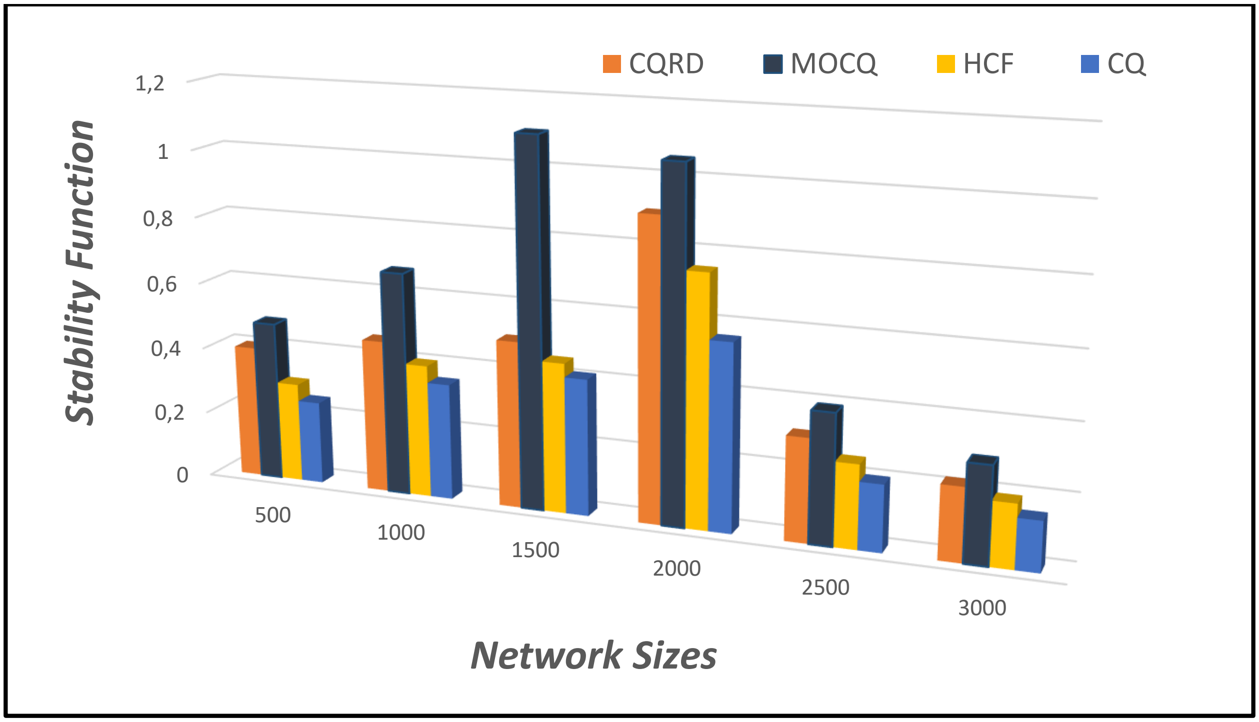


Figure 7.4: The Stability Function Performances of the approaches

7.3.2.5 Experiment V: Rate of Convergence

This section determined the rate of convergence for each of the four approaches and vividly showed under this section that both CORD and MOCQ performed well above other ones. The INET framework application which was used in Calvert (1997) was also applied to produce the required network topology. The performance was tested using the number of success rates achieved and the duration of time at which the algorithm performed the simulation (running time). We considered the performance of the MOCQ with one competitive algorithm from the earlier mentioned algorithms in terms of CORD. This section also evaluated the performances of the algorithm when there is an increase in the number of traffic flows while the flow size was kept constant, as well as when the flow size was varied with a constant traffic flow rate within the network.

The diagrams in Figures 7.5 and 7.6 compare the performance of both MOCQ and CQRD algorithms especially in terms of the duration of delays before a stable state is reached by the algorithms. A total of 10 controllers and 1500 switches were used to investigate the impact of network interference on the stability that was attained in traffic flows. The delays were measured in milliseconds while the progressive results were derived in sending flows.

The figures clearly showed that for network nodes of 1500 running with 10 controllers, it took CQRD approximately 245 ms to attain network stability at the convergence point while MOCQ in Figure 7.5 only used 140 ms. The differences between these two values occurred as a result of network interferences as earlier highlighted. Thus, a vast difference of 105ms existed in the performance of the two algorithms when compared. It is noteworthy to understand that at each interval, the number of delays could be calculated to determine the corresponding values under similar conditions. The remaining results for other approaches are depicted in Table 7.2. In a case where we are short of network resources, especially controllers and at the same time, trying to accommodate more network hosts, a high stability routing protocol will be of importance as it can help to reduce, if not totally alleviate the effect of network interferences. Based on these four tested approaches, MOCQ provided us with the best network stability as it caused a 42.9% reduction in the network delays as depicted by the experiments.

Table 7.2: Comparison of Network Stability Delays

SDN resources		Algorithm Delays			
Switches	***Controllers***	***CQ***	***HCF***	***CQRD***	***MOCQ***
300	2	128	88	47	26
600	4	255	176	96	55
900	6	388	267	145	83
1200	8	518	356	194	111
1500	10	650	450	245	140

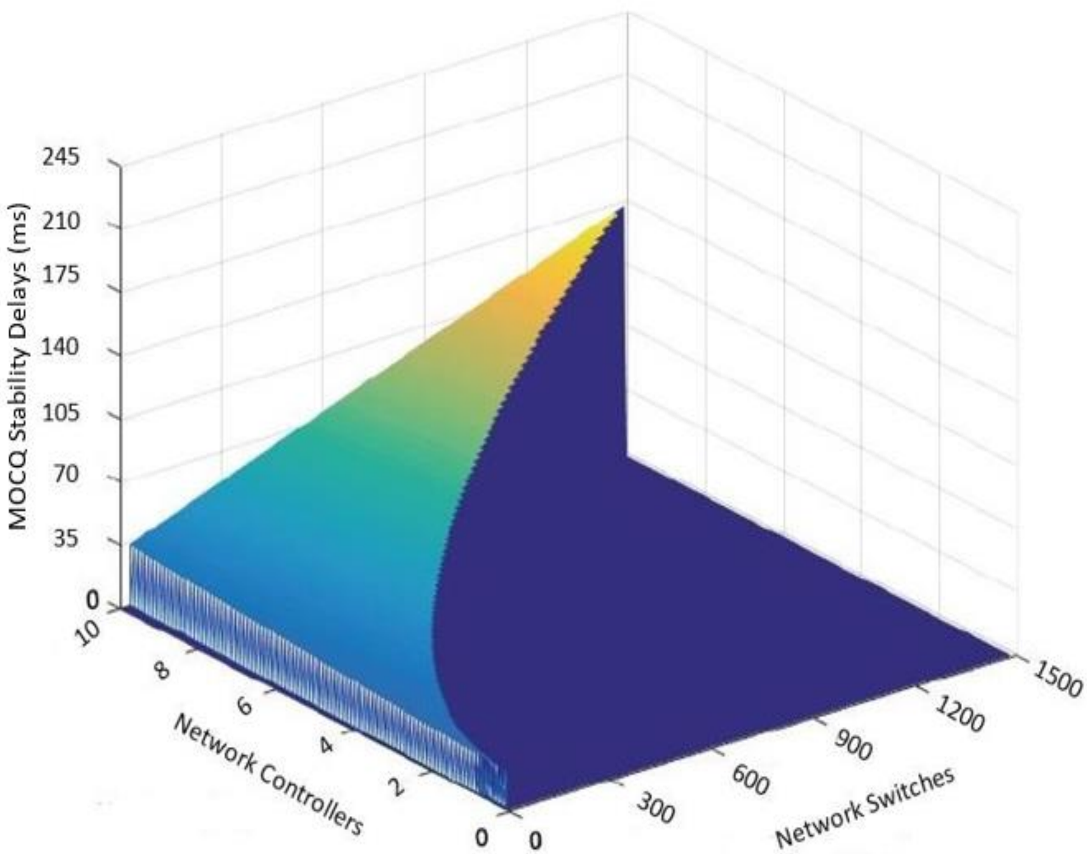


Figure 7.5: MOCQ Delay before attaining network convergence

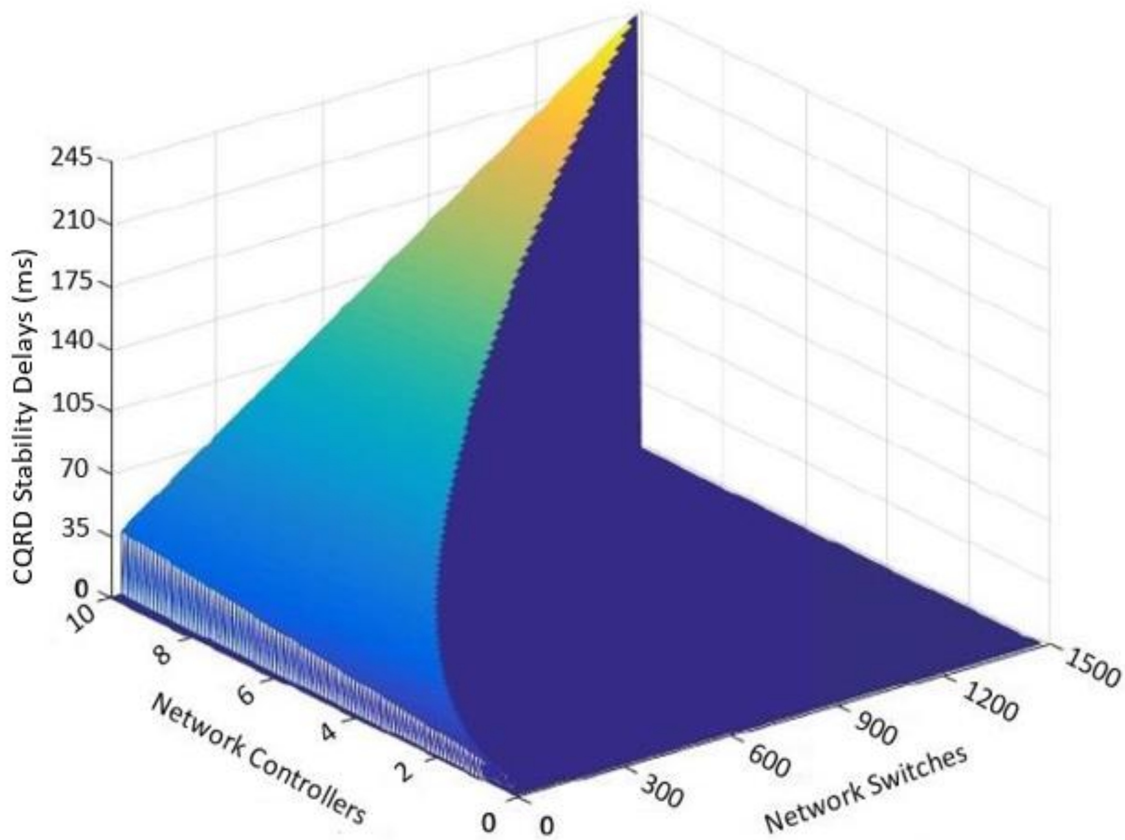


Figure 7.6: CQRD Delay before attaining network convergence

7.4 Discussion of Results

The chapter has been written to balance up the stakeholders' (service providers and end-users) interests as one of the objectives and contributions of this thesis. Earlier in chapter 6, this study dealt with the perspective of the service providers' satisfaction which utilises the network performance metrics to evaluate the proposed system. The stability performance metrics have been used in this chapter to determine the level of satisfaction that the end-users derived. Significant among these metrics is the derivation of stability function whose increase in value guarantees improved network stability. The evaluation introduced this useful metric to measure the stability performance, thus stability function is a measure that gives the level of stability guaranteed by a particular network setup. The stability time is, therefore, the time frame for a single dynamic topology to remain stable upon reconfiguration due to its adaptability (Vilca 2016, Chen 2013).

Thus, it is expected that the TCP's performance under any network stability mechanism will improve significantly once a change is encountered in the system. Although scholars in this area have tried to address this, their focus is mainly on either favouring the provider or the end-users. The MOCQ proposed an adaptive rendering that provided the stability function which is dependent on the hierarchical setup with both active and multiple mapping solutions running on top of it.

Therefore, the MOCQ delivered a higher ratio of the rate of change of link failures to switch failures. This implied that although a link or two could fail, several alternative links are utilised so far the switches are still active because there are multiple mappings in place and they are already in process. Although some expressions in other scholars do add to the rate of change of controllers to the stability function expression at the denominator, this study regarded this as insignificant based on the assumption that the controllers are relatively never overwhelmed due to cloud support. Therefore, the stability function of a typical network provided the level of satisfaction that the users derived and served as the source of motivation for end-users to demand more services. The demand and supply law ensures that the end-users adhere to and keep demanding the services from the providers as a result of the guaranteed level of stability that is provided to them.

Moreover, the rate of network convergence is another important metric that expresses the level of satisfaction or benefit to the end-users. When a network takes a long time to attain its convergent state, it implies that it had encountered some network traffic problem or issues that had accounted for the utilization of longer time to reach execution. Therefore, the rate of convergence provided by the MOCQ happened to be the lowest amongst them all. This served as a motivation to the end-users to continue to use the service provider.

Other common metrics such as response time, throughput, and availability further provided the end-users with reasons to either continue or discontinue the use of the service from the service provider. In conclusion, the significance of these experiments in this chapter is to balance up the already proven network provision satisfaction from chapter six from the end-user's perspective either positively or negatively. The analyses were drawn to bring about a balanced perspective while deploying MOCQ to alleviate the traffic flow interference in a DCN.

7.5 Chapter Summary

We addressed basic challenges by deploying a deterministic ART (algorithm) that used the MOCQ approach that contains the flow sorting and multiple network element mapping which assisted in establishing the stability of the network performance in relation to the effect of network interferences. The mapping approach harnessed the versatility of distributed networks as it can render a wider influx of transferred flows while maintaining the QoS of varying network users. Hence, the proposed algorithm was able to optimize the network metrics requirements to an optimal level on the basic hierarchical network setup data centre with MOCQ on PHMS. The evaluation of the algorithm against others showed that our proposed algorithm was able to maintain a lower network running time and higher success ratio to provoke low latency and higher throughput respectively. We proposed an algorithm for the determination of the rate of convergence of a network traffic flow resolution approach. This was because the simulator used for the experiments does not have an inbuilt network convergence algorithm.

Thus, the experiments also showed a faster convergence with a reduction of 42.9% in the network delays in attaining the network stability state for MOCQ than the existing algorithms even though the interference could not be eradicated. This implied that our proposed algorithm could assist in managing the network resources especially in a situation where the number of controllers available is less than what is required. The study was also able to reduce the key inhibitors to network stability which is predominantly the control flow and the channel interferences. The last chapter concludes the whole thesis and explains briefly the discussions on future research.

Chapter 8

CONCLUSIONS AND FUTURE RESEARCH

The study in this thesis investigated the problem of traffic flow interference within a DCN environment which often resulted in network instability that is experienced in the entire network setup. The deductions from literature attested to the fact that the challenge has persisted for a long time in the network setting because the approach which has been applied by various scholars in the field has not been adequate. Many of these approaches are still prone to slow convergence, high computational complexity, complex memory requirements, and lack of adaptive mechanisms that could act intelligently thereby ensuring all the stakeholders' satisfaction. The need for addressing the instability in DCN necessitates the investigation of an approach called the MOCQ approach which evolved from the ART. The investigations showed that the performance of a DCN that is challenged with network instability could be enhanced by the deployment of an adaptive technological system that can extend the mapping optimization of the network flows in a manner that inhibits traffic flow interference.

The analyses and discussions regarding the conducted investigations showed that MOCQ proved to be useful, based on the results that were derived from the earlier chapters. The ART is a proven approach in computer graphics, often used for presenting an image in a manner that best suits the device, renders the picture accurately, and as well gives the viewer the best representation of the image. The similitude of this approach was deployed in solving the problem of traffic flow interference within the DCN environment through mathematical analysis, computer simulations, and miniature testbed methodology. With these methodological approaches, we can establish the fact that the network instability usually experienced by the DCN due to high traffic influx is drastically minimized and optimally controlled by the use of the MOCQ algorithm. This algorithm adaptively resolves the traffic to avert the occurrence of network instability. This chapter concludes the study conducted in this thesis as well as details the key contributions with future researches which are closely related to this research.

8.1 Overview of Thesis Findings

The study of ART concerning the occurrence of traffic flow instability in the DCN is of pivotal importance due to its unique ways of iteratively adapting itself toward optimally presenting images through the device. The approach proposed in our study usually undergoes a series of transformations and reformations for optimal rendering, thus making it adaptable to solving a traffic flow problem by adapting the network buffer mapping system. Several network management challenges have been addressed in literature. These include reliability, fault tolerance, slow network convergence, scalability, security as well as expandability. However, the aspect of the stability of the network is still an interesting area that needs to be addressed. Network instability remains an open issue that calls for more attention having received little or no recognition from literature especially in DCN environment.

The study in this thesis focused on addressing the network instability challenge that is incurred due to traffic flow interferences within a typical DCN environment. The occurrence of flow interference is primarily accounted for by the inefficient utilization of the network resources, poor buffer regulation approach, and lack of a stringent mechanism for low delay realisation in service requests. Our investigations into the approaches often used to address this problem include transport layer, switch-based, predictive flow scheduling, and Ethernet protocol solutions have not yielded the required optimal result. Most of these inclined attempts so far based on literature have not fully dealt with flow interference to the best of our knowledge. This is because the deployment of one single approach is insufficient to effectively address the challenge of traffic flow interference. To this end, Chapter 2 provided us with the uniqueness of the problem of flow interference as well as “why” and “how”, we planned to address it.

Moreover, the majority of the earlier types of researches in literature related their proposed solutions predominantly to either the service provider benefits alone or strictly motivated and aligned with the end user in mind as discussed in Section 2.3 of Chapter 2. Thus, the missing link further deepens with the importance of balancing proposed network stability solutions such that both the service provider and the end user can protect their interests without one affecting the other, through ensuring a strike of balance between high bandwidth utilisation and traffic flow stability (Jain et al 2013, Sudheera et al 2019, Akinola et al 2020, Tseng et al 2019). Similarly, the end-user must have a great influence in terms of their desire for the best user experience with relatively fair treatment while transacting through the data centre. We argued that these two strong observations are very pivotal and important resultant effects of a stable network because they are measurable quantities to ascertain the performance of the network environment in the long run.

The successful investigations we made resulted in the deployment of ART which gave birth to the Multi-Objective Optimisation Crosspoint Queue (MOCQ) approach which helped in tackling the flow interference challenge as well as provided us with an optimal solution in comparison with the existing solutions. The problem description was analysed in Chapter 3 to formulate the mathematical model of the proposed approach which yielded an optimal expression to determine the optimal performance while deploying the ART. A subsequent algorithm that also imbibed the primary stages of control flow rendering adaptation pipeline in its formulation was designed. Chapter 4 described the use of MOCQ in proxy-controller (P-C) deployment: an unspecified quality approach that budded from the hybrid of switch-based and transport layer features to test the performance of our idea on traffic flow interference using both simulation setup and numerical analysis methodology.

Furthermore, the network bandwidth which happened to be an important metric during network critical flows was used as a tool to address imbalanced traffic flows. The proposed framework is designed to determine the available bandwidth at any instance to use the information to provide a stabilized network environment that accrues both service providers and end-users with maximal benefits. Both Chapters 6 and 7 described a PHMS use case scenario that used Simulation and miniature testbed methods to validate the performance of the MOCQ approach. The simulation-based experimental results achieved in this study showed that the performance of P-C aided MOCQ generates 40 percent improvement over the cases where there is no reactive mode (nRM) in place as the ratio drops from 1.9 to 1.1 depicting the level of fairness in the performance of our approach. This fairness is further explained to imply that the measure of the momentum of mapping and re-mapping of switches to controllers is lower depicting that we attained a relatively stable network. Moreover, we determined the resistance constraint for the network to be 0.7 and below which is under-utilisation of the network resources and above which the network becomes unstable. This measure helps to determine the "elastic limit" for any network which is very important to the service provider for cost maximization and end-users for optimal user experience in the same vein. These results were further validated with both simulation and numerical analysis as both methods gave us similar resistance constraints for the referenced network setup.

We were also able to correlate the gain of the service provider in minimizing the cost implications as well as maximizing the user experience of the end-user by formulating an optimizing solution approach which provided a relatively stable user experience over a range of 0,50 irrespective of fluctuations in the average requests that was incurred. The outcomes of this optimisation objective helped us to attain three significant results that are vital to the service provider:

Firstly, the average overall response time (ms) for traffic flow in the system is determined at around 600 ms, thus the service provider and the end-user were able to derive an optimal benefit.

Secondly, we can determine the possibility of the service provider to vary his cost of rendering the services without a drastic effect on the output on the end user part thereby maximizing the providers' cost.

Thirdly, both parties will benefit from their service being rendered with the major interests of both parties protected. This result gave us a landslide contribution to the network stability field in that this research is one of its kind to have made a model which could provide a standard for setting up a reliable and efficient network environment.

In conclusion, the percentage of switch failures recorded is on the average of 0.25 percent which is equivalent to over 150 percent improvement over the performance of CQ and 75 percent over CQRD thereby validating better performance from MOCQ. Simulation and miniature testbed were also used to determine the stability function and rate of convergence with the MOCQ attaining faster convergence with a reduction of 42.9 percent. The evaluation of our proposed algorithm was able to maintain a lower network running time and higher success ratio to provoke low latency and higher throughput respectively as confirmed by our experiments.

8.2 Thesis Contributions

To achieve the goal of this thesis, the traffic flow interference challenge which rendered the network unstable was addressed by exploiting the ART to ensure that the network flow adapt dynamically by using the MOCQ approach. The case for the approach was motivated in Chapter 2. We stated that the challenge of traffic flow interference occurs when two or more flows encounter competition with the use of network resources such as flow path, queue memory, link capacity, and or network channel. Other pieces of research and most available literature only manage the loads and ensure scalability but the aspect of the instability caused by interference has not been fully addressed especially in ensuring that the interests of both parties on the platform were not jeopardised at the expense of another. Thus, this brought out the need for the approach we proposed in this thesis to ensure network stability. The main contributions of this thesis may be outlined as follows:

1. **Deployment of Adaptive Rendering Technique**:

   The motivation of being the first to deploy the computer graphics technique ART, to solve a network management problem is a novel contribution to the field. The technique utilised the adaptive rendering nature of the light rays' combination and its technique to produce a stable frame image buffer representation for the destined devices. The stages that are usually undergone in ART include display list, vertex shading, rasterization, and raster operations. Similarly, this same approach is replicated in creating a control flow list, sorting entry matching, matching flow table, and matching operations in the pipeline for the delivery of a stable Controller-Switch network state. This successful implementation thereby added value to the other research efforts when selecting an approach against which to compare and/or develop a new technique.

2. **Developing a unique mathematical model for network rendering system management**:

   The achievement of the mathematical model that solved a network problem has proved significant in the sense that the model is unique and could serve as a foundation for its deployment in solving other network challenges. Our research has successfully laid the foundation upon which interested and intending researchers can learn the modelling knowledge to design a similar but typical mathematical model that will address other problems in this area of research or a new area entirely.

3. **Extending and providing solutions that consider stakeholders' interests**:

   The service provider and end-users interests have not been taken into considerations of network management solutions until now. Several approaches that were proposed from literature are not only not optimal but also one-sided in the sense that they either propose the solutions considering the service provider costs in mind alone while others are with the view to maximize the end-user experience at the expense of the service provider. As the first of its kind, we extended the body of knowledge not only to address the traffic flow interference but also to solve it in a novel manner that is satisfying to both service providers and the end-users.

8.3 Future Research

Several promising areas of future research came out of this study. This section presents how this study reveals avenues for future research directions. Some of these possible further research directions could be summarised as follows:

One of the limitations of our study is the use of a miniature test-bed which cannot entirely send inflows as requested due to the vast number of network hosts used in the course of our experiments. The next interesting step would be to use a test-bed to verify the performance of our proposed approach in this thesis thereby extending the validation of the results that were obtained in this study. Moreover, the buffer size plays a significant role in the performance of the network elements, and in the case of this work, we used a fixed buffer size against the normal convention where several sizes of the buffer are experienced. This study would be improved by considering situations where the network elements consist of varying sizes of buffers and probably the performance regarding the network stability is available.

Having deployed the MOCQ approach in enhancing network stability in the DCN, it would be very interesting for us to see the performance of this proposed approach in other network environments, such as sensor networks, home-based networks, and probably in the Internet of Things networks as well. Other areas of endeavour for this research for future work is the introduction and the deployment of trained machine learning algorithms or other specialised Artificial Intelligence mechanisms to address cases where the incremental deployment of traffic flows occurs. By incremental deployment, we refer to the situation where flows are not categorised as we did in our study. The entire flows could be giant flows throughout a particular network in some specialized data centres. In such situations, the kind of heuristic approach which is devised to enhance improved latency might be out of consideration hence machine learning could play a pivotal role. This could, therefore, introduce some level of intelligence to enhance the stability of data centre enabled SDN.

Furthermore, as it is generally expected in any optimisation problem, some variables are oftentimes left out for others, not necessarily because they are not important but perhaps because they are not at the centre of consideration concerning the challenges at hand. One of such variables in this study is the issue of the network elements' cost while choosing the network setup and the model deployment. Although this study looked at the cost in terms of the relative satisfaction from the stakeholders involved, the cost is referred to here is that of the infrastructures. It would be very interesting to embark on future research that will factor in the problem of traffic flow interference and the effect of the cost of the network elements to be used in the industry. This is because industries are often concerned about the cost implication of the model that is proposed to them.

In conclusion, how we deploy the ART has made it perform better than the existing solutions from literature, the ART which is based on its adherent uniqueness in adaptive network flow regulation and demonstrated substantial improvements in terms of both numerical network delay and quality of network stability. This similar implementation in addition to standard deterministic adaptive rendering could be of pivotal use in addressing research problems. It will require tailored solutions in areas like sensor networks, vehicular ad hoc networks, campus area networks, ad hoc networks, RiNa networks, Peer to Peer networks, and of course 5G LTE networks. According to the authors Rousselle (2012) and Anshelevich (2008), the ART enables the fundamental framework to enhance easy implementations on top of any standard variety of ARTs (Monte Carlo, Object-based and Image-based) to derive a user-tailored solution. Hence, with the experience of the use of ART in this study, it is envisaged that similar irregularities in other networks could be stabilised by adopting appropriate tailored solution mechanisms on it.

Finally, we use the hierarchical setup on Elastic tree architecture for our data centre set up instead of the Optical Switch data centre architecture which from literature has been reported to have the best performance. It will be very interesting to determine how the evolved MOCQ from the ART in this study would perform using the Optical switch data centre architecture. Although the research report provided us with the insight that the Optical switch data centre architecture is one of the best performing data centre architectures, we are uncertain how MOCQ would perform on the platform. Hence, it would be interesting research to see how the outcome of this study expresses itself on it. The future research envisaged here is that the ability to dynamically reconfigure Polatis switches via the standard based SDN interfaces using MOCQ will enable the optical switches to coexist with other flow switches under centralized management control for efficient orchestration of topology or even traffic flow changes. The OpenFlow clients could probably enhance the envisaged data centre colocation with network operators to reconfigure for stability on demand and deploy more capacity where it is needed. This will make productive use of the network resources at an appreciably lower cost.

References

ABDALLAH, S., KAYSSI, A., ELHAJJ, I. H. & CHEHAB, A., 2018. Network convergence in SDN versus OSPF networks. 2018 Fifth International Conference on Software Defined Systems (SDS), pp. 130-137.

ABUTEIR, R. M., FLADENMULLER, A. & FOURMAUX, O., 2016. SDN based architecture to improve video streaming in home networks. 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), 2016. IEEE, pp. 220-226.

ABOUBAKAR, M., KELLIL, M. AND ROUX, P., 2021. A review of IoT network management: Current status and perspectives. Journal of King Saud University-Computer and Information Sciences.

AHMAD, S. AND MIR, A.H., 2021. Scalability, consistency, reliability and security in SDN controllers: a survey of diverse SDN controllers. *Journal of Network and Systems Management*, *29*(1), pp.1-59.

AKINOLA, A., ADIGUN, M., AKINGBESOTE, A. & MBA, I., 2015. Optimal route service selection in ad-hoc mobile E-marketplaces with dynamic programming algorithm using TSP approach. proceedings on International Conference on E-Learning Engineering and Computer Softwares, Kuala Lampur, Malaysia, pp.74-81.

AKINOLA, A. T. & ADIGUN, M. O., 2017. Performance Evaluation of Data Center Network Setup Architectures. Southern African Telecommunication Networks and Application Conference (SATNAC) Barcelona, Spain. pp. 166-171.

AKINOLA, A. T., ADIGUN, M. O. & MUDALI, P., 2018. Effects of Scalability measures on Control Plane Architectural Designs in different Data Centre Architectures. Southern

African Telecommunication and Network Applications Conference, Arabella, Hermanus, Western Cape, South Africa. pp.198-203.

AKINOLA, A. T., ADIGUN, M. O. & MUDALI, P., 2019. Effect of SDN Controller-Proxy Deployment on Network Stability. Southern Africa Telecommunication Networks and Applications Conference (SATNAC) Fairmont Zimbali Resort, Ballito RSA. pp. 1-10.

AKINOLA, A. T., ADIGUN, M. O. & MUDALI, P., 2020. Incorporating Stability Estimation into Quality of Service Routing in an SDN-Based Data Centre Network. Proceedings of Fifth International Congress on Information and Communication Technology, Springer, pp. 406-416.

AKINOLA, A. T., ADIGUN, M. O. & MUDALI, P., 2020. Performance Modeling of Software-Defined Networking Paradigm in a Public Health Management System. *International Journal of Information and Management Sciences,* 31(2)**,** pp.123-148.

ALIZADEH, M., GREENBERG, A., MALTZ, D. A., PADHYE, J., PATEL, P., PRABHAKAR, B., SENGUPTA, S. & SRIDHARAN, M., 2010. Data center tcp (dctcp). Proceedings of the ACM SIGCOMM 2010 conference, pp. 63-74.

ALIZADEH, M., GREENBERG, A., MALTZ, D. A., PADHYE, J., PATEL, P., PRABHAKAR, B., SENGUPTA, S. & SRIDHARAN, M., 2011. Data center tcp (dctcp). *ACM SIGCOMM computer communication review,* 41**,** pp. 63-74.

ALIZADEH, M., KABBANI, A., EDSALL, T., PRABHAKAR, B., VAHDAT, A. & YASUDA, M., 2019. Less is more: trading a little bandwidth for ultra-low latency in the data center. Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12), pp. 253-266.

ALIZADEH, M., YANG, S., SHARIF, M., KATTI, S., MCKEOWN, N., PRABHAKAR, B. & SHENKER, S., 2019. pfabric: Minimal near-optimal datacenter transport. *ACM SIGCOMM Computer Communication Review,* 43**,** pp. 435-446.

ALOTAIBI, M. AND NAYAK, A., 2021. Linking handover delay to load balancing in SDN-based heterogeneous networks. *Computer Communications*, *173*, pp.170-182.

ALTAHEINI, S., MOSTAGHIMI, P. AND CLARK, S.R., 2022. Incorporating adaptive well conditions into numerical reservoir simulators constrained by well and reservoir performance data. Journal of Petroleum Science and Engineering, 208, p.109-684.

ANEROUSIS, N., CHEMOUIL, P., LAZAR, A.A., MIHAI, N. AND WEINSTEIN, S.B., 2021. The origin and evolution of open programmable networks and sdn. *IEEE Communications Surveys & Tutorials*, *23*(3), pp.1956-1971.

ANSHELEVICH, E., DASGUPTA, A., KLEINBERG, J., TARDOS, É., WEXLER, T. AND ROUGHGARDEN, T., 2008. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing,* 38**,** pp. 1602-1623.

ASHOURI, M. & SETAYESH, S., 2018. Enhancing the Performance and Stability of SDN Architecture with a Fat-Tree Based Algorithm.

ATIEH, A.T., 2021. Assuring the Optimum Security Level for Network, Physical and Cloud Infrastructure. ScienceOpen Preprints.

BJÖRNSON, E. AND SANGUINETTI, L., 2020. Rayleigh fading modeling and channel hardening for reconfigurable intelligent surfaces. IEEE Wireless Communications Letters, 10(4), pp.830-834.

BOLOOR, K., CHIRKOVA, R., VINIOTIS, Y. & SALO, T., 2019. Dynamic request allocation and scheduling for context aware applications subject to a percentile response time SLA in a distributed cloud. In 2010 IEEE Second International Conference on Cloud Computing Technology and Science, IEEE, pp. 464-472.

BONALD, T. & MASSOULIÉ, L., 2001. Impact of fairness on Internet performance. Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pp. 82-91.

BRADEN, R., CLARK, D., SHENKER, S. & WROCLAWSKI, J., 2000. Developing a next-generation Internet architecture. *White paper, DARPA.*

BUYYA, R., RANJAN, R. & CALHEIROS, R. N., 2010. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. International Conference on Algorithms and Architectures for Parallel Processing, Springer, pp. 13-31.

CALVERT KL, D. M., ZEGURA EW., 1997. Modeling internet topology. *IEEE Communications magazine,* 35**,** pp. 160-163.

CARIA, M. & JUKAN, A., 2016. Link capacity planning for fault tolerant operation in hybrid SDN/OSPF networks. Global Communications Conference (GLOBECOM), IEEE, pp. 1-6.

CHAKRAVARTHY, V.D. AND AMUTHA, B., 2022. A novel software-defined networking approach for load balancing in data center networks. *International Journal of Communication Systems*, 35(2), p.e4213.

CHEN, G., ZHAO, Y. & PEI, D., 2015. Alleviating flow interference in data center networks through fine-grained switch queue management. *Computer Networks,* 91**,** pp. 593-613.

CHENG, G., ANSARI, N. & PAPAVASSILIOU, S., 2008. Adaptive QoS provisioning by pricing incentive QoS routing for next generation networks. *Computer Communications,* 31**,** pp. 2308-2318.

CHENG, T. Y., WANG, M. & JIA, X., 2015. QoS-guaranteed controller placement in SDN. 2015 IEEE Global Communications Conference (GLOBECOM), IEEE, pp. 1-6.

CHERAGHI, E., ZOHREHVANDI, S., VANHOUCKE, M. AND TOSARKANI, B.M., 2022. A multiproject scheduling and resource management model in projects construction. *Engineering, Construction and Architectural Management.*

CHOTCHOUNGCHATCHAI S, M. A., WITTHAYAPIPOPSAKUL W, PANICHKRIANGKRAI W, PATCHARANARUMOL W, TANGCHAROENSATHIEN V,. 2020. Primary health care and sustainable development goals. *Bulletin of the World Health Organization*, 98**,** pp. 792.

CICIOĞLU, M. & ÇALHAN, A., 2019. SDN-based wireless body area network routing algorithm for healthcare architecture. *Etri Journal*, 41**,** pp. 452-464.

CISCO, V., 2018. Cisco Visual Networking Index: Forecast and Trends, pp. 2017–2022. *White Paper*.

CLAYMAN, S. AND SAYIT, M., 2021, June. In-Network Scalable Video Adaption Using Big Packet Protocol. In Proceedings of the 12th *ACM Multimedia Systems Conference* (pp. 363-368).

CORREIA, L. M., ABRAMOWICZ, H., JOHNSSON, M. & WÜNSTEL, K., 2011. *Architecture and design for the future internet: 4WARD project*, Springer Science & Business Media.

D'AVILA, L.F., BARBOSA, J.L.V. AND DE OLIVEIRA, K.S.F., 2020. SW-Context: a model to improve developers' situational awareness. IET Software, 14(5), pp.535-543.

DIXIT, A., HAO, F., MUKHERJEE, S., LAKSHMAN, T. & KOMPELLA, R., 2019. Towards an elastic distributed SDN controller. Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, pp. 7-12.

DUKKIPATI, N. & MCKEOWN, N., 2020. Why flow-completion time is the right metric for congestion control. *ACM SIGCOMM Computer Communication Review,* 36**,** pp. 59-62.

EK, N., 1999. Ieee 802.1 p, q-qos on the mac level. *Apr,* 24**,** pp. 3-6.

ESAKI, S., KUROKAWA, A. & MATSUMOTO, K., 2007. Overview of the Next Generation Network. *Special Feature: NGN Communication Styles and Support Technologies, NTT Technical Review,* 5.

FARHADY, H., LEE, H. & NAKAO, A., 2015. Software-defined networking: A survey. *Computer Networks,* 81**,** pp. 79-95.

FEAMSTER, N., BALAKRISHNAN, H., REXFORD, J., SHAIKH, A. & VAN DER MERWE, J., 2004. The case for separating routing from routers. Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture, ACM, pp. 5-12.

FILIPPETTO, A.S., LIMA, R. AND BARBOSA, J.L.V., 2021. A risk prediction model for software project management based on similarity analysis of context histories. Information and Software Technology, 131, p.106497.

GAO, P. X., CURTIS, A. R., WONG, B. & KESHAV, S., 2019. It's not easy being green. *ACM SIGCOMM Computer Communication Review,* 42**,** pp. 211-222.

GAO, X., KONG, L., LI, W., LIANG, W., CHEN, Y. & CHEN, G., 2019. Traffic load balancing schemes for devolved controllers in mega data centers. *IEEE Transactions on Parallel and Distributed Systems,* 28**,** pp. 572-585.

GAUR, V., YADAV, O.P., SONI, G. AND RATHORE, A.P.S., 2021. A literature review on network reliability analysis and its engineering applications. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 235(2), pp.167-181

GEBREHIWOT M.E, A. S., LASSILA P. 2017. Energy efficient load balancing in web server clusters. *29th International Teletraffic Congress (ITC 29) IEEE,* 3**,** 13-18.

GEMAWAT, A., 2021, GraphGem: Optimized Scalable System for Graph Convolutional Networks. In Proceedings of the 2021 International Conference on Management of Data (pp. 2920-2922).

GOLESTANI, S. J., 1994. A self-clocked fair queueing scheme for broadband applications. Proceedings of INFOCOM'94 Conference on Computer Communications, 1994. IEEE, pp. 636-646.

GONZALEZ, L.F., VIDAL, I., VALERA, F., NOGALES, B., SANCHEZ-AGUERO, V. AND LOPEZ, D.R., 2019. Transport-layer limitations for NFV orchestration in resource-constrained aerial networks. Sensors, 19(23), p.5220.

GREENBERG, A., HAMILTON, J. R., JAIN, N., KANDULA, S., KIM, C., LAHIRI, P., MALTZ, D. A., PATEL, P. & SENGUPTA, S., 2019. VL2: a scalable and flexible data center network. ACM SIGCOMM computer communication review, ACM, pp. 51-62.

GU, Z., WANG, Z., LIU, Z. AND SABERI, M., 2022. Network traffic instability with automated driving and cooperative merging. *Transportation Research Part C: Emerging Technologies*, *138*, p.103626

GUAN, X., CHOI, B.-Y. & SONG, S., 2013. Reliability and scalability issues in software defined network frameworks, Research and Educational Experiment Workshop (GREE), Second GENI, IEEE, pp.102-103.

GUENNEBAUD, G., BARTHE, L. & PAULIN, M., 2007. High-quality adaptive soft shadow mapping. Computer graphics forum, Wiley Online Library, pp. 525-533.

GUO, C., WU, H., TAN, K., SHI, L., ZHANG, Y. & LU, S., 2018. Dcell: a scalable and fault-tolerant network structure for data centers. Proceedings of the ACM SIGCOMM 2018 conference on Data communication, pp. 75-86.

HADLEY, M., NICOL, D. & SMITH, R., 2017. Software-defined networking redefines performance for ethernet control systems, In Power and Energy Automation Conference.

HAMMADI, A. & MHAMDI, L., 2014. A survey on architectures and energy efficiency in Data Center Networks. *Computer Communications,* 40**,** pp. 1-21.

HE, M., BASTA, A., BLENK, A. & KELLERER, W., 2017. Modeling flow setup time for controller placement in SDN: Evaluation for dynamic flows. In 2017 IEEE International Conference on Communications (ICC), IEEE, pp. 1-7.

HONG, C.-Y., CAESAR, M. & GODFREY, P. B., 2022. Finishing flows quickly with preemptive scheduling. *ACM SIGCOMM Computer Communication Review,* 42**,** pp. 127-138.

HU, J., LIN, C., LI, X. & HUANG, J., 2014. Scalability of control planes for software defined networks: Modeling and evaluation. 2014 IEEE 22nd International Symposium of Quality of Service (IWQoS), IEEE, pp. 147-152.

HU, Y., WENDONG, W., GONG, X., QUE, X. & SHIDUAN, C., 2013. Reliability-aware controller placement for software-defined networks. 2013 IFIP/IEEE International Symposium on Integrated Network Management, pp. 672-675.

HUANG, V., CHEN, G., FU, Q. & WEN, E., 2019. Optimizing Controller Placement for Software-Defined Networks. *arXiv preprint arXiv:1902.09451*.

HUJAINAH, F., BAKAR, R.B.A., NASSER, A.B., AL-HAIMI, B. AND ZAMLI, K.Z., 2021. SRPTackle: A semi-automated requirements prioritisation technique for scalable requirements of software system projects. Information and Software Technology, 131, p.106501.

INET.OMNETPP.ORG. 2022. Step 7. Configuring a hierarchical network — INET 4.3.0 documentation.[online]Availableat:<https://inet.omnetpp.org/docs/tutorials/configurator/doc/step7.html> [Accessed 7 April 2022].

JAGANNATH, J., JAGANNATH, A., HENNEY, J., GWIN, T., KANE, Z., BISWAS, N. AND DROZD, A., 2022. Design of Fieldable Cross-Layer Optimized Network using Embedded Software Defined Radios: Survey and Novel Architecture with Field Trials. *arXiv preprint arXiv:2201.09454*.

JAIN, S., TRIPATHI, V. S. & TIWARI, S., 2013. Bandwidth allocation based on traffic load and interference in IEEE 802.16 mesh networks. *Journal of Engineering*.

JUDE MJ, D. V., SHIVARANJANI M, MADHUMITHA S, BALAJI VK, MYVIZHI M., 2021. Improving Fairness and Convergence Efficiency of TCP Traffic in Multi-hop Wireless Networks. *Wireless Personal Communications***,** pp. 1-27.

KAHN, J. M., KATZ, R. H. & PISTER, K. S., 1999. Next century challenges: mobile networking for “Smart Dust”. Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, 1999. pp. 271-278.

KAJIYA, J. T., 1986. The rendering equation. Proceedings of the 13th annual conference on Computer graphics and interactive techniques, 1986. pp. 143-150.

KALAI, E., 1985. Solutions to the bargaining problem. *Social goals and social organization: Essays in memory of Elisha Pazner***,** pp. 77-106.

KARAKUS, M. & DURRESI, A., 2016. A scalability metric for control planes in software defined networks (SDNs). 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA). IEEE, pp. 282-289.

KARAKUS, M. & DURRESI, A., 2017. A survey: Control plane scalability issues and approaches in software-defined networking (SDN). *Computer Networks,* 112**,** pp. 279-293.

KASTRINOGIANNIS, T., TCHOLTCHEV, N., PRAKASH, A., CHAPARADZA, R., KALDANIS, V., COSKUN, H. & PAPAVASSILIOU, S., 2010. Addressing stability in future autonomic networking. International Conference on Mobile Networks and Management, Springer, pp. 50-61.

KHORSANDROO, S., SANCHEZ, A.G., TOSUN, A.S., ARCO, J.M. AND DORIGUZZI-CORIN, R., 2021. Hybrid SDN evolution: A comprehensive survey of the state-of-the-art. Computer Networks, 192, p.107981.

KORKMAZ, T. & KRUNZ, M., 2001. Multi-constrained optimal path selection. Proceedings IEEE INFOCOM 2001 conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213), IEEE, pp. 834-843.

KUTEYI, D. AND WINKLER, H., 2022. Logistics Challenges in Sub-Saharan Africa and Opportunities for Digitalization. Sustainability, 14(4), p.2399

LAFORTUNE, E., 1996. Mathematical models and Monte Carlo algorithms for physically based rendering. *Department of Computer Science, Faculty of Engineering, Katholieke Universiteit Leuven***,** pp. 74-79.

LAFORTUNE, E. P. & WILLEMS, Y. D., 1994. A theoretical framework for physically based rendering. Computer Graphics Forum, Wiley Online Library, pp. 97-107.

LI C, W. Y., TANG H, ZHANG Y, XIN Y, LUO Y., 2019. Flexible replica placement for enhancing the availability in edge computing environment. *Computer Communications*, 1-14.

LI, M., SHEN, L. AND HUANG, G.Q., 2019. Blockchain-enabled workflow operating system for logistics resources sharing in E-commerce logistics real estate service. Computers & Industrial Engineering, 135, pp.950-969.

LIEBEHERR, J., FIDLER, M. & VALAEE, S., 2010. A system-theoretic approach to bandwidth estimation. *IEEE/ACM Transactions on networking,* 18**,** pp. 1040-1053.

LIMA, R., FILIPPETTO, A.S., HECKLER, W., BARBOSA, J.L. AND LEITHARDT, V.R., 2022. Towards ubiquitous requirements engineering through recommendations based on context histories. *PeerJ Computer Science*, 8, p.e794.

LINGUAGLOSSA, L., LANGE, S., PONTARELLI, S., RÉTVÁRI, G., ROSSI, D., ZINNER, T., BIFULCO, R., JARSCHEL, M. AND BIANCHI, G., 2019. Survey of performance acceleration techniques for network function virtualization. *Proceedings of the IEEE*, 107(4), pp.746-764

LIU, Z., LIN, M., WIERMAN, A., LOW, S. H. & ANDREW, L. L., 2019. Greening geographical load balancing. *ACM SIGMETRICS Performance Evaluation Review,* 39**,** pp. 193-204.

LIU, K., ZHAO, A. AND PAN, J., **2021**. Data Center Architecture, Operation, and Optimization. In *Future Networks, Services and Management* (pp. 185-212). Springer, Cham.

LIU, S., LIU, D., SRIVASTAVA, G., POŁAP, D. AND WOŹNIAK, M., 2021. Overview and methods of correlation filter algorithms in object tracking. *Complex & Intelligent Systems*, 7(4), pp.1895-1917.

MAHIMKAR, A., CHIU, A., DOVERSPIKE, R., FEUER, M.D., MAGILL, P., MAVROGIORGIS, E., PASTOR, J., WOODWARD, S.L., YATES, J., 2011. Bandwidth on demand for inter-data center communication. *10th ACM Workshop on Hot Topics in Networks.*

MAHMUD, R., SRIRAMA, S.N., RAMAMOHANARAO, K. AND BUYYA, R., 2019. Quality of Experience (QoE)-aware placement of applications in Fog computing environments. *Journal of Parallel and Distributed Computing*, *132*, pp.190-203.

MAIA, A.M., GHAMRI-DOUDANE, Y., VIEIRA, D. AND DE CASTRO, M.F., 2021. An improved multi-objective genetic algorithm with heuristic initialization for service placement and load distribution in edge computing. *Computer Networks,* 194, p.108146

MARIN, A. AND HAMPTON, K.N., 2019, June. Network instability in times of stability. In Sociological Forum (Vol. 34, No. 2, pp. 313-336)

MAYER, A., LORETI, P., BRACCIALE, L., LUNGARONI, P., SALSANO, S. AND FILSFILS, C., 2021. Performance Monitoring with Hˆ2: Hybrid Kernel/eBPF data plane for SRv6 based Hybrid SDN. Computer Networks, 185, p.107705

MCKENNEY, P., 1990. Stochastic fairness queueing. IEEE INFOCOM'90,. pp. 733-740.

MERLE, S., FERNÁNDEZ-PALACIOS, J.P., DE DIOS, O.G., GIFRE, L., VILALTA, R. AND STRITZINGER, P., 2021, November. Scalable and Resilient Network Traffic Engineering Using Erlang-based Path Computation Element. In 2021 IEEE *Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN*) (pp. 100-101). IEEE.,

MILJKOVIĆ, B., ŽIŽOVIĆ, M. R., PETOJEVIĆ, A. & DAMLJANOVIĆ, N., 2017. New weighted sum model. *Filomat,* 31**,** pp. 2991-2998.

MISHRA, S. AND TYAGI, A.K., 2022. The Role of Machine Learning Techniques in Internet of Things-Based Cloud Applications. In *Artificial Intelligence-based Internet of Things Systems* (pp. 105-135). Springer, Cham

MO, J. & WALRAND, J., 2020. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on networking,* 8**,** pp. 556-567.

NAOURI, A., WU, H., NOURI, N.A., DHELIM, S. AND NING, H., 2021. A Novel Framework for Mobile-Edge Computing by Optimizing Task Offloading. *IEEE Internet of Things Journal*, *8*(16), pp.13065-13076

NG, T.-T., PAHWA, R. S., BAI, J., TAN, K.-H. & RAMAMOORTHI, R., 2012. From the rendering equation to stratified light transport inversion. *International Journal of Computer Vision,* 96**,** pp. 235-251.

NGUYEN, V. D., BEGIN, T. & GUÉRIN-LASSOUS, I., 2013. Multi-constrained routing algorithm: A networking evaluation. 2013 IEEE 37th Annual Computer Software and Applications Conference Workshops, IEEE, pp. 719-723.

OTHMAN, M. O. & OKAMURA, K., 2013. Hybrid control model for flow-based networks. 2013 IEEE 37th Annual Computer Software and Applications Conference Workshops (COMPSACW), IEEE, pp. 765-770.

PANDI, V., PERUMAL, P., BALUSAMY, B. AND KARUPPIAH, M., 2019. A novel performance enhancing task scheduling algorithm for cloud-based E-health environment. *International Journal of E-Health and Medical Communications (IJEHMC)*, *10*(2), pp.102-117.

PEETA, S. & YANG, T.-H., 2003. Stability issues for dynamic traffic assignment. *Automatica*, 39**,** pp. 21-34.

POLESE, M., CHIARIOTTI, F., BONETTO, E., RIGOTTO, F., ZANELLA, A. AND ZORZI, M., 2019. A survey on recent advances in transport layer protocols. IEEE Communications Surveys & Tutorials, 21(4), pp.3584-360.

RAMAKRISHNAN, K. & FLOYD, S., 1999. A proposal to add explicit congestion notification (ECN) to IP. RFC 2481, January.

REHMAN, A.U., WADUD, Z., ELAVARASAN, R.M., HAFEEZ, G., KHAN, I., SHAFIQ, Z. AND ALHELOU, H.H., 2021. An optimal power usage scheduling in smart grid integrated with renewable energy sources for energy management. IEEE Access, 9, pp.84619-84638.

ROUSSELLE, F., KNAUS, C. AND ZWICKER, M., 2012. Adaptive rendering with non-local means filtering. *ACM Transactions on Graphics (TOG),* 31**,** pp. 1-11.

RYGIELSKI, P., KOUNEV, S. & ZSCHALER, S., 2013. Model-based throughput prediction in data center networks. 2013 IEEE International Workshop on Measurements & Networking (M&N), IEEE, pp. 167-172.

SAFRON, A., KLIMAJ, V. AND HIPÓLITO, I., 2022. On the importance of being flexible: dynamic brain networks and their potential functional significances. Frontiers in Systems Neuroscience, p.149.

SAMARAS, I. K., HASSAPIS, G. D. & GIALELIS, J. V., 2012. A modified DPWS protocol stack for 6LoWPAN-based wireless sensor networks. *IEEE Transactions on Industrial Informatics,* 9**,** pp. 209-217.

SAUL, N., WANG, K., BAG, S., BALDWIN, H., ALEXANDER, K., CHANDRA, M., THOMAS, J., QUINN, H., SHEPPEARD, V. & CONATY, S., 2018. Effectiveness of maternal pertussis vaccination in preventing infection and disease in infants: the NSW public health network case-control study. *Vaccine,* 36**,** pp. 1887-1892.

SELVI, H., GÜR, G. & ALAGÖZ, F., 2019. Cooperative load balancing for hierarchical SDN controllers. 2016 IEEE 17th International Conference on High Performance Switching and Routing (HPSR), IEEE, pp. 100-105.

SEZER, S., SCOTT-HAYWARD, S., CHOUHAN, P. K., FRASER, B., LAKE, D., FINNEGAN, J., VILJOEN, N., MILLER, M. & RAO, N., 2013. Are we ready for SDN? Implementation challenges for software-defined networks. *IEEE Communications Magazine,* 51**,** pp. 36-43.

SHAFIQ, D.A., JHANJHI, N.Z., ABDULLAH, A. AND ALZAIN, M.A., 2021. A Load Balancing Algorithm for the Data Centres to Optimize Cloud Computing Applications. IEEE Access, 9, pp.41731-41744

SHPINER, A., KESLASSY, I., BRACHA, G., DAGAN, E., INY, O. & SOHA, E., 2012. A switch-based approach to throughput collapse and starvation in data centers. *Computer Networks,* 56**,** pp. 3333-3346.

SHREEDHAR, M. & VARGHESE, G., 1996. Efficient fair queuing using deficit round-robin. *IEEE/ACM Transactions on networking,* 4**,** pp. 375-385.

SINGH, A., KORUPOLU, M. & MOHAPATRA, D., 2018. Server-storage virtualization: integration and load balancing in data centers. SC'08: Proceedings of the 2008 ACM/IEEE conference on Supercomputing, IEEE, pp. 1-12.

SONG, P., LIU, Y., LIU, T. & QIAN, D., 2017. Controller-proxy: Scaling network management for large-scale SDN networks. *Computer Communications,* 108**,** pp. 52-63.

SRIDHARAN, V., GURUSAMY, M. & TRUONG-HUU, T. 201,7. On multiple controller mapping in software defined networks with resilience constraints. *IEEE Communications Letters,* 21**,** pp. 1763-1766.

SUDHEERA, K. L. K., MA, M. & CHONG, P. H. J., 2019. Link stability based optimized routing framework for software defined vehicular networks. *IEEE Transactions on Vehicular Technology,* 68**,** pp. 2934-2945.

TAKANO, Y., MUTOH, R., OGUCHI, N. & ABE, S., 2016. Estimating available bandwidth in mobile networks by correlation coefficient. 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE, pp. 1-4.

TOOTOONCHIAN, A. & GANJALI, Y., 2020. Hyperflow: A distributed control plane for openflow. Proceedings of the 2010 internet network management conference on Research on enterprise networking.

TSENG, S.-H., TANG, A., CHOUDHURY, G. L. & TSE, S., 2019. Routing stability in hybrid software-defined networks. *IEEE/ACM Transactions on Networking,* 27**,** pp.790-804.

TSO, F. P., JOUET, S. & PEZAROS, D. P., 2016. Network and server resource management strategies for data centre infrastructures: A survey. *Computer Networks,* 106**,** pp. 209-225.

UL HUQUE, T. I., YEGO, K., SIOUTIS, C., NOBAKHT, M., SITNIKOVA, E. & DEN HARTOG, F., 2019. A system architecture for time-sensitive heterogeneous wireless distributed software-defined networks. 2019 Military Communications and Information Systems Conference (MilCIS). IEEE, pp. 1-6.

VAMANAN, B., HASAN, J. & VIJAYKUMAR, T., 2012. Deadline-aware datacenter tcp (d2tcp). *ACM SIGCOMM Computer Communication Review,* 42**,** pp. 115-126.

VASANTHARAJ, A., RANI, P.S., HUQUE, S., RAGHURAM, K.S., GANESHKUMAR, R. AND SHAFI, S.N., 2021. Automated Brain Imaging Diagnosis and Classification Model Using Rat Swarm Optimization with Deep Learning Based Capsule Network. *International Journal of Image and Graphics,* p.2240001

VASUDEVAN, V., PHANISHAYEE, A., SHAH, H., KREVAT, E., ANDERSEN, D. G., GANGER, G. R., GIBSON, G. A. & MUELLER, B., 2009. Safe and effective fine-grained TCP retransmissions for datacenter communication. *ACM SIGCOMM computer communication review,* 39**,** pp. 303-314.

VERLOOP, M., BORST, S. & NÚÑEZ-QUEIJA, R., 2005. Stability of size-based scheduling disciplines in resource-sharing networks. *Performance Evaluation,* 62**,** pp. 247-262.

VILCA J, A. L., MEZOUAR Y., 2016. Adaptive leader-follower formation in cluttered environment using dynamic target reconfiguration. *Distributed Autonomous Robotic Systems* pp. 237-254.

WANG, S. & DEY, S., 2010. Rendering adaptation to address communication and computation constraints in cloud mobile gaming. 2010 IEEE Global Telecommunications Conference GLOBECOM IEEE, pp. 1-6.

WANG, T., LIU, F., GUO, J. & XU, H., 2016. Dynamic SDN controller assignment in data center networks: Stable matching with transfers. IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE, pp. 1-9.

WAHYUNI, S., SITOMPUL, O.S., NABABAN, E.B. AND SIHOMBING, P., 2021, November. Social Network Analysis Text Mining on Networks Publication Citation. In 2021 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA) (pp. 35-39). IEEE.

WEN, X., CHEN, Y., HU, C., SHI, C. & WANG, Y., 2013. Towards a secure controller platform for openflow applications. Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, ACM, pp. 171-172.

WENDELL, P., JIANG, J. W., FREEDMAN, M. J. & REXFORD, J., 2021. Donar: decentralized server selection for cloud services. Proceedings of the ACM SIGCOMM 2010 conference, pp. 231-242.

WILSON, C., BALLANI, H., KARAGIANNIS, T. & ROWTRON, A., 2021. Better never than late: Meeting deadlines in datacenter networks. *ACM SIGCOMM Computer Communication Review,* 41**,** pp. 50-61.

WILSON, B.A., KUDYSHEV, Z.A., KILDISHEV, A.V., KAIS, S., SHALAEV, V.M. AND BOLTASSEVA, A., 2021. Machine learning framework for quantum sampling of highly constrained, continuous optimization problems. Applied Physics Reviews, 8(4), p.041418.

WU, H., FENG, Z., GUO, C. & ZHANG, Y., 2012. ICTCP: Incast congestion control for TCP in data-center networks. *IEEE/ACM transactions on networking,* 21**,** pp. 345-358.

WU, H., FENG, Z., GUO, C. & ZHANG, Y., 2013. ICTCP: Incast congestion control for TCP in data-center networks. *IEEE/ACM Transactions on Networking (ToN),* 21**,** pp. 345-358.

XU, H. & LI, B., 2012. Cost efficient datacenter selection for cloud services. In 2012 1st IEEE International Conference on Communications in China (ICCC), IEEE, pp. 51-56.

XU, H. & LI, B., 2021. Joint request mapping and response routing for geo-distributed cloud services. 2013 Proceedings IEEE INFOCOM, IEEE, pp. 854-862.

YANG, H., ZHANG, J., ZHAO, Y., HAN, J., LIN, Y. & LEE, Y., 2016. SUDOI: software defined networking for ubiquitous data center optical interconnection. *IEEE Communications Magazine,* 54**,** pp. 86-95.

YANG, L., DANTU, R., ANDERSON, T. & GOPAL, R., 2004. Forwarding and control element separation (ForCES) framework.

YEGANEH, S. H., TOOTOONCHIAN, A. & GANJALI, Y., 2019. On scalability of software-defined networking. *IEEE Communications Magazine,* 51**,** pp. 136-141.

ZEHAVI, E. & LESHEM, A., 2009. Alternative bargaining solutions for the interference channel. 2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), IEEE, pp. 9-12.

ZHANG, B., HAO, J. & MOUFTAH, H. T., 2013. Bidirectional multi-constrained routing algorithms. *IEEE Transactions on Computers,* 63**,** pp. 2174-2186.

ZHANG, S. Q., ZHANG, Q., BANNAZADEH, H. & LEON-GARCIA, A., 2015. Routing algorithms for network function virtualization enabled multicast topology on SDN. *IEEE transactions on Network and Service Management,* 12**,** pp. 580-594.

ZHANG, Y.-H., LI, D., CHU, Q.-X., GUO, H.-L., ZHANG, Y. & JIANG, L.-L., 2014. Stability Analysis of Networked Control Systems Based on Industry Ethernet Ring Networks. Seventh International Symposium on Computational Intelligence and Design. pp. 93-96.

ZHU, W., SONG, M. & OLARIU, S., 2006. Integrating Stability estimation into Quality of Service Routing in Mobile Ad-hoc Networks. IEEE, pp. 122-129.

ZHANG, T., ZHANG, Q., LEI, Y., ZOU, S., HUANG, J. AND LI, F., 2022. Load balancing with traffic isolation in data center networks. Future Generation Computer Systems, 127, pp.126-141.

ZHANG, Q., YUMING, W.A.N.G., CHENG, E., LIYUN, M.A. AND YAZHOU, C.H.E.N., 2021. Investigation on In-band Interference Effect and Out-of-band Interference Mechanism of B1I Navigation Receiver. *Radioengineering*, *30*(3).

ZHANG, J., HUANG, Y. AND GAO, M., 2022. Video features, engagement, and patterns of collective attention allocation: An open flow network perspective. *Journal of Learning Analytics*, 9(1), pp.32-52.

ZHU, Y. AND DU, Z., 2021. Research on the Key Technologies of Network Security-Oriented Situation Prediction. Scientific Programming, 2021.

Appendix:

DART_MCP Algorithm I.

```
Dijkstra(s, v, graph)(Control Flow List)
q ← empty
visited ← empty
for node ← graph.nodes do
        width[node] ← 0
        previous[node] ← null
end for
width[s] ← ∞
q.add(s)
while q is not empty do
        node ← q.dequeue()
        if visited.contains(node) then continue
        end if
visited.add(node)
        for link ← graph.connectedTo(node) do
        neighbor ← link.getDestination()
        altWidth ← max(width[neighbor],
                min(width[node],link.getWidth())
        if altWidth > width[neighbor] then
                width[neighbor] ← altWidth
        previous[neighbor] ← node
        q.add(neighbor)
                end if
        end for
end while;

getAnalysePath(flow)(Sorting Entry Matching)
token  ← extractMPTCPToken(flow)
if exists(cachedPaths[token]) then
        p ← dequeue(cachedPaths[token])
        install(p)
        return
end if
linkCapacity ←  topology.getAvailableCapacity()
for i ← 0 to maxPaths-1 do
        p ← dijkstra(flow.src, flow.dst,linkCapacity)
        w ← width(p)
        for l ← p.links do
                linkCapacity[l] ← linkCapacity[l]-w
        end for
        cachedPaths[token].add(p)
end for
p ← dequeue(cachedPaths[token])
install(p))
```

Include_Constraints (G(V, E))
For each node u in G
if (g(u)+r(u)>K)
    remove node u;
    remove link (u,v);
g* = $min_{v \in NB}\, max_{k=1}^{K}([g_k(u) + \; d_k(u)]/c_k)$
for each u in NB
E(u) = ma(g(u) + d(t) + g*(e)
Z = $\sum_{u \in NB}\left[{-E(u)}/_{t}\right]$
X~uniform(0,1)
sum = 0
for each u in NB
sum = sum + $\left[{-E(u)}/_{t}\right]$
if sum > x THEN return u;
Reduce(u, v)

DART_Optimalrequire (*Matching Flow Table and Matching Operations*)
procedure *rendering* takes s_k as Switch *k*
    source: Netstab
    pt: InterferenceStability
    q: *MOCQ*
returns
    flows: globalstab
    stability: Isoefficiency
begin
    $(\alpha_k, \beta_k, \gamma_k)$ ← Map(pt, Isospeed-Isoefficiency)
    $(\sigma_{x,}\ \sigma_y)$ ← InterferenceStab(source.Netsize)
    Netpos ← $(\sigma_{x,}\ \sigma_y$, 1, 0)
    Netpos ← Map(Netpos, Isospeed-Isoefficiency)
if source.at-stability **then**
    interferencesq ← 1
else
    interferencesq ← |Netpos − edge| ^{2}
endif
    stability ←Isoefficiency (Netpos − pt)
    $(\alpha_p, \beta_p) \leftarrow (\alpha_k - \sigma_x \ *\gamma_k,\ \beta_k - \sigma_y *\gamma_k)$
    list ← source.Monte-Carlo(zz)buffer$[\alpha_p, \beta_p]$
    atten ←source.packetflow(stab.ility)/interferencesq
    flows ← atten ∗ source.globalstab
    while list ≠ nil
and not TooCongested *MOCQ*(flows) **do**
    tile ← list.first; obj ← tile.obj
if tile.γmin < $\gamma_k - \in$
and (tile.packdrop obj.HitTest(α_p, β_p, $\sigma_{x,}\ \sigma_y$))
then
    γlist ← obj.Compute$\gamma_k(\alpha_p, \beta_p$, $\sigma_{x,}\ \sigma_y)$
for each γ'_k in γlist **do**
if $\gamma'_k < \gamma_k - \in$ **then**
    $pt' \leftarrow (\alpha_p + \sigma_x *\gamma'_k,\ \beta_p + \sigma_y * \gamma'_k . \gamma'_k)$

```
        pt′ ← Map(pt′, Isospeed-Isoefficiency)
        alpha ← obj.GetAlpha(pt′, stability)
        flows ← flows ∗ ((1, 1, 1) − alpha)
        end if
      end for
    end if
  end while
end procedure
```

Algorithm2: Find the rate of convergence (source, destination, reqRange)

```
waitlist ← empty
visited ← empty
prevLink ← empty
waitlist:add(source)
visited:add(source)
dstFound ← false
while (waitlist is not empty) and (dstFound = false) do
        node ← waitlist:remove()
for link ← topologyLinks:connectedto(node) do
neighbor ← link:getDestination()
if visited:contains(neighbor) then
continue
end if
if getAccessRange(link) < requiredRange then
continue
end if
waitlist:add(neighbor)
visited:add(neighbor)
prevLink[neighbor] ← link
if neighbor = destination then
dstFound ← true
break
end if
end for
end while
if dstFound = true then
route ← empty
node ← destination
while (node ≠ source) do
route:addFirst(prevLink[node])
node ← prevLink[node]:getSource()
end while
return route and do
consider all routes links
end while
return conRoutes
else
return null
    end if.
```

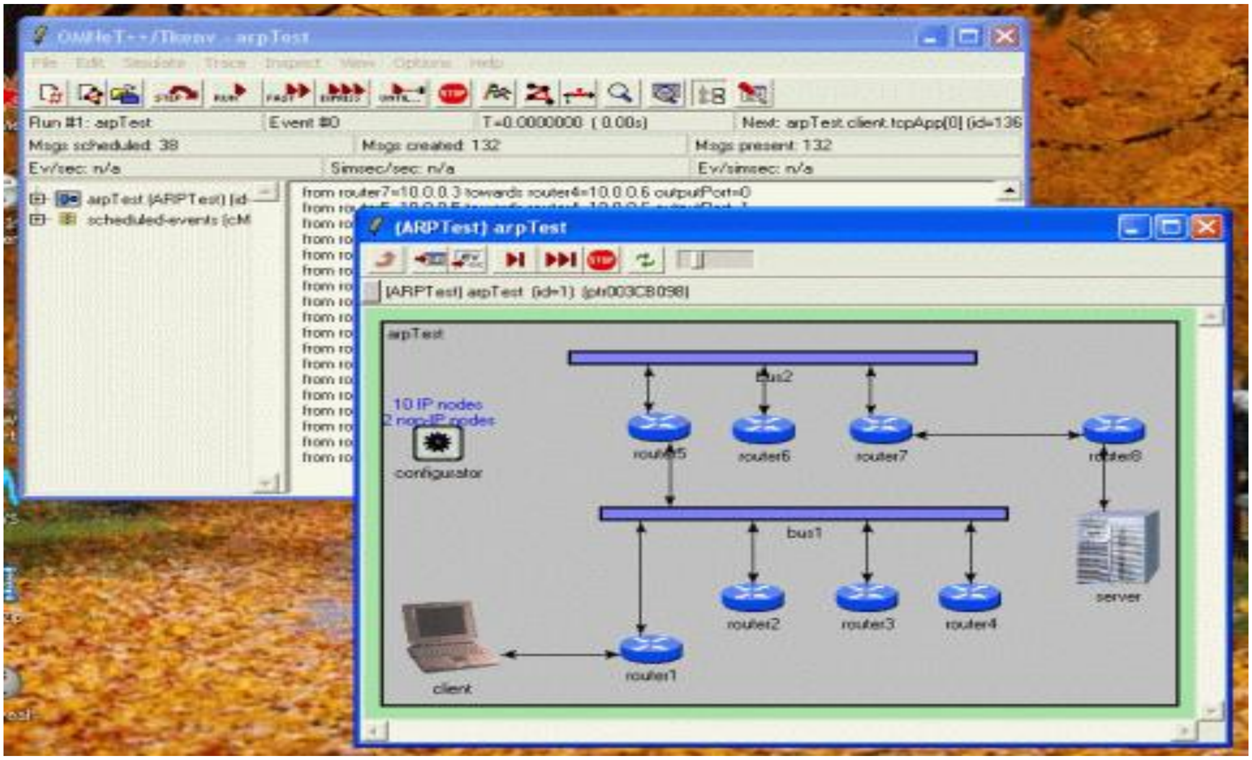
```
import ned.IBidirectionalChannel;
import inet.nodes.inet.StandardHost;
import inet.nodes.inet.Router;
import inet.networklayer.autorouting.ipv4.FlatNetworkConfigurator;
import inet.nodes.ethernet.Eth10M;
import inet.nodes.ethernet.Eth100M;

package threetierdc;
module Rack
{
    parameters:
        int N @prompt(" Nodes per rack ");
        @display("bgb=406,179");
    gates:
        inout iogate[];
    submodules:
        ComputingServer[N]: StandardHost {

        }
        AccessRouter: Router {
            @display("p=278,50");
        }
    connections:
        for i=0.. N-1 {
            AccessRouter.ethg++ <--> Eth10M <-->

ComputingServer[i].ethg++;

        }
        AccessRouter.ethg++ <--> iogate++;

}
network ThreeTierDatacenter
{
    parameters:
int N =default(1);
 int AGR =default(2);
 int CR =default(4);


submodules:
AGRouter[AGR]: Router;
CRouter[CR]: Router;
Racks[N]: Rack;
Configurator: FlatNetworkConfigurator;
connections allowunconnected:
for i=0..CR-1, for j= 0.. AGR-1{CRouter[i].ethg++  <--> Eth100M <-->
AGRouter[j].ethg++; }
```

```
for i=0..0, for j=1 ..1 { AGRouter[i].ethg++ <-->  Eth100M <--> AGRouter[j].ethg++; }
for i=0.. AGR-1 , for j=0..N-1 {AGRouter[i].ethg++ <-->  Eth100M <-->
Racks[j].iogate++  ;}
}
```

My NED file:

```
import inet.nodes.inet.Router;
import inet.nodes.inet.StandardHost;
import ned.DatarateChannel;
import inet.nodes.ethernet.EtherSwitch;
import inet.nodes.ethernet.EtherHost;
import inet.networklayer.autorouting.FlatNetworkConfigurator;

network gyte
{
    @display("bgb=457,318");
    types:
    channel geth extends DatarateChannel
    {
        datarate = 1Gbps;
    }

    channel hgeth extends DatarateChannel
    {
        datarate = 512Mbps;
    }

    submodules:
    // Routers
    routers[3]: Router {
        parameters:
            @display("p=208,272,row=id;i=abstract/router");

        gates:
            pppg[3];
    }

    // Switches
    switches[3]: EtherSwitch {
        parameters:
            @display("p=179,162,row");

        gates:
            ethg[3];
    }

    // Hosts
    ehosts[4]: EtherHost {
        parameters:
            @display("p=384,56,row");
```

```
        gates:
            ethg;
    }

    // Servers
    eservers[2]: EtherHost {
        parameters:
            @display("i=device/server;p=117,71,row");

        gates:
            ethg;
    }

    configurator: FlatNetworkConfigurator {

        @display("p=22,25");
    }

connections:
    ehosts[0].ethg <--> switchs[0].ethg[0];
    eservers[0].ethg <--> switchs[0].ethg[1];

    ehosts[1].ethg <--> switchs[1].ethg[0];
    eservers[1].ethg <--> switchs[1].ethg[1];

    switchs[0].ethg[2] <--> routers[0].pppg[0];
    switchs[1].ethg[2] <--> routers[1].pppg[0];

    routers[0].pppg[1] <--> routers[1].pppg[1];
    routers[
[0].pppg[2] <--> routers[2].pppg[0];
    routers[1].pppg[2] <--> routers[2].pppg[1];
    routers[2].pppg[2] <--> switchs[2].ethg[0];

    switchs[2].ethg[1] <--> ehosts[2].ethg;
    switchs[2].ethg[2] <--> ehosts[3].ethg;
}
```
routing



INET framework integration for simulation.