DETERMINATION OF BUFFER SIZE FOR ACCESS NETWORKS IN FIRST GENERATION OPTICAL NETWORKS

BY

BARUCH MULENGA BWALYA

A dissertation submitted to the University of Zambia in partial fulfilment of the requirements for the award of a degree in Master of Engineering in Computer Communications

UNIVERSITY OF ZAMBIA
LUSAKA

2019

DECLARATION

I, Baruch Mulenga Bwalya, declare that this dissertation is a representation of my own work and has been solely composed by myself, except where otherwise stated by reference or acknowledgement.

Signature: ……………………………………………….

Date: ……………………………………………….

CERTIFICATE OF APPROVAL

This document by Baruch Mulenga Bwalya is approved as fulfilling the requirements for the award of the degree of Master of Engineering in Computer Communications of the University of Zambia

Examiner 1: ……………………………. Signature: ……….. Date:…………….

Examiner 2: …………………………… Signature: ……….. .Date: …………

Examiner 3: …………………………… Signature: ……….... Date: ………….

Chairperson

Board of Examiners: ………………….. Signature ………… Date: ………….

Supervisor: …………………………….. Signature: ……… Date: ……………

ACKNOWLEDGEMENT

I render my sincere gratitude to my supervisor, Dr. S. Tembo. Your guidance, encouragement and patience were invaluable in coming up with the complete work for this dissertation. There were many times I felt discouraged, and almost gave up, but was uplifted by the persistent words of your fatherly encouragement.

I extend my gratitude to all my lecturers for the invaluable words of encouragement and the many lessons and life experiences that were shared during my studies. Your wisdom helped shape this dissertation.

A big thank you to my fellow students who were always at hand to help me whenever I got stuck.

To my family, I am forever grateful for everything. My sister Carol, my fiancée Alice, and all my brothers and sisters were pillars of strength and grace throughout these studies. To my parents, Bishop Dr. Dennis and Lena, I can see you in the future, and it looks better than where I am right now, this one is for you.

To the source of all things, the sustainer of life, I am eternally grateful, God. Thank you for the ability, indeed your grace is sufficient.

DEDICATION

To Dennis and Lena.

ACRONYMS

ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
BDP	Bandwidth Delay Product
CPU	Central Processing Unit
DFR	Datagram Forwarding Rate
DNS	Domain Name Service
DARM	Dynamic Random-Access Memory
DWDM	Dense Wavelength Division Multiplexing
FDL	Fibre Delay Lines
FTP	File Transfer Protocol
GSM	Global System for Mobile
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Management Protocol
IGMP	Internet Group Management Protocol
IP	Internet Protocol
LAN	Local Area Network
OPS	Optical Packet Switching
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PCM	Pulse-code Modulation
QoS	Quality of Service
RAM	Random Access Memory
RFC	Request for Comments
RTT	Round Trip Time
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SRAM	Static Random-Access Memory
TCP	Transmission Control Protocol
TV	Television
UDP	User Datagram Protocol
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing
WWW	World Wide Web

ABSTRACT

A collection of computers linked together through a communication channel are known as a computer network. Such a network facilitates communication and resource sharing between networks. The linkages are facilitated by data transmission lines, which can be optical or Ethernet cables. The data switching device can be a switch or a router.

In times of delay or congestion in the network, routers hold packets in buffers. They do so before sending the packets to the next destination. The size of a buffer has a significant impact on the performance and design of a router. The performance issues are related to network jitter and the physical size of routers due to buffer assignment standards.

Over time, it has been proven that congestion in a network is in the access or edge network and not the core of the network.

In addition to the traditional assignment of buffer size in a network, which is backed by Request for Comments and router manufactures, there have been alternative models proposed for the assignment of buffer size, but no buffer size is suitable to work for the whole length of the network.

In this study, Riverbed Academic Modeller was used, to come up with a suggested model for the calculation of buffer size for low and high traffic throughput. This model does not use link utilization as a metric for assignment of buffer size. In suggesting the model, the study establishes that for an access network of not more than 48 nodes, 32 packets and 28 packets are suitable buffers for a high and low throughput network respectively.

Keywords: Internet, Transmission Control Protocol, Buffer Size, Datagram Forwarding Rate.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE:

INTRODUCTION TO THE RESEARCH

1.1 Introduction

This chapter gives a background to computer networks and the types of computer networks available. The chapter also gives an overview of an access network and the types and sizes of buffers in optical networks. The motivation and significance of the study is presented, with the scope of the study defined. The objective of the study, research questions and organization of the dissertation are all covered.

1.2 Background to Computer Networks

A computer network is a collection of connected computers, connected for the purpose of sharing network resources. Computers can be connected using wires such as Ethernet cables or wirelessly by radio waves. Shared resources in a network can include printers and file servers as well as the Internet [2].

The Internet can be considered as a global computer network where information can be exchanged. To connect the Internet, additional network equipment such as routers, switches and firewalls are added to the computer network. The exchange of information across the Internet is made possible by common conventions called protocols, which communicate over the communication media by a set of agreed conventions so that data can move from one computer to another across the network [1, 2, 3].

Computers and their networks have become increasingly ubiquitous. A computer network today is much more than a collection of interconnected devices. Networked computers are now frequently used for collection, analysis and transmission of data and information across organizations, exchange which can be internally or across geographical regions, depending on how the computer network is structured. Such information is critical for the profitability of an entity or organization [3].

Computer networks can be logically classified into two categories, for the exchange of information and sharing of files and resources. These two classifications that connect nodes in a network are peer-to-peer and client-server networks. Computers in a network are called nodes, each with a unique address [4, 5, 6].

In a peer-to-peer network, all computers have the same status, and communicate on an equal footing. This implementation is common where there are less than 10 computers in a network and less security is required and where no centralized management of resources is needed. Each node in the network oversees its own local users, files and folder permissions. Users who want to access resources on other computers are required to have the appropriate access rights to that remote machine. With a limit of 10 users in a peer-to-peer network, this is only ideal as a workgroup for users on a small project, a home network or small office network. In such a setup, files such as a word processing or spreadsheet documents can be shared across the network, on all computers. Printers can be shared in a similar manner across a peer-to-peer network. A hub connects all the nodes in the network [4, 5, 9].

In a client-server network, the setup is such that there are clients or workstations that connect to a server or servers, to access network resources. A client-server network is interconnected by a switch or router.

Such a network has centralized network authentication, where user accounts are created on the domain and servers and nodes are joined to the domain and authentication occurs on the domain. Logins and resource access are centralized on the domain, thereby easily managing network resources. Servers usually store most of the shared network resources such a service, databases or files, which client’s access. Server’s usually have higher processing power than the average nodes.

A client-server model is highly scalable compared to the peer-to-peer network configuration as nodes can be added to the domain and users created, with the constraint being the hardware and software licensing. Other limitations can be the cost of implementation, complexity and the risk of a single point of failure if no redundancies are put in place [4, 5, 9].

Networks can also be classified according to their geography. The network calcifications are Wide Area Network (WAN), Metropolitan Area Network (MAN) and Local Area Network (LAN) [6, 7, 8].

A WAN covers a large geographical area, such as a country or continent. A WAN is largely a multiple of MANs and LANs, which join to form a WAN. A WAN consists of numerous cables which can include optical cables and copper wires, each connecting a pair of routers. Routers and switches facilitate communication within such a network.

In a WAN, data can be transferred using packet switching technologies or circuit switching. Examples of WAN technologies using packet switching are Asynchronous Transfer Mode (ATM) and Integrated Services Digital Network (ISDN). Different technologies come together in a WAN and can work collectively due to agreed protocols.

A MAN is larger version of a LAN. MAN, typically spans a metropolitan region, to cover a few hundred kilometres. Optical communication systems influenced the creation of the MAN where optical fibre technology reduced the overall cost of these systems besides the provision of higher speed transit networks operating over point to point and ring topologies without a need for a repeater. Most MANs today are synchronous optical networks (SONET)/synchronous digital hierarchy (SDH) ring networks which suffer from several drawbacks such as the high cost of SONET/SDH. The bursty nature of asymmetric data traffic is supported very inefficiently in the mostly synchronous MANs.

Unlike WANs and MANs, LANs are generally confined to a specific location, covering a floor or a building or several buildings in the same locality connected. The proximity of the connections makes it possible to use only one medium for connections, which is Ethernet. Other technologies used includes Token Rings, which are based on the ring topology. Fibre Distributed Data Interconnect (FDDI) LANs use optical fibre with an improved token ring mechanism based on two rings flowing in opposite directions. LAN technology is usually less expensive to implement than MAN and WAN technologies as these technologies are implemented in a small area [7, 9].

A LAN accesses the WAN, usually through a MAN, by connecting through a router or switch from the first or last mile of the network. A LAN is therefore an edge network or an access network. An access network is a user network that connects users or subscribers, either home or business, to a service provider, for onward connection to other networks such as the Internet. An access network serves as the “last mile” as well as “first mile” of the information flow.

Some types of access network include Ethernet, wireless LAN, fibre optic such as fibre to the home (FTTH) and Asymmetric Digital Subscriber Line (ADSL) [10, 11]. When optical fibres are used in a WAN to connect routers and switches, the network is called an optical network. The replacement of copper cables by optical fibres in a WAN has served to increase data transfer speeds and transmission capacity. Optical access networks are networks that seek to satisfy the growing demand for high speed Internet in the last mile of the network, making optical fibre get closer and closer to the subscriber.

Optical networks have still retained the electronic switching of data at the routers, even with a full replacement of copper wires which transmitted data electrically, with optical cables which transmit data optically. At the ends of the physical fibre link connections, electronics take over all the necessary switching, routing and bit processing procedures using different technologies. Some of these technologies in access networks are Wi-Fi, Long Term Evolution (LTE), Asymmetrical Digital Subscriber Line (ADSL) and Worldwide Interoperability for Microwave Access (WiMAX).

Technologies such as passive optical network (PON) based FTTH access networks are used in a point to multipoint setup, where fibre goes to the premises network architecture in which unpowered optical splitters are used to enable a single optical fibre to serve multiple premises. Gigabit PON (GPON) FTTH architecture offers converged data and voice services of up to 2.5 Gbps. Some of its components are an Optical Line Terminal (OLT) which is the main element of the network and is usually placed in the Local Exchange and is the engine that drives FTTH system. It performs traffic scheduling, buffer control and bandwidth allocation. An Optical Splitter splits the power of the signal. An Optical Network Terminal (ONT) is deployed at the customer's premises. ONTs are connected to OLTs by means of optical fibre and no active elements are present in the link [140]. However, this study does not focus on FTTx access networks, it focuses on an access network found in a first generational optical network, where packet switching is performed in an electronic domain.

Users connecting through the access network transmit and receive electrical signals which are only converted to and from optical signals for the sake of transmission between point-to-point links usually via a router or switch. This conversion is transparent to end users, who do not notice the change of data transfer modes.

All the intermediate nodes process only electrical signals and thus do convert the signals from electronic to the optical domain and back, which is unavoidable after every hop from source to destination. Such types of networks are called first generation optical networks [12]. A second-generation optical network is one where all or some of the optical switching at the routers is done in an optical domain.

When a router is switching packets, there are times when the rate of input of packets into the router does not match the rate at which packets are output from the router. In such a case, packets are dropped from the router and Transmission Control Protocol (TCP) mechanisms must ensure the retransmission of the dropped packets so that they reach the receiver. To minimize such occurrences, routers are fitted with buffers. A buffer is a memory space, in a switch or router, set aside for storing packets awaiting transmission over networks or storing packets received over networks during transmission delays in the network. These memory spaces are either located on the network interface card or in the computer system that holds the card [13].

Buffers can be measured in different ways, such as by maximum number of packets that can be held in queue or switched, the number of bytes or queueing time limit [18, 20, 21, 22]. In [22], routers for two manufactures were compared, and it was observed that one router measured the buffer in packets while the other measured it in milliseconds.

Buffers can be either electronic or optical. An electronic buffer consists of electronic components while an optical buffer is made by introducing delays in the fibre transmission medium, so that packets take longer to reach the router, giving enough time for the router to process and output the packets into the network for onward transmission. By creating a difference in the rate of input and output of packets, an optical router avoids packets drops [14, 28].

All optical packet switching (OPS) is theoretically capable of switching packets in the optical domain but this technology has not yet matured [14]. The immediate buffer solution for a second-generation optical network is optical burst switching (OBS).

Both OPS and OBS suffer from output port contention resolution. Ways of countering packet dropping due to output port contention include buffering, defection routing and wavelength conversion. Optical buffers use fibre delay lines (FDL), to delay the light. This is realized in three ways, namely, input buffering, output buffering and shared buffering. Shared buffering is the most advantageous as it provides both switching and buffering [14, 28].

OPS's have the challenge of non-feasibility due to the absence of optical Random-Access Memory (RAM), which could be the equivalence of electronic RAM. Hence an alternative to this is FDLs, which are very limited in storage due to the physical constraints like dispersion, crosstalk and noise [15].

Other ways of buffering in first generational optical networks include the bandwidth delay product, which uses the round-trip time of a packet and capacity of a link as a product, to come up with the buffer size for the network. There is a request for comments (RFC) recommendation that buffer size should be at least 250mm of RTT and manufactures use this as a standard in today's networks [18, 19, 32].

Other strategies to assign buffer size in a network include a hybrid approach for packet switching between an optical and electronic buffer for the buffering of contending packets. A Silicon based CMOS RAM has been used as a storage medium for optical packets at data rates of up to 40 Gbps, by using a combination of optical and electrical components. Such a design for routing traffic offers long storage times, large capacity and random access at arbitrary times [15, 27].

One of the ways in which traffic is routed through network nodes in the core of the network is through multipath routing. Multipath routing is an alternative approach to transmitting traffic through a single channel, which distributes traffic among several good paths instead of routing all traffic along a single best path. This kind of routing can be fundamentally more efficient than a single-path routing protocol. Its purpose is to reduce congestion in the network, by deviating traffic to unused resources in a network, thus improving network utilization and providing load balancing. This is one of the reasons for little or no congestion in the core of the network compared to the edge of the network [16, 17].

1.3 Statement of the Problem

Congestion is an old problem on the Internet, appearing in various forms with different symptoms, one of which is latency. Today's networks continue to suffer from unnecessary latency and poor system performance due to congestion.

Latency is made up of transmission delays, processing delay and queuing delay. Transmission delay is the time it takes to send a packet from the source to the destination across communication links. Processing delay is the time spent by the packet at each network element and queuing delay is the time a packet spends while waiting to be processed usually at the routers. Queuing delay is higher in networks where buffers are very large.

Large buffers have been inserted all over the Internet without enough thought or testing. Large buffers lead to bufferbloat, which is the existence of excessively large and frequently full buffers inside the network. Large buffers appear in networks largely due to the assignment of buffers in the network by using the bandwidth delay product (BDP).

Large buffers disrupt the fundamental workings of congestion avoidance algorithms of the Internet. Created in the 1980's, the design choice for Transmission Control Protocol (TCP) congestion control algorithms, interpret packet loss or packet delay in a network as congestion, which triggers congestion avoidance mechanisms.

The last mile continues to be a major bottleneck in the Internet. Its low bandwidth and flexibility prevent the deployment of new services and the development of new applications. Previous studies, measuring hosts connected to Digital Subscriber Lines (DSL) companies revealed the problem of over buffering in the broadband edge of the network. There were excessive latencies in both the downlink and uplinks in all broadband technologies [74].

The use of optics in the last mile can ease congestion in access networks but traditional optical solutions based on point to point architectures are expensive for access configurations [8]. Controlling buffer size therefore, makes it possible for one TCP to work well everywhere as opposed to attempting to create a version of TCP specifically for access links [74, 139].

BDP sized buffers are not appropriate for a highly multiplexed core network. Although the rationale for maintaining a BDP buffer still applies at the network edge, where a single flow can congest a link, the problem is in determining the appropriate BDP. An access link that has a constant bandwidth and a buffer size set to 100 ms might still be 3 to 10 times too large. A focus on TCP tuning and Active Queue Management (AQM) algorithm failed to improve network performances in access links and has not been successful [74, 75, 76], with successes recorded in controlling buffers.

In computer networks, it has traditionally been known that the available bandwidth, delay and jitter between two end-to-end devices have been used as a parameter that can give a rough idea of the expected quality of service (QoS). It has come to be understood that QoS is also affected by the behaviour of the intermediate router buffer, which is mainly determined by its size and its management policies. The buffer may therefore cause different packet loss behaviour and may also modify some of the QoS parameters [18]. QoS can be enhanced by a good AQM algorithm that considers as an input, the rate at which data leaves the buffers, without which bufferbloat will be hard to defeat [74].

Mid and low-end routers, which do not implement advanced traffic management mechanisms, are usually used in access networks. Due to this, the design characteristics of router buffers and the implemented scheduling policies are of primary importance to ensure the correct delivery of the traffic of different applications and services [18]. When the size and behaviour of buffers are known, some techniques can be used to improve link utilization. There is a relationship between buffer size and link utilization, since an excessive amount of memory would generate a significant latency increment when the buffer is full. Conversely, a very small amount of memory in the buffer would increase packet loss in times of congestion [18, 34].

Buffer size has an influence on real time services offered on a network like the Internet. In [23, 24], an online game with very tight real-time requirements was studied and the results show that there was a mutual relationship between the size and policies of the buffer and there was subjective quality which depended on delay and jitter in this case. The study shows that tiny buffers, which are tens of packets of buffer, are more adequate in maintaining game quality to acceptable levels, as opposed to big buffers which add to delay and jitter, a case that is not acceptable for the game experience. The game behaviour for a loaded game server, multiplayer game show that current game designs target the saturation of the last narrowest, last mile link. Such interactivity demands that the routers are designed with enough capacity to manage any bursts in traffic without delay. However, current routers are designed for bulk data transfers with large packets.

The bottleneck in optical networks is the absence of all-optical switching and routing. With the large amount of data transfer capacity offered by the current fibre networks, the processing and buffering at switch and access points on the network is causing a bottleneck problem [10, 25]. Current routers use large capacities of electrical RAM to resolve contention and congestion in the network. An example is the state-of-the-art Cisco CRS-1 core router, which is based on line cards that operate with 2 GB of memory. Such a capacity is not feasible with any of the proposed optical buffering approaches. If access links are slower than the core network and traffic is smoothed, only five packet buffers per output port are needed for 80% link utilization [25, 26].

A proposed hybrid solution for contention resolution using both optical and electronic buffering has a challenge of loss of packet and complexity of design, which limits packets buffer size of less than 10 bytes, thereby limiting the maximum load of the network [27].

In electronically packet switched networks, the appropriate size estimation of a buffer in a network is a critical and still open research question [29, 30, 31].

Large buffers in a network with electronic routers lead to packets being held in the network switch or router buffer queue for too long, therefore causing TCP to respond far slower and most likely too late to congestion in the network. This has an effect of opening the congestion window and incorrectly overestimating the allowed link capacity size and leading to more traffic from the source, especially in times where the network is already congested [33]. This therefore shows that over buffering in the network can lead to increased application latency and jitter, resulting in less deterministic and unpredictable application. Today's data centres are characterized by low latency, high throughput and an increasing East-West traffic flow. These networks have very low Round Trip Times (RTTs) due to the flatter network architectures with fewer hops between servers and moving to high speed 10Gbs connectivity from server to network and 40Gbs in the core of the network. The need to reduce network latency between VMs and applications is becoming important [33].

Over the last decades, there have been several buffer architectures that have achieved good results, but whose success has been limited by either large component counts or component complexity and a lack of industrial application [25, 18, 20, 21, 22].

Wavelength Division Multiplexing (WDM) is today a dominant technology for use in optical networks. This technology has significantly increased the capacity of fibre by allowing simultaneous transmission of multiple wavelengths, each operating at rates of up to 10 Gb/s. Systems with well over 80 wavelengths on one cable are available, with capacities reaching over 1 Tb/s. Packet switching over these links will require that the transmission speeds over the links is matched by equivalent switching capacities in the nodes. Current packet switches perform data processing in the electronic domain, and there is a growing discrepancy between channel capacity and switching capacity. All-optical switching can alleviate this problem, but this solution is not yet mature and feasible for application [14, 42, 43, 44].

Most of the buffer sizing work has focused on maintaining a link utilization of nearly 100%. This has raised concerns as this is not the only metric to consider when assigning buffer size in the network. Other metrics to consider may be the packet loss rate and TCP throughput [45]. In [45], it concludes by pointing out that it might not be possible to derive a single universal formula to dimension buffers at any routers interface in a network. Instead, a network administrator should decide, by taking into account several factors such as flow size distribution, nature of TCP traffic and the input/output capacity ratios.

An optimal operating point for a network maximizes delivered bandwidth while minimizing delay and loss, not only for single connections but for the network. Finding that optimal operating point has been elusive, since any single network measurement is ambiguous. Network measurements are the result of both bandwidth and propagation delay, and those two cannot be measured simultaneously [72].

With the issues raised in this section, this study seeks an alternative model for the assignment of buffer size in a network that switches packets electronically. The study approaches the buffer size problem by seeking to understand the relationship between buffer size and network throughput or link utilization, from which an estimated model for buffer size in access networks can be presented, as congestion is largely in access network. The focus of the study is on how the input and out ratios can be used to simplify the buffers size problem.

1.4 Significance of Research

Growing demand for high speed Internet is a primary driver for new access network technologies. The demand for Internet based e-commerce and high bandwidth demanding applications is increasing daily. Optical communication provides high speed and high bandwidth networking as a solution to the high network and capacity demands. The wide bandwidth available in low attenuation in the optical fibre can be divided into several independent wavelength channels based on the packet switching technology. The important application parameters of packet switching are control of packets in the network, packet synchronization, clock recovery, packet recovery, packet routing, and contention resolution [28, 140].

The major issue in all these aspects for successful packet switching and routing is contention resolution at the output of the routers. In the electronic domain, contending packets are stored in RAM for future processing while in the optical switching domain, which does not have the electronic RAM equivalent, packets can be buffered in fibre delay lines. Packet switching in an optical system is still assumed that packets are destined to their respective destinations on a first come first served basis scheduling policy. However, fibre is a nonlinear medium, therefore, there is no guarantee that there will no continuous recirculation of packets in the delay lines. The size of a buffer plays a big role in how packet switching is handled in a network [28].

Large buffers in network switches or routers can significantly add to the system cost and operational complexity. This takes on greater significance as networks approach 10/40GbE speeds and port densities increase in data centres networks. Therefore, having a network with large off-chip packet memory buffers adds significant cost to the switch due to the complexity of memory controllers required to address both very high bandwidth and very high density [33]. As the main bandwidth bottleneck in today’s networks is the access segment [46], this study suggests an alternative assignment of buffer size in an access network. This alternative contrasts with inclusion of fibre access technologies such as passive optical networks (PONs) and implementation of wavelength division multiplexing (WDM), which are indispensable solutions but come at an added cost [47].

The use of link utilization and the measure of packet loss in a network as a sign of congestion has had its challenges. TCP congestion control mechanisms and the use of BDP to assign network buffers has led to large buffers in the network [28, 45]. This research suggests a buffer assignment method which does not focus on link utilization or packet drops in the network for assignment of buffer size. This has proved to be a challenge in high speed links, like first generation optical networks, which have a large bandwidth delay product (BDP) for buffer size assignment, and the alternative can be found by looking at packet loss rate and network throughput [45, 49]. A buffer size assignment that does not depend on link utilization will reduce contention resolution at the output ports and therefore lead to low packet drops in the network.

1.5 Aim of Research

The main aim of this study is to explore the possibility of developing a model that assigns buffer size in access networks independent of link capacity as a metric of assignment. The buffer size can then be assigned based on the number of packets that a router can switch per second, which is the network throughput or the output and input capacity ratios at the routers. This is because an increase in the capacity of the links has caused a lag in the switching of packets by routers using the BDP model [45].

1.6 Objectives

The objectives of this study are as follows:

1) To assess the buffer sizing methods, their challenges and determine the suitability of current buffer sizing techniques for first generational optical networks.
2) To determine whether a homogenous buffer size is suitable for a first-generation optical network in heterogeneous traffic.
3) To devise and design a model that can be used to assign buffer size independent of link capacity where link utilization is not a metric for the assignment of buffer size.

1.7 Research Questions

This research will be guided by the following research questions;

1) What is the current buffer sizing model for the assignment of router buffer size in the Internet?
2) What are the challenges of the current buffer sizing techniques used in the Internet today?
3) How suitable is a homogeneous buffer size for a first-generation optical network with heterogeneous traffic?
4) How can a minimal buffer size for first generation optical network be modelled?
5) How can we devise a method for the assignment of buffer size in an access network in a first-generation optical network?

1.8 Scope of Research

The scope of this research is to devise a method for buffer size assignment in an access network in a first-generation optical network in which the network is transmitting heterogonous traffic.

1.9 Research Contributions

The major contribution to this study is a method of assigning buffer size in an access network that shifts the focus from basing assignment on link utilization and considering the number of packets that a router is switching per second. The classic congestion control algorithms used by Transmission Control Protocol (TCP) are not well adapted to today's very high-speed links, which still use large bandwidth delay product (BDP) for buffering [49].

1.10 Organization of the Dissertation

Chapters in this dissertation are organized into six chapters. Chapter 1 is the Introduction to the Research and gives a background to computer networks and buffer sizing in general. The chapter also discusses the significance of the research, its aim, scope, objectives and research questions that assist in addressing the objectives of the study.

Chapter 2 gives a theoretical framework of the study by citing definitions and practical applications of common terms used in this research. It gives a detailed view of congestion control in the network and network definitions that assist in understanding the problem of buffer sizing in a network. The chapter also reviews similar work done on the problem of buffer sizing, the performance of the suggested solutions and challenges with applicability.

Chapter 3 gives an overview of the research methodology. The chapter highlights the methodology chosen and the methods used to achieve the objectives. Chapter 3 also discusses the tools used in collecting and processing data in this research and gives details on the design and implementation of the minimum buffer size model that has been designed.

Chapter 4 discusses the results from the methodologies used in this dissertation. An analysis of the results is presented in this chapter. Results were arrived at after following the stated methodology in Chapter 3. Chapter 5 contains the conclusions and recommendations for this study, further works and possible applicability of the model.

1.11 Summary

In this chapter, we looked at the basic introduction of the work in this dissertation. We began by looking at the basic concept of a network, what an access network is and an understanding of what a first generational optical network is. The current buffer sizing models and suggested solutions have also been presented in this chapter.

The significance of the research, as well as the aim, objectives and research questions have also been presented in this chapter. This chapter closes with an outline of the dissertation and the summary.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter reviews the literature to give the background theory related to our work. The chapter starts by exploring congestion control and avoidance mechanisms in networks, then reviews the current buffer sizing techniques or methods and how these relate to methods of buffer size assignment for first generational optical networks. Suggested solutions to the current approach are examined, after which current methods are related to their viability in first generation optical networks.

2.2 Network Congestion Control and Avoidance

The Internet works by using a protocol called Transmission Control Protocol/Internet Protocol (TCP/IP). This is the underlying communication language of the Internet, enabling computers to talk to one another on the Internet by compiling packets of data and sending them to the correct destination. TCP is the dominant protocol in modern communication networks, which handles issues of reliability, flow control and congestion control [48, 49].

TCP is a connection oriented, bidirectional, point to point transport protocol with reliable, in-order data delivery. TCP transports a serial byte stream between applications and manages the recovery from erroneous, lost or duplicate segments. In the source, the byte stream is fragmented into appropriately sized segments with a maximum segment size (MSS), according to the maximum transmission unit (MTU) of the path or link. Each type of network has an MTU, which is the largest packet that it can transfer. If a datagram or packet received from one network is longer than the other network's MTU, it is necessary to divide the datagram into smaller fragments for transmission, a process called fragmentation. The resulting packets are passed to the IP layer and reassembled at the destination. Figure 2.1 shows the structure of an application byte stream that is transported in a TCP segment and encapsulated in an IP packet. The TCP and IP header can be extended by header options. TCP options are frequently used during the initial state synchronization by the three-way handshake [49, 50].

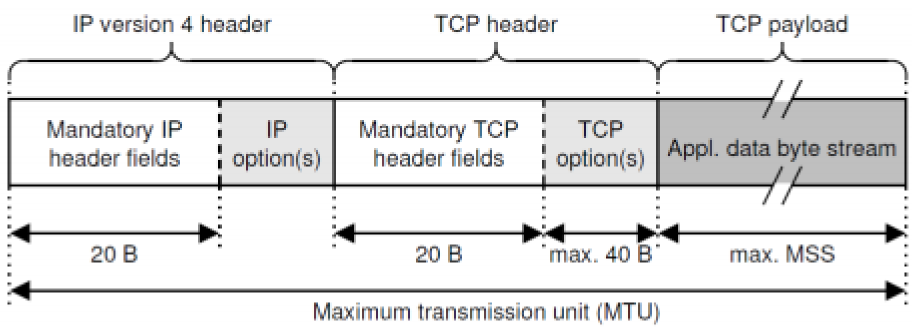

Figure 2.1: Structure of an application TCP byte stream [50].

User datagram protocol (UDP) is another protocol which is used in the Internet. UDP offers connectionless, unreliable transport of datagrams across the network. Unreliability is a means used by UDP to mean that there are no techniques in the protocol for verifying that the data has reached the destination correctly without any errors. UDP is basically a multiplexing layer on top of IP. UDP is used by applications that need multicast or broadcast delivery, basically for services not offered by TCP. It is preferred when the amount of data being transmitted is small, where the overhead of creating connections and ensuring reliable delivery by using TCP, may be greater than the work of re-transmitting the whole dataset when need arises. UDP gives application programs direct access to a datagram delivery service, like IP.

Unlike UDP, TCP is a window-based protocol that realizes congestion avoidance, flow control and congestion control. Congestion occurs in a communication network when resource demands exceed the capacity of the network or computer resources. This can occur in instances where there are too many packets present in a part of the subnet and it subsequently leads to performance degradation. A network can also be congested if there is an overload on the network by too many packets trying to access the same routers' buffer, resulting in packets being dropped, causing excessive queuing delays and packet losses in a network. [49, 51, 52, 53].

TCP does not assume any explicit knowledge about network internals and other sessions. To control congestion, TCP uses a form of end-to-end flow control. When a sender sends a packet, the receiver acknowledges receipt of the packet. When the receiver receives an acknowledgment, a sender knows that the packet reached its destination. The sender can then send new packets on the network, where both the sender and receiver agree on a common window size for packet flow. The window size represents the number of bytes that the source can send at a time. This size varies according to the condition of traffic in the network to avoid congestion. This way, a sending source can use the acknowledgement arrival rate as a measure of network congestion [56, 61].

During congestion, actions need to be taken by both the transmission control protocols and the network routers to avoid a congestion collapse and ensure network stability, throughput efficiency and fair resource allocation to network users. Congestion control refers to a technique that can either prevent congestion before it happens or remove congestion after it has happened. TCP does this by maintaining a congestion window (*cwnd*), which indicates the maximum amount of data that can be sent into the network without being acknowledged, which is basically the number of packets that can be outstanding at any time. It is incorporated into the transport layer and controls the number of packets put into the network [51, 53].

Additive Increase Multiplicative Decrease (AIMD) is the traditional congestion control mechanism of the Internet. It is a feedback control algorithm which is the primary mechanism for adjusting the rate of a TCP flow. For each connection, TCP maintains a congestion window (*cwnd*) limiting the total number of unacknowledged packets that may be in transit end-to-end. AIMD allows a sliding window sender to adaptively find a window size equal to the BDP of the network. AIMD evaluation was confined to a simulated network and required changes to the routers, something that was quite undesirable as it went against the ethic of minimalism in the routers [59].

When the *cwnd* is reached, meaning the number of unacknowledged segments is equal to the *cwnd*, the sender stops sending data until more acknowledgements are received. If the acknowledgements are received after some timeout has passed, then it is considered as an indication of packet loss and the segments are retransmitted. The *cwnd* variable is not advertised or exchanged between the sender and receiver, but is a private variable maintained by each host.

When a connection is set up between a sender and receiver, the *cwnd* is set to a small multiple of the maximum segment size (MSS) allowed on that connection. The AIMD algorithm then goes into additive increase, where the *cwnd* is increased by a fixed amount for every round-trip time (RTT) that no packet is lost. Once the *cwnd* window is reached and there is congestion in the network, multiplicative decrease decreases the *cwnd* by a multiplicative factor every RTT that a packet loss occurs. This creates a "*sawtooth*" *cwnd* pattern [51, 56, 57].

There are two main parts to AIMD algorithm. Slow start, where TCP probes for available network resources with a small *cwnd* and congestion avoidance, where TCP probes for unused bandwidth conservatively by increasing *cwnd* by one segment per RTT without delayed acknowledgement [59, 61].

Slow start allows a sliding window sender to discover the “right” value of the window size given the links’ capacity, its minimum round-trip time and the number of senders on the network. The sender starts off with an initial window of packets and on every acknowledgement (ACK), increases the window by 1 packet to reflect the fact that the previous window size worked, and it is fine to increase the window a little but further.

As a start, at time 0, the sender sends out an initial window worth of packets, which for simplicity can be 1. An RTT later, the sender receives an ACK for the first packet. The ACK gives the sender permission to do two things;

1. As per sliding window (Stop-And-Wait) protocols, it allows the sender to release one more packet because one packet has been ACKed and the sender is now permitted to have one more unACKed packet.
2. As per Slow Start, the sender can increase the window itself by 1. Hence, the window is now 1 + 1 = 2, and the sender is permitted to have two unACKed packets. This allows the sender to release an additional packet. At the end of each RTT, the sender releases two packets. These two packets are ACKed after another RTT. Each of these two packets in turn trigger two new packets, one each for sliding window and slow start. This means a total of 4 packets are sent out at time 2RTT. Even though the *cwnd* only increases by 1 on every ACK, it doubles every RTT.

The *cwnd* grows exponentially with time when the sender is in Slow Start. When the bottleneck link for a sliding window flow drops a packet from this flow, the *cwnd* stops growing exponentially. This is a congestion signal which tell the sender that there is something wrong with packet transmission. The sender knows about this by receiving an acknowledgement of packets sent later or when the router places a bit in the packet header indicating that a packet has been dropped.

Once the sender detects that there is a loss of packets, like timeouts and duplicated acknowledgements, it cuts the *cwnd* size in half to the last window size sent. After this phase, TCP enters a congestion avoidance node, where more gentle changes are made to the window size, compared to Slow Start phase.

In congestion avoidance, the sender continuously tries to do two things. First it increases the window size in case spare link capacity has opened up because an old sender left the network and secondly it tries to decrease its window size in case spare link capacity went down because a new sender just joined the network. In this phase, the sender increases its window size by 1 every RTT, as opposed to 1 every ACK in Slow Start. Effectively, in congestion avoidance, this means the window increases *additively* by a constant amount. On the other hand, in Slow Start, the window increases *multiplicatively*, doubling every RTT. In practice, to ensure a gradual window increase, the window size increases by *1/W* every time it receives an ACK, where *W* is the current value of the window. This translates into the same 1 unit increase every RTT. In congestion avoidance, the sender decreases its window every time it detects a loss, wither through a timeout or by the receipt of an acknowledgement for a later packet. It decreases its window size in the same way as Slow Start, by halving the window.

Packet loss is not a very specific example of a congestion signal as this depends on how big the routers buffer was. Big buffers will lead to late packet loss notifications, as these notifications would only come after the buffer has built up to a point of an extremely large queueing delay [58, 59, 60]. This can lead to a congestion collapse.

Congestion control is one of the central problems of networking that has remained important despite changes in both the underlying link technology that supports the Internet and the applications running on top of the Internet [59].

To enhance congestion detection in a network, Drop-Tail routers drop incoming packets whenever their output buffer overflows, as a congestion signal to notify responsive transport end-points. This is known to produce bursty losses and a bias against flows with long RTTs and small packets.   Modern Active Queue Management (AQM) capable routers, such Random Early Detection (RED) schemes, can detect congestion even before buffer overflow occurs. RED capable routers monitor the queue length and the speed at which it increases, such that if the queue is below a lower threshold, no packets are dropped, and if above a threshold, all packets are dropped. When the queue length is in between the two thresholds, packets are dropped with a certain probability. With RED, it is very likely that the packet loss probability of a flow is proportional to its packet rate and the congestion signals, and the congestion signals can be distributed among different flows more fairly. There are other queuing mechanisms in literature that offer better fairness with higher complexity [61, 62].

There have been numerus ideas for improving TCP over the years. Some of the ideas have been adopted by mainstream operations. The recent focus has been improving TCP's behaviour with LongFatNetwork. It has been proven that current congestion control algorithms limit the efficiency in network resource utilization. Suggested improvements include changes in the size of the congestion window and a requirement extra feedback from the network. These are largely divided into explicit congestion control protocol that use explicit feedback from the router and implicit congestion control protocols that rely on implicit measurements of congestion such as loss or delay in the network [71].

There is a variety of TCP congestion control algorithms, each suited for a purpose. These TCP variants mostly adhere to the underlying framework of sow-start, AIMD, retransmit timers and ACK-clocking. None of these changes alter the fundamental underlying dynamics of TCP congestion control. These variants assist in avoiding unnecessary retransmit timeouts, correct unnecessary fast retransmits and retransmit timeouts resulting from disordered or delayed packets. They also reduce unnecessary costs in delay and retransmissions which are associated with the mechanism of congestion notification [63].

In 1988, Van Jacobson [64], introduced TCP congestion control after the Internet suffered a congestion collapse. TCP Tahoe was implemented as a congestion control algorithm, which added several new algorithms and refinements to earlier TCP implementations. New algorithms added to TCP Tahoe were *Slow-Start, Congestion Avoidance and Fast Retransmit*. These refinements catered for modifications to the RTT estimator which was used to set retransmission timeout values. With fast retransmit, after receiving a small number of duplicate acknowledgements for the same TCP segment, the data sender infers that a packet has been lost and retransmits the packet without waiting for a retransmissions time to expire, which leads to higher channel utilization and connection throughput [63, 64, 65].

Another variant of TCP algorithm is the TCP Reno, whose implementation retained the enhancements incorporated into TCP Tahoe but modified the fast-retransmit operation to include *Fast Recovery*. This new algorithm prevents the communication channel from going empty after a fast Retransmit, thereby avoiding the need to Slow-Start to re-fill it after a single packet loss. Fast recovery operates by assuming that each duplicate ACK received represents a single packet having left the pipe. Thus, during Fast Recovery, the TCP sender can make intelligent estimates of the amount of outstanding data. A TCP sender enters fast recovery after receiving an initial threshold of duplicate ACKs. Once the threshold of duplicate ACKs is received, which is generally 3, the sender retransmits one packet and reduces its congestion window by one half. After entering fast recovery and retransmit a single packet, the sender effectively waits until half of a window of duplicate ACKs have been received, and then sends a new packet for each additional duplicate ACK that is received. Upon receipt of an ACK for new data, the sender exits Fast Recovery. TCP Reno significantly improves upon the behaviour of TCP Tahoe when a single packet is dropped from a window of data [63, 66].

An extension of TCP Reno is TCP New Reno. The advantage over TCP Reno is that it can detect multiple packet loss and it does not leave the fast recovery until it receives acknowledgements of all packets which are present in the window. The fast recovery phase proceeds as in TCP Reno, when a fresh acknowledgement is received, then there are two cases which happen. First, if it acknowledges all the packets which are outstanding, when it enters in fast recovery, then it exits fast recovery and set *cwnd* to *ssthresh* and continues congestion avoidance. In the alternative, if the acknowledgement is an incomplete acknowledgment, then it deduces that the next packet in line was lost and it retransmits that packet and sets the number of duplicate acknowledgments received to 0. The advantages of TCP New Reno are that it can detect multiple packet loss and its congestion avoidance mechanism is very efficient and utilizes network resources much more efficiently. TCP New Reno has fewer retransmits because of its modified congestion avoidance slow start. [68, 69].

With TCP Selective Acknowledgment (SAK) the congestion control algorithm is an extension of TCP Reno's congestion control algorithm. These use the same algorithm for increasing and decreasing the congestion window and make minimal changes to the other congestion control algorithms. Adding SACK to TCP does not change the underlying congestion control algorithm. TCP SACK preserves the properties of Tahoe and Reno, of being robust in the presence of out-of-order packets and uses retransmit timeouts as the recovery method of last resort. The main difference between TCP SACK implementation and Reno is in the behaviour when multiple packets are dropped from one window of data. TCP SACK requires that packets should be acknowledged selectively. This is an option enabling the receiver to tell the sender the range of non-continuous packets received. Without SACK, the receiver can only tell the sender about sequentially received packets. The sender uses this information to retransmit selectively only the lost packets. During fast recovery, SACK maintains a variable called pipe that represents the estimated number of packets outstanding in the network, which feature is absent in TCP Reno and New Reno. When pipe < *cwnd*, then it sends data and set pipe = pipe + 1 However, when the sender receives an acknowledgement from the receiver then set pipe = pipe – 1. In this way, the sender only retransmits data when estimated number of packets in the path is less than the congestion window, thereby transmitting all the outstanding data in the network. When all the outstanding packets are acknowledged, SACK exits fast recovery and enters the next phase of congestion avoidance. Use of the pipe variable decouples the decision of when to send a packet from the decision of which packet to send. The sender maintains a data structure, called the scoreboard that remembers ACKs from the previous SACK options. When the sender can send a packet, it retransmits the next packet from the list of packets inferred to be missing at the receiver. The SACK sender has a special handling for partial ACKs. These are ACKs received during Fast Recovery that advance the ACK number field of TCP header, but do not take the sender out of fast recovery. The sender decrements pipe by two rather than one for partial ACKs, and the SACK sender never recovers more slowly than a slow-start [63, 65, 69, 70].

TCP Vegas adopts a more sophisticated bandwidth estimation scheme, which uses the difference between the expected and actual flow rates to estimate the available bandwidth in the network. The idea is that when the network is not congested, the actual flow rate will be close to the expected flow rate. The converse being the actual flow rate being smaller than the expected flow rate. TCP Vegas uses the difference in flow rates to estimate congestion levels in the network and to update the window size accordingly. The difference in the flow rates can easily be translated into the difference between the window size and the number of acknowledged packets during the round-trip time to come up with an equation *Diff = (Expected – Actual) BaseRTT,* Where expected is the expected rate, Actual is the actual rate and BaseRTT is the minimum round-trip time. In essence, the sender computes the expected flow rate, then the sender estimates the current flow rate by using the actual round-trip time. Then the sender, using the expected and actual flow rates, computes the estimated backlog in the queue from diff = (Expected - Actual) BaseRTT. TCP Vegas tries to keep at least alpha packets but no more than beta packets in the queue. The reason behind this is that TCP Vegas attempts to detect and utilize the extra bandwidth whenever it becomes available without congesting the network. This mechanism is fundamentally different from that used in TCP Reno. TCP Reno always updates its window size to guarantee full utilization of the available bandwidth, which leads to constant packet losses, whereas TCP Vegas does not cause any oscillation in the window size once it converges to an equilibrium point [63, 67].

In [63] it is concluded that TCP Vegas and TCP SACK have better performance than TCP Reno. TCP Vegas archives higher throughput than TCP Reno and TCP SACK for large loss rates. TCP SACK is better when more than one packet is dropped in one window. TCP Vegas causes much fewer packet retransmissions than TCP Reno and SACK. TCP Vegas also leads to a fair allocation of bandwidth for different delay connections, whereas TCP Reno and SACK show a bias against long delay connections. TCP Vegas is one of the smoothest TCP algorithms next to Cubic, as it increases the timeout delay for packets, which allows more to be received, but at a higher rate. It also has set timeouts, which help with speed because it’s constantly being refreshed.

TCP Cubic is one of the best, most recommended TCP option. It is less aggressive and inflects the windows prior to the event.

Low Priority TCP is a distributed algorithm whose goal is to utilize the excess network bandwidth as compared to fair share of bandwidth as targeted by TCP. The key mechanisms unique to TCP LP congestion control are the use of one-way packet delays for early congestion indications and a TCP transparent congestion avoidance policy.

TCP Binary Increase Congestion Control (BIC) is optimized for high speed networks with high latency. It has a unique congestion window algorithm. This algorithm tries to find the maximum where to keep the window at for a long period of time, by using a binary search algorithm.

H-TCP or Hamilton TCP is an enhancement to existing TCP congestion control protocols with a good mathematical and imperial basis. The authors assume that their improvement should behave as regular TCP in standard LAN and Wan networks where RTT and link capacity are not extremely high. In the long-distance fast networks, H-TCP is far more aggressive and more flexible while adjusting to available bandwidth and avoiding packet losses. Therefore, two modes are used for congestion control and mode selection depends on time since last experienced congestion. In the regular TCP window control algorithm, the window is increased by adding a constant value alpha and decrease by multiplying by beta. The corresponding values are by default equal to 1 and 0.5. In H-TCP, the window estimation is a bit more complex, because both factors are not constant values but are calculated during transmission. The alpha value depends on the time between experienced congestion events while beta depends on RTT and the achieved bandwidth values. H-TCP has an advantage of fairness and friendliness in that the algorithm, is not greedy and will not consume the whole link capacity. It can share fairly the bandwidth with either another H-TCP like transmissions or non-H-TCP transmissions. H-TCP congestion window improves the dynamic sending ratio and therefore provides better overall throughput value than regular TCP implementation [77, 78].

As stated before, the major problem with TCP and its current implementations is that it relies on packet loss to indicate network congestion. This problem is compounded by the fact that TCP does not have the capacity to distinguish congestion loss from loss invoked by noisy links or any other event. In TCP Westwood (TCPW), there is a small modification to the standard TCP congestion algorithm. When the sender perceives that congestion has appeared, the sender uses the estimated available bandwidth to set the congestion window and the slow start threshold sizes. TCPW avoids huge reductions of these values and ensure both faster recovery and effective congestion avoidance. TCPW does not require any explicit congestion feedback from the network. Measurement of bandwidth in TCP Westwood lean on simple relationship between the amounts of data set by the sender to the time of receiving an acknowledgment. The estimation bandwidth process is a bit like the one used for RTT estimation in standard TCP with additional improvements. When there is an absence of ACKs because no packets were sent, the estimated value goes to zero. Delayed or cumulative ACKs indicating the wrong order of received segments can disrupt the bandwidth estimation process. In such cases, the source must keep track of the number of duplicated ACKs and should be able to detect delayed ACKs and act accordingly [79].

TCP BBR or Bottleneck Bandwidth and RTT is a TCP congestion control algorithm which strives to optimize both throughput and latency by estimating the bottleneck bandwidth and RTT to compute a pacing rate. The goal is to avoid filling up the bottleneck buffer, which might induce bufferbloat. BBR has significantly increased throughput and reduced latency for connections on Google’s internal backbone networks and google.com and its YouTube web servers.

BBR only requires changes on the sender side, not in the network or the receiver side. BBR does not use loss-based congestion control, which the Internet largely uses, like Reno or CUBIC, which relies on packet loss as the signal to slow down data transmission, because loss-based congestion control is outdated for today’s Internet.

Loss-based congestion control causes bufferbloat, causing seconds of needless queuing delays, since it fills the boated buffers in many last-mile links. TCP BBR is in practice, rate based rather than widow based. That is, at any one time, TCP BBR sends at a given calculated rate, instead of sending new data in direct response to each received ACK.

This implementation has several stages. In the STARTUP phase, BBR tries to quickly approximate the bottleneck bandwidth. It does so by increasing the sending rate until the estimated bottleneck bandwidth stops growing. The bottleneck bandwidth is estimated from the amount of data ACKed over a given period, filtered through a windowed max-filter. In the DRAIN phase, the sending or pacing rate is reduced to get rid of the queue that BBR estimates to have created while probing the bottleneck bandwidth during STARTUP. In STEADY STATE, BBR will pace to the estimated bottleneck bandwidth. Periodically it tries to improve its network model by doing probing in two ways: PRBE_BW mode, where BBR probes for a bandwidth increase at the bottleneck by increasing the pacing rate, then decreasing the rate to remove temporary queuing in case the bottleneck bandwidth hasn't grown. PROBE_RTT mode is when RTT is filtered through a windowed min-filter. At times the algorithm will reduce the pacing rate to better approximate the base RTT in case queuing or bufferbloat is in effect [71, 72, 73].

2.2.1 Theoretical Framework

The behaviour of TCP can cause queuing delays in the last mile of the network, or the access network. There have been numerous ideas for improving TCP over the years. Some of the ideas have been adopted by mainstream operations. The recent focus has been improving TCP's behaviour in the advent of high-speed networks. It has been proven that current congestion control algorithms limit the efficiency in network resource utilization. Suggested improvements include changes in the size of the congestion window and to require extra feedback from the network. These are largely divided into explicit congestion control protocol that use explicit feedback from the router and implicit congestion control protocols that rely on implicit measurements of congestion such as loss or delay in the network [71, 72].

There are a variety of TCP congestion control algorithms, each suited for a purpose. These TCP variants mostly adhere to the underlying framework of slow-start, AIMD, retransmit timers and ACK-clocking. None of these changes alter the fundamental underlying dynamics of TCP congestion control. These variants assist in avoiding unnecessary retransmit timeouts, correct unnecessary fast retransmits and retransmit timeouts resulting from disordered or delayed packets. They also reduce unnecessary costs in delay and retransmissions which are associated with the mechanism of congestion notification [63].

In 1988, Van Jacobson [64], introduced TCP congestion control after the Internet suffered a congestion collapse. TCP Tahoe was implemented as a congestion control algorithm, which added several new algorithms and refinements to earlier TCP implementations. New algorithms added to TCP Tahoe were *Slow-Start, Congestion Avoidance and Fast Retransmit*. These refinements catered for modifications to the RTT estimator which was used to set retransmission timeout values. With fast retransmit, after receiving a small number of duplicate acknowledgements for the same TCP segment, the data sender infers that a packet has been lost and retransmits the packet without waiting for a retransmissions time to expire, which leads to higher channel utilization and connection throughput [63, 64, 65].

The Internet has predominantly used loss-based congestion control algorithms since the 1980's, which rely on packet loss as a signal of congestion in the network. Loss-based congestion control is outdated for today's networks as it causes bufferbloat. To overcome this challenge, BBR was developed, an algorithm that is rate based as opposed to loss-based. BBR only requires changes on the sender side and nowhere else in the network. With its limited implementation, BBR has worked well so far in today’s high-speed long-haul links using commodity switches with shallow buffers.

Low Priority TCP (TCP LP) is a distributed algorithm whose goal is to utilize the excess network bandwidth as compared to fair share of bandwidth as targeted by TCP. The key mechanisms unique to TCP LP congestion control are the use of one-way packet delays for early congestion indications and a TCP transparent congestion avoidance policy.

As stated before, the major problem with TCP and its current implementations is that it relies on packet loss to indicate network congestion. This problem is compounded by the fact that TCP does not have the capacity to distinguish congestion loss from loss invoked by noisy links or any other event. Therefore, the behaviour of TCP affects buffer size.

2.2.2 Backbone Networks

A backbone network is a network containing a high capacity connectivity infrastructure that forms the main link or backbone of the network. The backbone has a capacity that far exceeds that of the individual networks connected to it [83, 84].

A backbone network has cabling, switches, bridges, routers and gateways in varying segments. Connections to the backbone or core of the network by access networks is via LANs or ISPs [83].

There are different types of backbone networks namely serial, distributed and collapsed networks. A serial backbone is a backbone architecture that consists of two or more connected devices or nodes. This is rarely used for the enterprise network because it is highly susceptible to faults and system downtime. If a link between two routers faulty, the whole network may get disrupted as there are no alternative data transmission routes. This is ideal in small network setups [83].

A distributed backbone network comprises of a hierarchical formation of devices of which such devices are adaptable to multiple connectivity and are used to connect devices in hierarchy. A distributed backbone network is we suited for enterprise wide connectivity. Expanding and troubleshooting the network is simple and layers of the network are easily added and managed [84].

A collapsed backbone network makes use of a single but high specification router that serves as the actual backbone or central connection that supports the rest of the network. As a backbone, it is characterized by high computational power to adequately handle the traffic from various networks. This setup entails the whole network is dependent on a single router, making the network extremely vulnerable. When the router goes down, the entire network is down. Such networks are used when two different types of sub networks need to be interconnected [83, 84].

A few examples of backbone networks are Zambia Research and Education Network (ZAMREN), ZESCO Fibercom, Research and Education Network for Uganda (RENU) and Abilene [80, 85].

ZAMREN is a specialised Internet Service Provider dedicated to supporting the needs of the research and education communities in Zambia. It provides secure, low cost broadband connectivity to its members, allowing them to share their educational resources via dedicated infrastructure of ZAMREN using optical fibre as a carrier medium. ZAMREN was accorded a free lambda path on the ZESCO optical fibre network grid. The ZAMREN backbone network is shown in Figure 2.2 [81, 80].

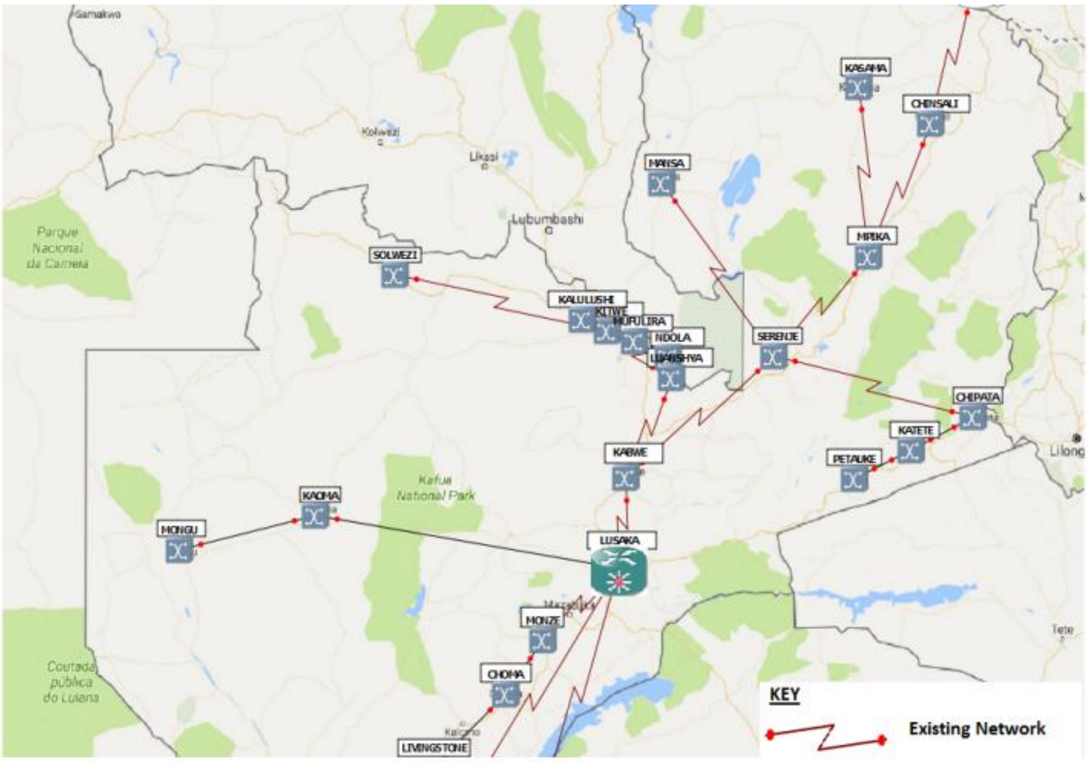

Figure 2.2: ZAMREN network topology [80].

ZESCO Fibrecom, which is presented in Figure 2.3, is the only network with the widest coverage of optic fibre in the country of Zambia. FibreCom services are available in all 10 provincial centers and surrounding districts of Zambia. FibreCom provides Zesco with mission critical services of power grid protection and Supervisory Control and Data Acquision (SCADA) and also provides a backbone network to other service providers. It offers reliable broadband services for high speed data, voice and vide cmmunication that meets the requirements of modern day technology and the business environment [82].

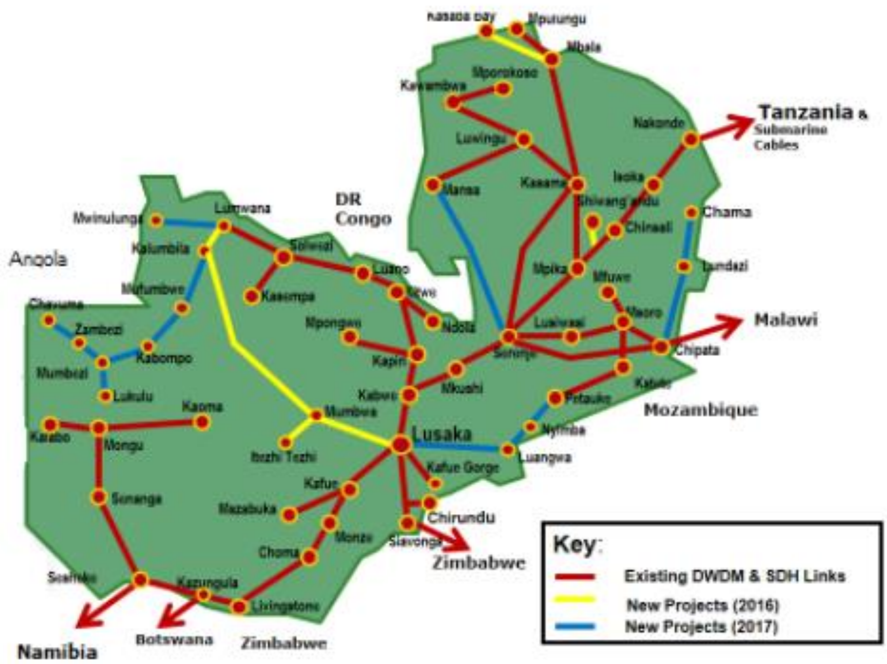

Figure 2.3: FibreCom network topology [82].

Uganda has an educational site which has a network topology as below. RENU encourages ICT enabled collaboration among Ugandan researchers and the higher education community by linking them to their peers and partners, nationally and internationally. Their aim is to enable affordable access for as many research and education intuitions as possible across Uganda through engagement with the institution's leadership, technical capacity building and awareness creation among end users. Figure 2.4 shows the RENU topology and a legend which details the nodes and links in the network.

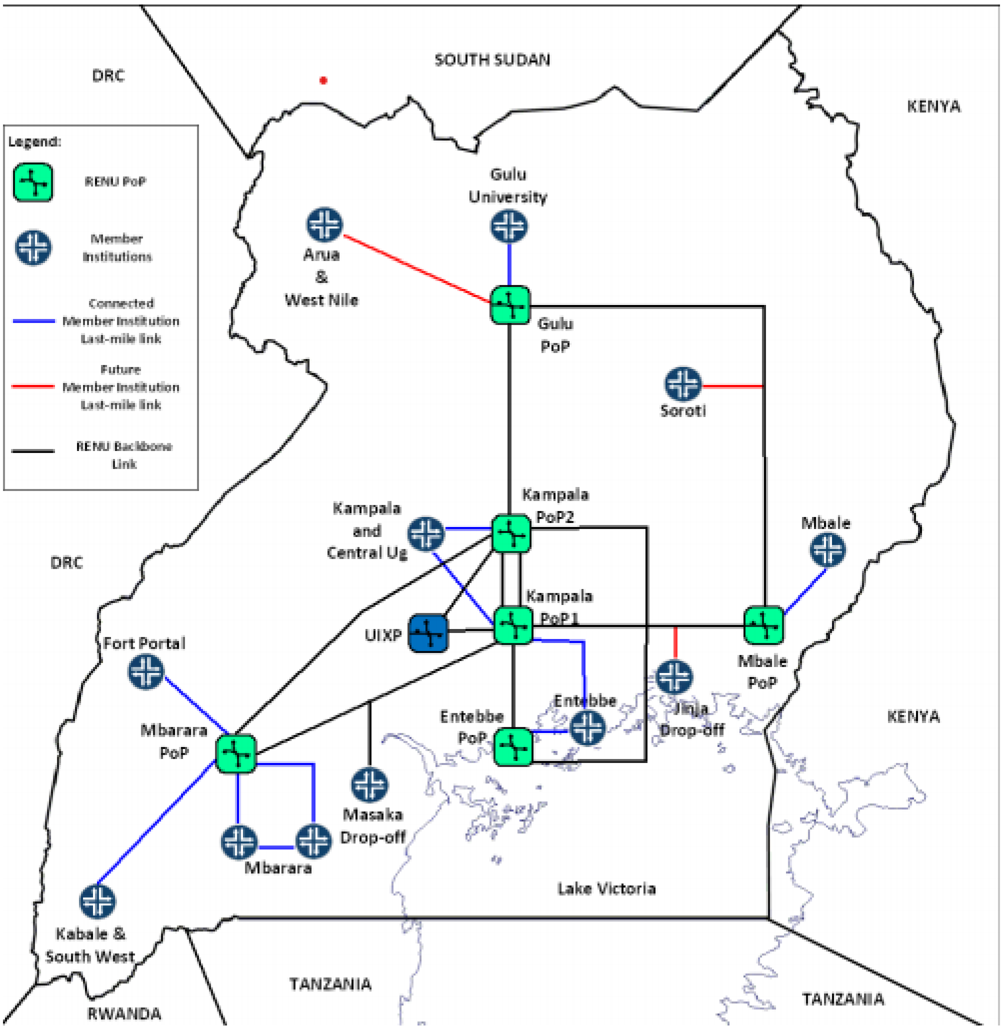

Figure 2.4: RENU network topology [80].

The Abilene Network, in Figure 2.5, was a high-performance backbone network created by the Internet2 community in the late 1990's. In 2007, the Abilene Network was retired, and the upgraded network became known as the Internet2 Network.

Abilene, the most advanced research and education network in the United States of America (USA) achieved 10 Gbps connectivity between all nodes in 2006. This enabled the development of advanced Internet applications and the deployment of leading-edge network services by Internet2 universities and research laboratories across the USA for teaching and research [85, 142, 143].

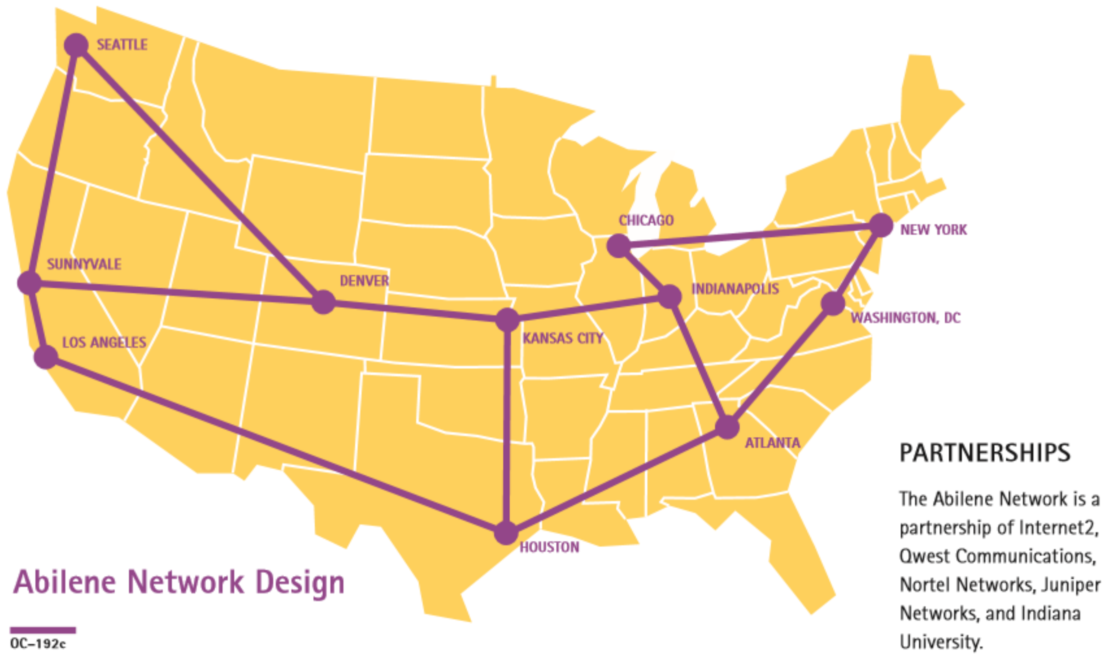

Figure 2.5: Abilene network topology with OC-192 links [85].

The problem of obtaining an accurate picture if the Internet topology is an old one. The shape of the Internet is not known today. An inter-domain graph can indicate the when domains are connected, but it does not provide information on link redundancy. No topology generator can produce router-level topologies of the Internet in a satisfactory way, suitable for simulation. Obtaining a topology of an existing network is limited because operators feel nervous when asked to review a precise view of their network topology [86].

Real world networks are the outcome of a carefully designed process. The network design problem consists of multiple, sometimes contradictory objectives. No single optimal solution exists, rather a front of possible solution. The objectives of network design may be summarized in minimizing the latency, dimensioning the links so that the traffic can be carried without congestion an adding redundancy so that rerouting is possible in case of link or router failure and the network must also be designed at the minim cost.

Simulation and modelling of the Internet is known to be a difficult task. The constant growth of the Internet has made it difficult to develop a representative model for analysis of its topology. Some of the important considerations when generating a synthetic topology is path diversity, which is the existence of alternative paths between a source and a destination. Path diversity is important for network robustness and traffic engineering [86, 87].

In [86], when assigning Interior Gateway Protocol (IGP) weights, a scheme which consists of assigning to each link an IGP weight that is proportional to the link propagation delay or the link mileage is used. The analysis shows a clear correlation between the scheme deployed and the Abilene backbone network. Abilene has the least number of edges and vertices compared to the network topologies under consideration in section 2.2.2. The number of vertices and edges in Abilene network is suitable for running simulation using Riverbed Academic Modeller based on the maximum allowed number of core nodes [141].

2.3 Network Simulators

A network simulator is a software program that imitates the working of a computer network using devices and performance analysed. Network simulators allow researchers to test the scenarios that are difficult or expensive to simulate in the real world by designing and testing various network topologies using nodes, hosts, bridges, firewalls and routers [87, 88, 89, 90].

Computer network simulators can be viewed based on availability, easiness, scalability and data manipulation, including graphical display. Discrete event simulators are in which system behaviour can be simulated by modelling the events in a system as per order of the scenarios the user has setup. Such software is NS2, NS3, NetSim, OMNet++ and OPNET to mention just a few.

Network Simulator version 2 (NS2) is an open source discrete event simulator designed specifically for network research. NS2 provides support for both wired and wireless simulation of functions and protocols such as TCP, UDP, HTTP and RTP. NS2 is flexible and modular in behaviour and is one of the most popular network simulators. NS2 can be used to perform simulation to explore different issues like protocol interaction, congestion control, and effect of network dynamics and scalability of a network. To interpret simulation results graphically and interactively, tools such as network animator (NAM) and Gnuplot are used. The advantage of NS2 is that it has many available models, realistic mobility models and a powerful and flexible scripting and simulation setup. Complex scenarios can also be tested, with easy traffic and movement pattern having an efficient energy model. The disadvantages of NS2 are the recompilation each time there is a change in the user code and the complexity of modelling a real system with complex infrastructure [86, 87].

NS3 is another discrete event network simulator targeted primarily for research and educational use. It defines a model of working procedure of packet data networks and provides an engine for simulation. NS3 can be used to model non-Internet based systems. In NS3, simulations can be written in C++ or python and animators are used to visually display the results.

NS3 has high modularity than NS2 and as supports TCP, UDP and ICMP, multicast routing and CSMA protocols. The disadvantages are that NS3 lacks a lot of specialized maintainers to avail the merits of NS3 as the commercial and academic Riverbed network simulator. NS3 does not have an active base [86, 90].

NetSim is a stochastic discrete event network simulation tool used for network laboratory experiments and research. It is a leading network simulation software for protocol modelling and simulation. It allows users to analyse computer networks with unmatched depth power and flexibility. NetSim provides network performance metrics at various abstraction levels of the network, such as network view, sub-network, node and detailed packet trace.

NetSim has a GUI feature which has drag and drop functionality for devices and links. Data packet and control packet few can be visualised through NetSims built in packet animator. The downside to using NetSim is that it is a single process discrete event simulator. A single event queue is used for the simulation which at any given time contains one entry for each statin n the network NetSim does not have any free version of software.

OMNeT++ or objective Modular Network Tested in C++, is an open source, extensible, modular, component based discrete event simulator like NS2 and NS3 used to simulate both wired and wireless networks. Tracing and debugging using OMNet is much easier to use than other simulators. With its powerful GUI environment, accurately modelling most hardware and physical phenomenon is easy. It supports parallel distributed simulation. OMNeT++ has domain-specific functionality such as support for sensor networks, wireless ad-hoc networks and Internet protocols.

OMNet++ does not offer a great variety of protocols and very few protocols have been implemented, leaving users with significant background work. There is also poor analysis and management of typical performance and the mobility extension is relatively incomplete.

OPNET (Optimized Network Engineering Tools) or now Riverbed Modeller can be flexibly used to study communication networks, devices, protocols and applications. It is a discrete event system simulation. It offers relatively much powerful graphical support for the users. The graphical editor interface can be used to build network topology and entities from the application layer to the physical layer. Object-Oriented Programming techniques is used to create mapping from the graphical design to the implementation of real systems. All topologies configuration and simulation results can be presented intuitively and visually. The parameters can be adjusted, and experiments repeated easily without recompiling any code.

The advantage of OPNET is the inherent three main functions of modelling, simulating and analysis. It has a high-level user interface with an intuitive graphical user interface. It arranges its hierarchical structure model into three specific domains of network domain, node domain and process domain.

The disadvantage is that it does not allow a high number of nodes, not more than 20, within a single connected device. Simulation is ineffective if nothing happens for long periods of time and accuracy of results is limited by the sampling resource distribution [86, 88, 141].

In [92] a study of different network simulators showed that OPNET provided the most accurate results. Due to its commercial nature, a complete set of OPNET is well suited.

2.4 Why routers need buffers

There are three main reasons why routers have buffers. First its congestion. Congestion occurs when packets for a switch output arrive faster than the speed of the outgoing line Packets might arrive continuously at two different inputs all destined to the same output queue. If a switch output is continuously overloaded, its buffer will eventually overflow, no matter how large it is, it will not be able to transmit the packets as fast as they arrive. Short term congestion is common due to the statistical arrival time of packets. Long-term congestion is usually controlled by an external mechanism, such as the end-to-end congestion avoidance mechanisms of TCP, the

XON/XOFF mechanisms of Ethernet, or by the end-host application. In practice, we must decide how big to make the congestion buffers. The decision is based on the congestion control mechanisms – if it responds quickly to reduce congestion, then the buffers can be small; else the buffers will have to be large. Congestion buffers are the largest buffers in a router. A typical Internet router today holds millions of packet buffers for congestion [93].

The second reason is internal contention. In cases when the external links are not congested, most packet switches can experience internal contention because of imperfections in their data paths and arbitration mechanisms. The amount of contention, and therefore the number of buffers needed is determined by the switch architecture. For example, an output-queued switch has no internal contention and needs no contention buffers, while an input-queued switch can have lots of internal contention which limits the throughput of an input-queued switch to 58% of its maximum [93, 94]. It is possible to build input-queued switches with 100% throughput. Such switches need large internal buffers, which theoretically can be of infinite depth, to hold packets during times of contention. Some architectures can emulate output queueing through careful arbitration and a combination of input and output queues (CIOQ). Figure 2.6 shows an example of buffering in such an architecture. These switches still need contention queues at their inputs to hold packets while the arbitration algorithm decides when to deliver each to its output queue. Most switches today use CIOQ or multiple stages of CIOQ. CIOQ switches typically need very small contention buffers [95, 96].

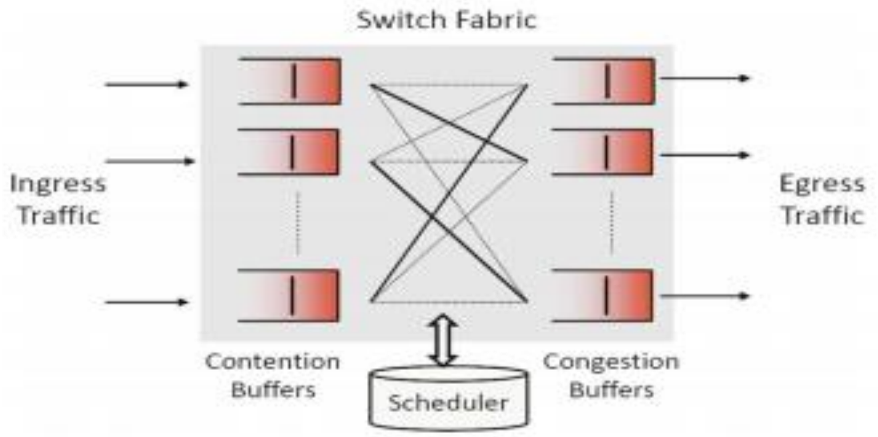

Figure 2.6: Buffering in a CIOQ router [93].

The third reason is staging. Packet switches also have staging buffers for pipelining and synchronization. Most designs have hundreds of pipeline stages, each with a small fixed-delay buffer to hold a fixed amount of data. Most designs also have multiple clock domains, with packets crossings several domains between input and output; each transition requires a small fixed-size FIFO [95, 96].

2.5 Bandwidth Delay Product

The size of router buffers is important for effective operation of networks with statistically multiplexed traffic. The use of TCP or most data communication in the Internet as led to buffer sizing recommendations based on the product of round-trip time and link capacity. Such large buffers ensure that buffer underflows can be avoided when TCP reacts to pacts loss, indicative of congestion, with multiplicative decrease in transmission rate [97].

2.6 Related Works

2.6.1 Introduction

High-bandwidth data communication is one of the key foundations of the future internet. Today’s internet use has increased the need for raw bandwidth in the network. Optical fibre has long been used for long-haul, point to point data transmission, and the need to convert from optical domain to electronic domain for switching has limited the throughput that can be achieved in first generation optical networks. Optical packet switching, and optical burst switching have been proposed as an alternative to optical-electronic-optical conversion, where packets are switched independently in the optical domain.

While such technologies are promising approaches to optical networks that match the needs of current internet protocols, and dynamics, there are still important technological challenges that need to be addressed. Apart from the difficulties of building a switch that operates in the optical domain, there are also challenges imposed by the basic operation of the Internet. The fact that packet switching is based on the statistical multiplexing and traffic is forwarded opportunistically without prior resource reservations entails that traffic may compete for link bandwidth on the output port of a switch or router. To avoid packet losses, packet buffers are used to queue packets, and this absorb short periods of overload [97].

In electronic networks, and first-generation networks, buffers can be implemented easily with SRAM or DRAM technologies and can be sized to hold many packets or small number of packets. In all optical networks or second-generation networks, buffers are more difficult to implement since there exists no practical solution to store light other than sending it through delay lines. These optical buffers can only hold a very small number of packets [98]. The efficiency of small buffers operating with network, transport and application layer protocols that have been designed for large buffer networks has been a subject of research [98]. Buffer sizing has also been explored in the context of reducing the cost of electronic routers by reducing the amount of buffer memory per port, modelling the buffer input and output rates, and modelling buffer size in mixed traffic conditions [18, 97, 99, 100]. One key concern with small-buffer networks is their interaction with the transmission control protocol

(TCP). The use of congestion control in TCP leads to significant performance drops in throughput when packet loss occurs. Studies have explored how traffic characteristics impact packet losses in small buffer networks and how these losses affect TCP throughput [97, 101, 102, 103, 104, 105, 106].

Traffic aggregation of multiple connections leads to traffic characteristics that ensure efficient operation even with smear buffer sizes. [104] shows that buffer size proportional to square root of the number of flows may be enough. While small buffer networks can operate efficiently with TCP for some scenarios, it has also shown that this is not the case for many practical scenarios. The presence of TCP connections with short lifetimes are problematic. Haesegawana et al. show that a high ratio of short-lived flows has a negative effect on link utilization for small buffers. Sivaraman et al. show the impact of small buffers on real time TCP traffic and identify short timescale bustiness as a major contributor of performance degradation [104, 105, 107, 108, 109].

2.6.2 Data Rate and Router Buffer Size

The explosive growth of multiple reliable connections in networks that causes severe traffic congestion problems which makes it common to see internet gateways drop a good percentage of the incoming packets because of local buffer overflows limited bandwidth. Bandwidth is the amount of information that can pass through a communication channel in a given amount of time, often expressed in bits per second or bytes per second [110].

In building a model for simulating the traffic flows and TCP behaviour using Optimum Network Engineering Tools (OPNET), Yakubu et al [110], highlight the challenges of performance evaluation for systems. Prototyping, which is building the system or a scaled down version of it, to see how it works. Analytical modelling, which involves building a mathematical model of a system and use it to analyse the system and Simulation, which is building a software model of the system. Prototyping is not feasible many times or time consuming, especially for large systems. It provides limited control and observation as well. Analytically modelling cannot fully capture highly complex systems. There are many obstacles in the process of performing computer studies, such as building a real-world test-bed for performance analysis. The cost of such is most times manageable or feasible. The cost for the problem increases as the need for different configuration parameters arises.

Thus, simulation has emerged as an attractive alternative that is heavily used in performance evaluation of computer systems because an emulated network architecture can be controlled, and predefined test configurations may be run automatically, making the execution of tasks more convenient [110].

In [110], using OPNET, an emulation environment was set up using ten client hosts on the network and two different hots were set to generate voice and video streams, with each of the ten hosts generating TCP traffic. In performing a correlation analysis of data rate and router buffer size on TCP performance, Yakubu et al., selected two application traffic models, one in an environment with reliable connections without UDP traffic and another with UDP traffic.

The simulation involved varying the wireless data rate of 128 Kbps, 256 Kbps and 512 Kbps, running a simulation of these rates over a fixed buffer size of 16 MB, 32 MB and 64 MB. This was to see the effect of data rate on the performance of TCP over the same router buffer size. By varying data rates and router buffer sizes to very small buffer size values, they showed that different data rates significantly yield different packet drops and retransmission behaviour.

They note that based on their simulation results, under different scenarios, router buffer size or data rate only cannot optimize the performance of TCP. Therefore, in the performance of TCP, data rate and router buffer size are strongly related, with data rate having a significant effect on the performance of TCP [110].

2.6.3 Buffer Size based on Link Utilization

The widely used rule of thumb, attributed to [112] was obtained experimentally by using at most 8 TCP flows on a 40 Mbps core link in 1994. No recommendation was made for sizing buffers when there is a significant number of TCP flows that have different RTTs [111]. The rule of thumb was meant to guarantee a 100% link utilization, such that even when a buffer overflows, and TCP reacts by reducing its transmission rate, there are enough pockets stored in the buffer to keep the output link busy, thereby ensuring that the link capacity is not wasted in times when TCP has t increase its transition rate after reducing its flow.

In a first-generation optical network, with a link capacity of 40 Gbps or an OC-768 link, a buffer size based on the rule of thumb with an RTT of 250 ms would equal 1.25GB. This poses a lot of challenges even though it keeps link utilization at a near 100% [111].

Due to complications with router design, power consumption and challenges with an optical router, alternative buffer size solutions have been suggested, researchers from Stanford University showed that when a large number N of long-lived flows share a bottleneck link in the core of the Internet, the absence of synchrony among the flows permits a central limit approximation of the buffer occupancy [111, 113, 114].

A combined effect of multiplexing many asynchronous flows lead to a buffer size that achieves a near 100% link utilization. The result assumes that there are sufficiently large number of TCP flows such that they are asynchronous and independent of each other. This result holding means that a core router carrying 10,000 long lived TCP flows will only need 12.5 MB of buffer instead of 1.25 GB as governed by the rule of thumb. This is the small buffer model [111].

Other small buffer models have also reported that a near 100% link utilization can be achieved with fewer buffer than suggested by the rule of thumb. Using a fluid model, Avranchenkov et al. [115], formulates the buffer sizing problem as a multi-criteria optimization problem and suggests that the buffer size needed to get full link utilization decreases as the number of long-lived TCP flows increases. The study suggests a buffer size that is even lower than what was suggested by the small buffer model [111]. Andrew et al., in [116] show that in the absence of TCP timeouts and in the presence of many long-lived TCP flows, buffer size much smear than the rule of thumb is sufficient to get high throughput. Using a doubly-stochastic Markovian model, another study shows that as the number of flows increases, good performance is realized in ensuring full link utilization, with almost zero packets lost and negligible queueing delay [117].

Further reduction is buffer size has been recommended in several studies. Using control theory, differential equations and extensive simulation, [118, 119, 120, 121, 122, 123, 124], have argued in favour of further reducing the buffer size and recommend that as a few as 20-50 packets of buffering sacrifice at core routers for TCP traffic to realize acceptable link capacities. This is the tiny buffer model in literature [111].

The tiny buffer model comes with a trade-off for link capacity as reducing buffers to only a few dozen kilo bytes can lead to a 10-20% drop in link utilization. The tiny buffer model relies on the fact that TCP flows are not synchronized, and network traffic is not bursty. Such traffic scenarios can happen in two ways, first, as core links operate at a much higher speed than access links, packets from the source node are automatically spread out and burst are broken. Second, if the TCP stack running at end-hosts is altered such that it can space out packet transmissions [111].

The slight drop in link utilization resulting from the tiny buffer model seems worthwhile since core links are usually overprovisioned, and it pays to sacrifice a bit of link capacity if this permits a move to either an all-optical packet switch or more efficient electronic router design. The preceding studies suggest that acceptable link utilization can be obtained if smooth TCP connections flow through a single router with tiny buffers [111].

In [125], the authors explore if an arbitrary network topology can sustain this lowered utilization and their results indicate that if a network has a tree structure, then no modification is needed in the routers. For a general topology, the use of a simple active queue management mechanism called bounded jitter policy is suggested, which will render the arrival traffic at each router to behave as if it is directly being fed by the ingress ports of the network, and thus tiny buffers still suffice to maintain acceptable link utilization.

The tiny buffer model is also suggested by Gorinsky et al., in [126, 127] by proposing a very small buffer by considering the needs of other diverse Internet applications in a heterogeneous network. They suggest a buffer size that is twice (2L) the number of input links, where L is the number of input links. The constant two helps to keep buffer small enough so that queuing delays do not become significant. The objective is that applications in the access layer have means of dealing with link underutilization, but they have no way of mitigating queuing delays induced by other applications at buffers. In this study, it was observed that some variants of TCP like TCP-New Reno can achieve at least 75 % link utilization. Higher utilizations can be obtained by using smoother versions of TCP such as TCP-Vegas [111].

.

With the preceding studies focused on link utilization, there has been concerns regarding the use of link utilization as the only performance metric in determining buffer size [111]. Dhamdhere et al., suggest that in addition to link utilization, packet loss rate is also an important metric to consider, stating that the study of buffer size should aim to keep the loss less bounded to a small value. A minimum buffer size required to keep a link fully utilized is derived by N long lived TCP flows with varying round-trip times, while at the same time attempting to bound the loss rate and queuing delay. A buffer size required by N heterogonous TCP flows at a router employing the drop tail queue management depends on the harmonic mean of their round-trip times. This is called Buffer Size for Congested Links (BSCL). This study suggests that Stanford's small buffer model is more appropriate when provisioning buffers at core routers since it rarely becomes a bottleneck for most of the traffic flowing through it. The small buffer model can result in high losses in the edge and access routers where links can become congested with large TCP flows that are locally bottlenecked and, in such cases, BSCL should be preferred [128, 111].

The probability of packet loss in buffer sizing context was considered in [129, 130] where it was shown that loss rate increases with the square of the number of competing TCP flows. Therefore, sizing buffers, by only basing them on the rule of thumb alone, can result in frequent TCP timeouts and significant variations to the per-flow throughput of the various competing flows. A suggested solution to this problem, is an adaptive buffer sizing mechanism called Flow-Proportional Queueing (FPQ), which typically adjusts the amount if buffers according to the number of TCP flows. Other studies model the buffer sizing problem as the Lur's problem and presents an active drop tail algorithm to determine the buffer size that minimizes queueing delay while maintain a certain average link utilization. Other studies model the buffer size problem as an adaptive buffer sizing algorithm (ABS), where the router adapts its buffer size to suit the dynamics of incoming traffic. Using the monotonic relationship between buffer size, link utilization, loss rate and queuing delay, ABS aims to maintain system performance above a certain given target objective. Performance results of ABS show its applicability and its scalability to increasing link capacities [131, 132].

Using simulations, the authors of [133] show that employing the small buffer model can result in 5-15% loses for TCP, which may be unacceptable. High loss rate can prove perilous to certain applications like audio and video, which require high reliability and interactivity. They show that the rule of thumb can also lead to high packet loss rates in access networks, and that bigger buffers may be needed. Experimental data from [134] shows that loss rate can be as high as 25 % during times of high throughput, even when buffer size is high enough to maintain 95% link utilization. Tiny buffers results indicate that loss rates can remain high, about 20% while link utilization is consistently low, of below 60%, thus deterring the use of tiny buffers on a heavily loaded link.

In [135], the study concludes that at the core of the Internet, where there are many TCP flows at any given time, buffers can be safely reduced by a factor of ten without affecting the network performance. Care must be exercised when directly employing the small buffer model since it may not hold in all parts of the network, particularly the access side. The use of tiny buffers is justifiable in a future all-optical network, since bandwidth will be abundant, but technological challenges limit the buffer size to a few dozen packets. Thus, the 10-20% reduction in link utilization may be acceptable.

Another alterative view is from [136], where they argue that the small buffer size model of maintain a near 100% link utilization is not applicable in small buffer networks since it does not account for the traffic variability on input links. A new method suggests that uses a single fixed-point equation to statistically characterize a large network with small buffers. This is derived by combining a series of models that capture the traffic arrival distribution on bottleneck links, instantaneous arrival rates, queue occupancy and packet loss rates. These interrelated models can be used to size buffers given certain target objectives such as maximum packet loss probability or maximum expected packet delay.

2.6.4 Buffer Sizing Based on Per-Flow Metric

Prasad et al., examined the problem of buffer sizing from three new viewpoints. First, instead of assuming that most of the traffic consists of persistent TCP flows, or very long transfers that are mostly in congestion-avoidance, the study works with a more realistic model of non-persistent flows that follow a heavy-tailed size distribution. The implications of this modelling deviation are two-fold, first non-persistent flows do not necessarily saturate their path, and secondly, such flows can spend much of their lifetime in slow-start, and third, the number of active flows is highly variable with time. The flow results show that flows that spend most of their lifetime in slow-start require significantly less buffering than flows that live mostly in congestion-avoidance [137].

Secondly, instead of only considering link-level performance metrics, such as utilization, average delay and loss probability, the focus is on the performance of individual TCP flows on the relation between the average throughput of a TCP flow and the buffer size in its bottleneck link. TCP accounts for more than 90% of Internet traffic, and so a TCP-centric approach to a router buffer sizing would be appropriate in practice for both users and network operators. On the other hand, aggregate metrics, such as link utilization or loss probability, can hide what happens at the transport or application layers. For instance, the link may have enough buffers so that it does not suffer from underutilization, but the per-flow TCP throughput can be very low [137].

The third focus is on the structural characteristics of a link or traffic multiplexer that has been largely ignored in the past with a few exceptions. This characteristic is the ratio of the output/input capacities. A link of a certain output capacity that receives traffic from several links, each of a certain input capacity, with a product of the number of links and input capacity being greater than the output capacity. In the simplest case where sources are directly connected to the input links, and the input capacity is the source peak rate. Generally, however, a flow can be bottlenecked at any link between the source and output port under consideration. In such a case, the input capacity then, is the capacity of that bottleneck link. Considering an edge router interface with output capacity 10Mbps. Suppose that the input interfaces of that router are 100Mbps. If the traffic sources are directly connected to that router, the ratio of output capacity to input capacity is equal to 0.1. On the other hand, if the sources are connected to the router through 1Mbps DSL links, then the ratio is 10. The ratio of output to input capacities largely determines the relationship between loss probability and buffer size, and consequently, the relationship between TCP throughput and buffer size [137].

The authors of [137] determine two approximations for the relationship between buffer size and loss rate, which anchor on the source of the traffic being heavily tailed. If the output to input ratio is less than 1, the loss rate can be approximated by a power law of the buffer size. The buffer requirement can be significant in that case, especially when the aim is to maximize the throughput of TCP flows that are in congestion-avoidance. This is so because the buffer size of requirement for TCP flows in slow-start is significantly lower. On the other hand, when the output to input ratio is greater than 1, the loss probability drops almost exponentially with the buffer size, and the optimal buffer size is extremely small to be just a few packets in practice and zero theoretically. This ratio is often lower than 1 in the access links of server farms, where hosts with 1 to 10 Gbps interfaces feed into lower capacity edge links. Conversely, the ratio is typically higher than one at access networks, as traffic enters the high-speed core from limited capacity residential links. These results were experimentally arrived at using a testbed bottleneck of a Gig-Ethernet output interface that connects the router to the distribution switch [111, 137].

A Riverstone RS-15008 router was used, with a switching fabric having a much higher capacity than the bottleneck link. There was no significant queueing delay at the input interfaces or at the fabric itself. The router had a tuneable buffer size at the output line card. The experiment used 20 buffer sizes, non-uniformly selected in the range of 30KB to 38MB, with Ethernet MTU packets of 1500 bytes, the minimum buffer size is about 20 packets while the maximum buffer size is approximately 26,564 packets. Using a drop-tail queuing configuration, they confirmed that the maximum queuing delay for a buffer size is equal to the buffer size divided by the capacity of the output link.

This study provides further evidence that the buffer provisioning formula that sets out the buffer size equal to the links BDP is probably far less from optimal. The BDP only applies in the very special case that the link is saturated by a single persistent TCP connection, and so it can be quite misleading in most practical cases. Buffer size can be significantly less than the BDP when the link carries many flows. The study concludes that it is difficult to arrive at a simple and handy formula that one can use for sizing the buffers of any router interface. Finding a universal buffer size might not be practical as an optimal buffer size at an Internet link depends on several parameters that are related to both the offered load and network design. The most practical recommendation is that network operators be the ones to determine the buffer size by first determine the capacity ratio of their links and to decide whether they will optimize the throughput of connections in slow-start or in congestion avoidance [111, 137].

2.7 Buffer sizing and Simulation

There are generally three approaches to a research problem. These are quantitative, qualitative and mixed methods approach. The use of each of these methods or a combination of quantitative and qualitative methods is determined by the type of problem at hand [31].

A qualitative approach to research is favoured when a research problem will need to be analysed for trends or when relating variables using statistical analysis tools and then interpret the results by comparing them with past research. A conclusion would then have to be made using a standard and fixed structure that takes an objective and unbiased approach.

Qualitative research on the other hand is suitable for a research problem that has no known variables and therefore needs exploration. The problem will need to be explored and develop a detailed understanding of the central phenomenon, which can be a key concept, process or idea that forms the research. Data is collected from a small group of people as participant's views and is analysed for description and themes using text analysis and interpreting the larger meaning of the findings. A conclusion is reached based on flexible, emerging structures and evaluation criteria, including the researches subjective flexibility and bias [28, 29].

However, there are times when both types of approaches are needed when addressing a research problem. The approach taken in such studies are a mixed method approach. The research problem is addressed by following a process of collecting, analysing and mixing both quantitative and qualitative data in a single study or in a multiphase series of studies. A conclusion is reached with the researcher choosing what kind of data of the two types of approaches, to prioritise and whether data is collected concurrently or sequentially. This guides on how the data will be integrated and used to form a conclusion [27, 30, 31].

In addressing the objectives and research questions of this study, there is need to have both subjective ad objective views when deriving the conclusion. This study will need to answer questions as stated in section 1.7. To do this, this study needs to identify variables for data collection and look at any text or charts to properly analyse the data.

The appropriate research approach for this study is a mixed method approach. This approach will consist of collection of data by simulating a system model. Simulation suing Riverbed Modeller will help collect both quantitative and qualitative data concurrently and simultaneously. Quantitate data will be obtained by varying the variables that will be input into the system model, which will yield both quantitative in terms of the numbers or values that constitute the simulation, and qualitative results, in form of graphs and charts which will help in analysing the quality of the methods employed. During the simulation, the qualitative behaviour of the system will also be analysed by observing the packet drops during the simulations.

2.8 Summary

In this chapter, a comprehensive background, descriptions and theory of buffer sizing has been given. Network topologies and the related works have been examined and the most common approach to the problem of buffer sizing. Simulation tools and network simulation procedures have been stated. The chapter also highlights from literature, the challenge of having a universal optimal buffer size that fits at all points of the network. It has been noted that buffer size requirements vary at different points in a network and this must be taken into consideration when designing a network.

CHAPTER THREE:

METHODOLOGY

3.1 Introduction

This chapter is about the research methodology that was followed when conducting this study. It starts with a review of some common research approaches then states the adopted approach for this study. The tools used in this study are highlighted and the method of data analysis used. The chapter ends with a summary.

3.2 Research Approach and Methodology

There are different philosophies for the attainment of knowledge. As one philosophy, positivism adheres to the view that only factual knowledge gained through observation and measurements is trustworthy. Positivism depends on quantifiable observations that lead to statistical analyses. This is in accordance with the empiricist view that knowledge stems from human experience. It has an atomistic, ontological view of the world as comprising discreet, observable elements and events that interact in an observable, determined and regular manner [148, 149].

In positivism studies, the researcher is independent from the study and there are no provisions for human interest within the study. The role of a researcher is limited to data collection and interpretation in an objective way, where findings are observable and quantifiable. A positivist approach follows a deductive approach in research as it relates to a viewpoint that a researcher needs to concentrate on facts [149, 150].

Another philosophical view for finding knowledge is known as interpretivist. This involves researchers interpreting elements of the study, thus interpretivist integrates human interest into the study. The belief is that access to reality can only be through social constructions such as language, consciousness, shared meaning and instruments. Interpretivist is based on a critique of positivism and rejects the objective view that meaning resides within the world independent of consciousness [151, 152].

A discussion of a research approach is a vital part of any scientific study regardless of the research area. Research approaches can be divided into three types. Deductive approach, inductive approach and abductive approach. Deductive approach tests the validity of assumptions or theories, while inductive approach contributes to the emergence of new theories and generalizations. With an abductive approach, research starts with surprising or not well known or understood facts and the process seeks to find an explanation for such [144, 145].

The logic for deductive approach is such that when the premises are true, the conclusion must also be true, while in induction, known premises are used to generate untested conclusions. An inductive approach starts with the observations and theories are proposed towards the end of the research process as results of observations. In an abductive approach, premises are used to generate testable conclusions. A deductive approach will move from general to specific and data collection is used to evaluate propositions or hypotheses related to an existing theory where theory is either falsified or verified. An inductive approach will move from the specific to the general and use data collection as a means of exploring a phenomenon, identify themes ad patterns and create a conceptual framework which will generate a new theory or build on an already existing theory. Abductive approach will generalize based on the interactions between the specific and the general and use data collection to explore a phenomenon, identify themes and patterns and locate these in a conceptual framework and test this through subsequent data collection. In this approach, theories are either generated or modified by incorporating existing theories where appropriate and build new theory or modify existing theory [144, 145, 146].

Studies on buffer size and the relationship between buffer size and TCP have been based on theoretical analysis and validations using simulations and in some cases simulations on internal networks [153]. These studies have used both quantitative and qualitative approaches to their studies which fall under an inductive and deductive approaches.

This study uses a mix of the inductive and deductive approaches. The inductive approach is used because this research is searching for a pattern from observation and the development of explanations. The deductive approach will be used in the quantitative analysis of simulation results. With this approach, meaning will be generated from the data sets that are collected to identify any patterns or relationships from the variables used in the research. Variables are the inputs to finding the objectives in this research [110, 137], where packets are in slow start and can be routed in a network with a small buffer.

The application of inductive approach is associated with qualitative methods of data collection and data analysis whereas deductive approach is perceived to be related to quantitative methods. An inductive approach will focus on grounded theory and an exploratory data analysis while deductive approach will focus on quantitative comparative analysis and structural equation modelling for qualitative and quantitative approaches respectively [147].

3.3 Research Design

Research design is a comprehensive plan for data collection in an empirical research project. This is a blueprint for the research aimed at answering specific research questions or testing specific hypothesises. Based on the chosen research approach and methodology, research methods were designed to achieve the objectives of the research. A mixed method approach will require a combination of methods from both a qualitative and quantitative viewpoint [155, 156, 157, 158, 159]. This study uses a mixed method approach to determine an alternative buffer size assignment in an access network.

Each objective and the associated research questions have a specific method for collection of data and its analysis. For qualitative analysis, secondary data analysis on buffer sizing was used as one of the methods for data collection and analysis, while quantitatively, simulations were used [156, 160]. The two methods therefore, that this study undertakes in the quest to understand and answer research questions as set out in section 1.7 are secondary data analysis for research question one and two, and the method used for research questions three to five is simulations.

Figure 3.1 shows the conceptual framework which illustrates the simulation process. At the start of the simulation, environmental variables are adjusted based on the tiny buffer models, where buffer size is based on flows being in slow start, which is a state where there is no congestion in the network. Variables are adjusted to reach an optimal performance in the network. An optimal performance is when packet loss is minimized, and throughput is maximized.   The cycle is repeated until optimal values are obtained from which a suggested model for assignment of buffer size in an access network is obtained.

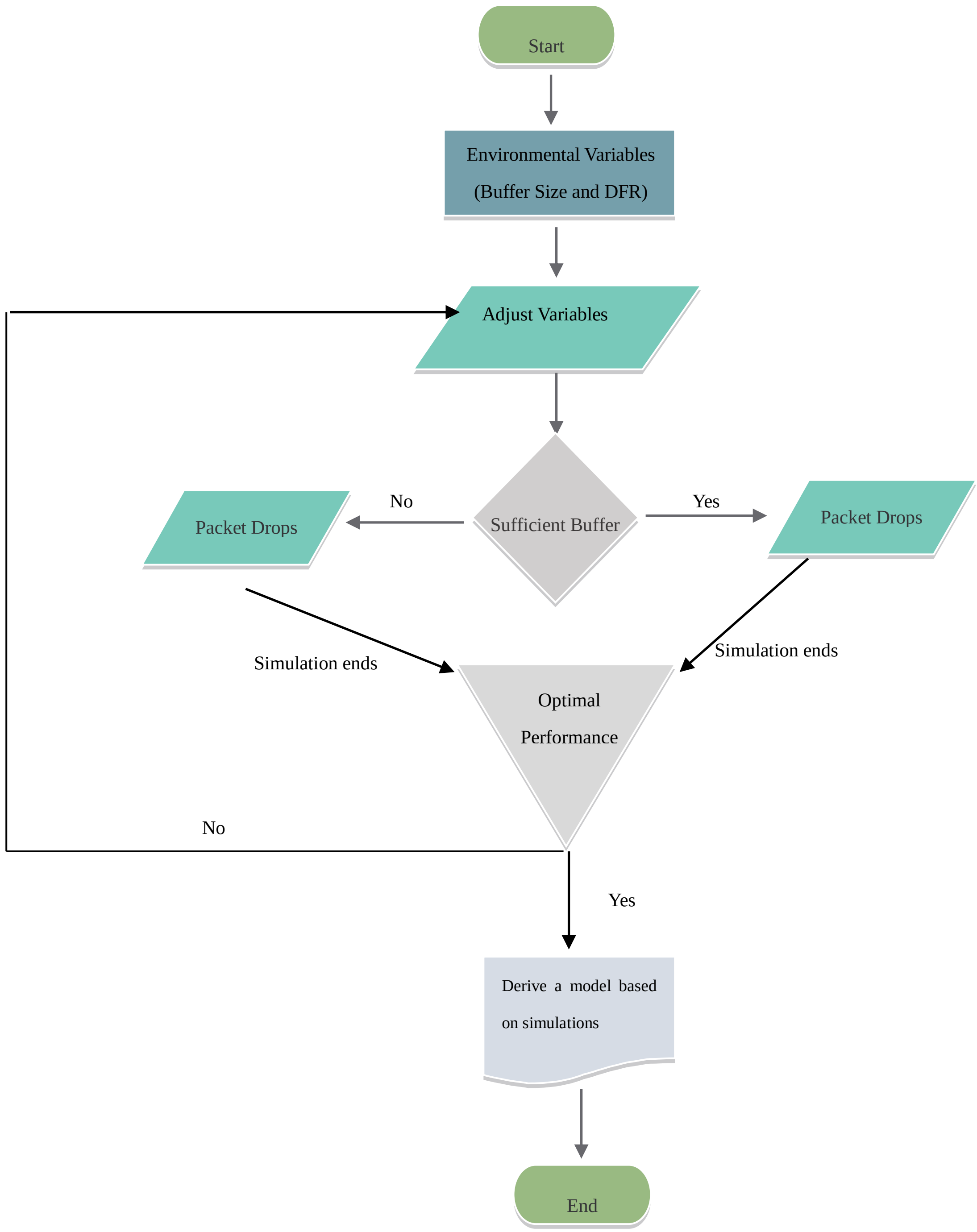

Figure 3.1: Flow chart for the conceptual framework illustrating the simulation process.

Table 3.1 shows the research method that was used to address the research objectives and the questions that were asked to reach a conclusion.

Table 3.1: Research objectives, questions and methods used.

OBJECTIVE	RESEARCH QUESTION	METHOD
1. To assess the buffer sizing methods, their challenges and determine the suitability of current buffer sizing techniques for first generation optical networks.	a) What is the current buffer sizing model for assignment of buffer size in the Internet? b) What are the challenges of the current buffer sizing techniques used in the Internet today?	a) Secondary data analysis.
2. To determine whether a homogeneous buffer size is suitable for a first-generation optical network in heterogeneous traffic.	a) How suitable is a homogeneous buffer size for a first-generation optical network carrying heterogeneous traffic?	a) System simulation b) Observation
3. To devise and design a model that can be used to assign buffer size independent of link capacity where link utilization is not a metric for assignment.	a) How can a minimal buffer size for first generation optical network be modelled? b) How can we devise a method for assignment of buffer size in access network in a first-generation optical network?	a) System simulation b) Observation

3.4 Methods and Tools Used

Simulation software as a modelling tool was used is this research. Simulation is used in many studies of buffer sizing, most of which use OPNET/Riverbed Academic Modeller or NS2 among other tools used [110, 111, 113, 114]. As stated earlier, in 3.4, the methods used are secondary data analysis and model simulation.

This study uses Riverbed Academic Modeller to conduct the simulations. The simplicity of use and design of Riverbed and its adaptation for academic modelling made it an easy choice for this study. Riverbed has inbuilt analysis and graphing tools which is also an advantage for this software.

Other tools used were Microsoft Excel, which was used for conversion of the simulation results into a comma separated file (csv) which was then input into Student Magic Plot, a graphing software, for a graphical representation of results of the simulation.

3.5 Simulation Setup

The simulation was set up using Riverbed Modeller, based on the Abilene network topology. The specifications of the computer on which Riverbed Modeller was installed are;

a) Windows 10 Pro version 1703
b) Intel CPU Celeron at 2.16 GHz
c) RAM of 4 GB
d) 64-bit operating system.

The Riverbed Academic Modeller Edition, version used is 175.A PL7 (build 13311 32 bit). Abilene network topology was modelled using Riverbed as shown in the following figures.

Figure 3.2 shows a setup for routers in the core of the network, set between Seattle and New York, and the associated profile and application configurations. The routers carry traffic from the sender to the receiver across the network. The application and configuration profiles for the whole simulation are detailed in Figure 3.5.

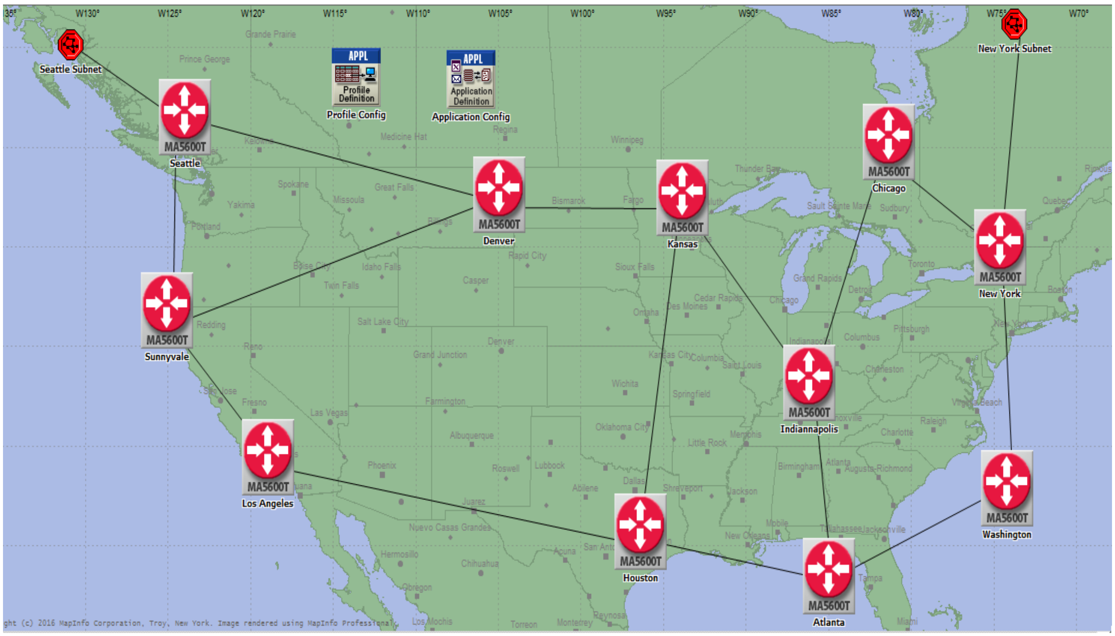

Figure 3.2: Abilene backbone model setup in Riverbed Modeler [160].

Figure 3.3 shows the setup of the computer and routers on the server side of the network at Seattle. There is only one server setup, which generated all the types of traffic in the network. One server is used for the generation of all traffic that is being simulated because the server can handle as many applications as can be placed on it. A single server ensures all the technology is similar and server configurations are done on a single server. A single server minimizes the number of connections that can be made to a router. This is done to keep the connections to the router very minimal due to modeller restrictions on the number of connections that a router can have. A router that directly connects with the server is named as the Seattle local router and the router that connects Seattle to the core of the network is named Seattle main router.

Figure 3.3: Server-side setup at Seattle [160].

On the client or host side, there were several computers connected to the local router. The host side had two routers in the network. A router directly connecting to the hosts was named as New York local router while the router that connected to the core of the network was named New York main router Figure 3.4 illustrates this.

The number of hosts was varied, between 6 and 48, for both low and high throughput in the network, during the simulation. The six hosts are considered because that is the minimum number of simulation scenarios present for this research and 48 is the maximum number of connections that can be made a router. The number of hosts was initially set at 6, with each computer carrying one of the types of traffic to be simulated. The hosts were then increased to 24 and finally to the maximum of 48.

The throughput in the network was varied between the low and high network throughput to simulate each of the scenarios based on the research questions.

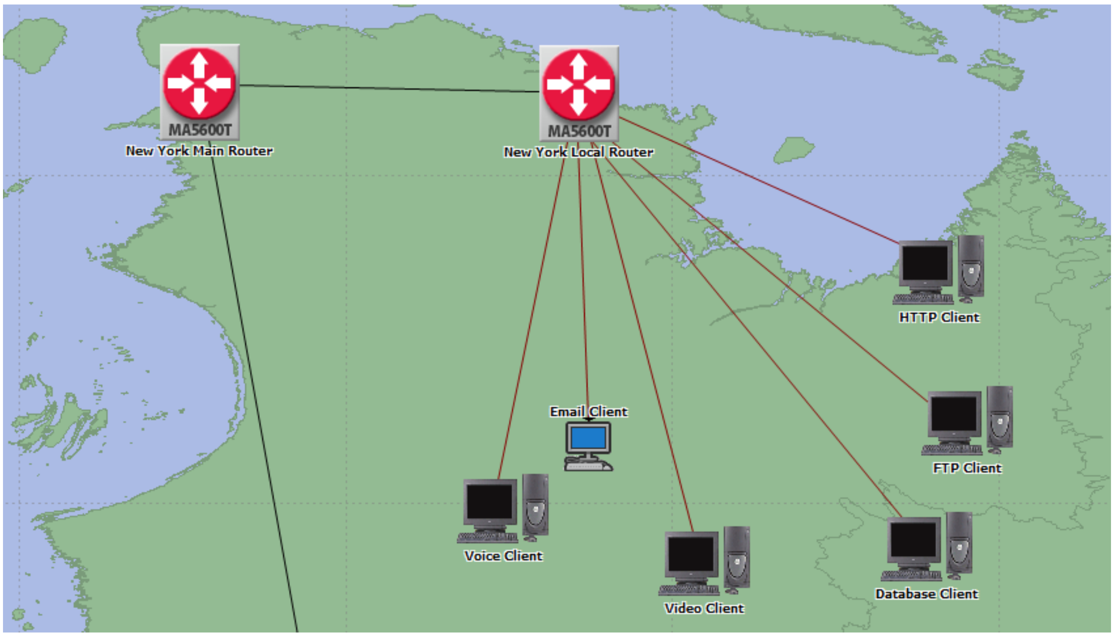

Figure 3.4: Host setup at New York [160].

After the setup of the network in Riverbed, the traffic that was to be modelled was defined. The definition of the traffic was done under application configuration and profile configuration settings. The application configuration settings defined the type of services that were to be offered by the server. For the hosts, the definition of the type of traffic was done under profile configuration settings. Profile configuration settings defined the type of traffic that was to be simulated, mapping it with the services that were being requested for by the network hosts. The definitions were done as illustrated in Figure 3.5.

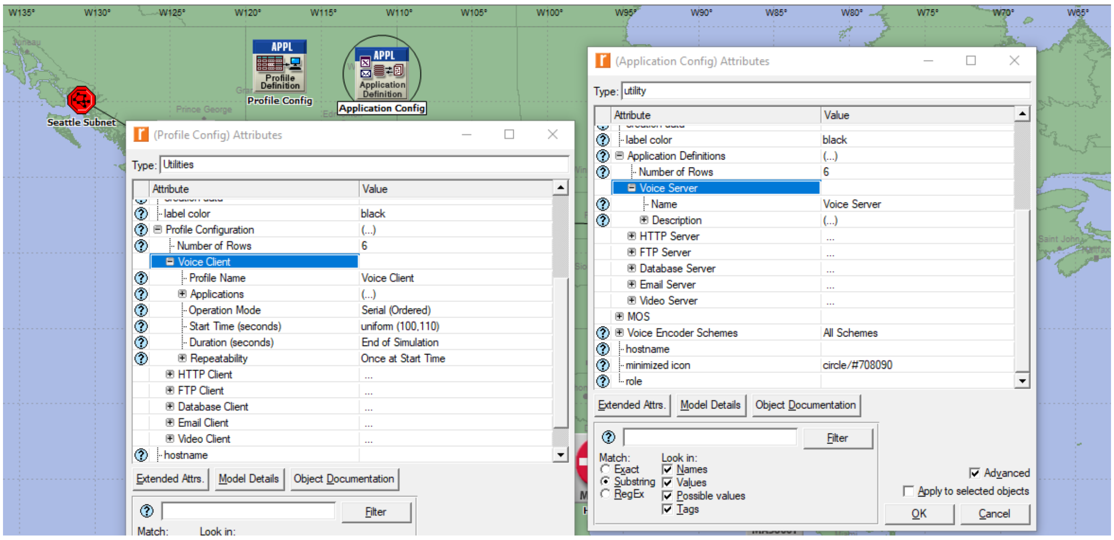

Figure 3.5: Application and profile configuration for the whole network.

The mapping of the application definitions on the server was done as shown in Figure 3.6, where the server was to generate the traffic for the types of applications data definition in Figure 3.5. All services defined under application configuration profile were being offered from one server.

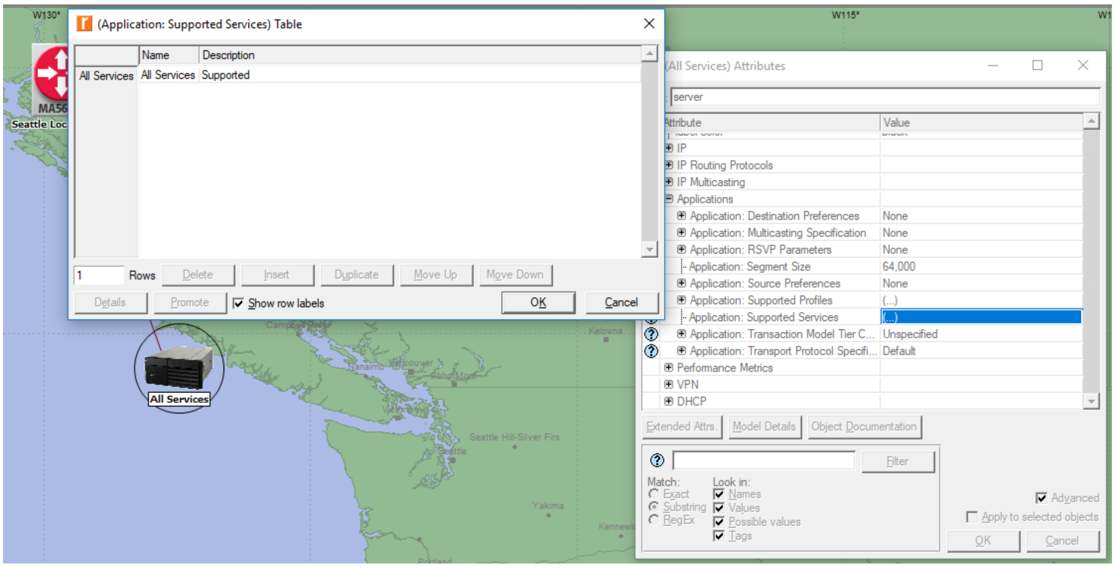

Figure 3.6: Application supported services on the server

The network hosts were then mapped, per computer for each traffic type that was to be requested from the server. Figure 3.7 shows the assignment of a computer to be accessing HTTP resources from the server. The segment size was also defined for each of the hosts.

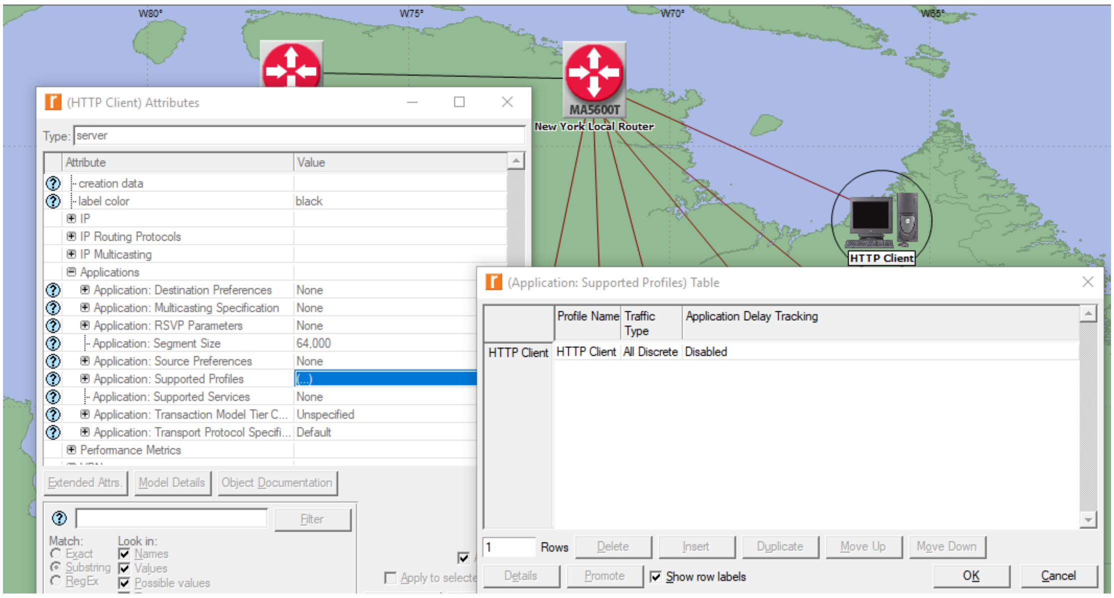

Figure 3.7: Application supported profiles on the hosts

3.6 Simulation Process

Simulations were conducted based on the objectives that were outlined in section 1.6 and research questions in section 1.7. These objectives were mirrored by the research questions to guide the study.

The simulation process involved inputting variables into the Riverbed Modeller and running the simulation with all types of traffic under study in a low and high throughput environment. The variables identified were buffer size and datagram forwarding rate.

Table 3.2 and Table 3.3 show the types of traffic that were defined for the simulations. Application definition represents the type of traffic that was simulated, and value is a parameter for the throughput required during simulation. These values were then configured in Riverbed Modeller based on the throughput, for each type of traffic. The variables were split onto low throughput and high throughput.

Table 3.2: Types of traffic for low throughput in the network.

Application Definition	Value
Voice	PCM Quality and Silence Suppressed
Video	Low resolution video
HTTP (Internet)	Light browsing
Email	Low load
Database	Low load
FTP	Low load

Table 3.3: Types of traffic for high throughput in the network.

Application Definition	Value
Voice	GSM quality and silence suppressed
Video	High resolution video
HTTP (Internet)	Web TV
Email	High load
Database	High load
FTP	High load

In setting up the simulation environment, both types of throughput were made to run in the same environment. The routes in this simulation are computed using the shortest paths using ink metrics as the lengths for the inks using open shortest path first (OSPF), in a multipath environment. OSPF was used as this is the most widely deployed intra domain routing protocol in IP networks [8]. Figure 3.8 shows this environment, with the routing domain legend depicting the type of routing algorithm employed for this simulation.

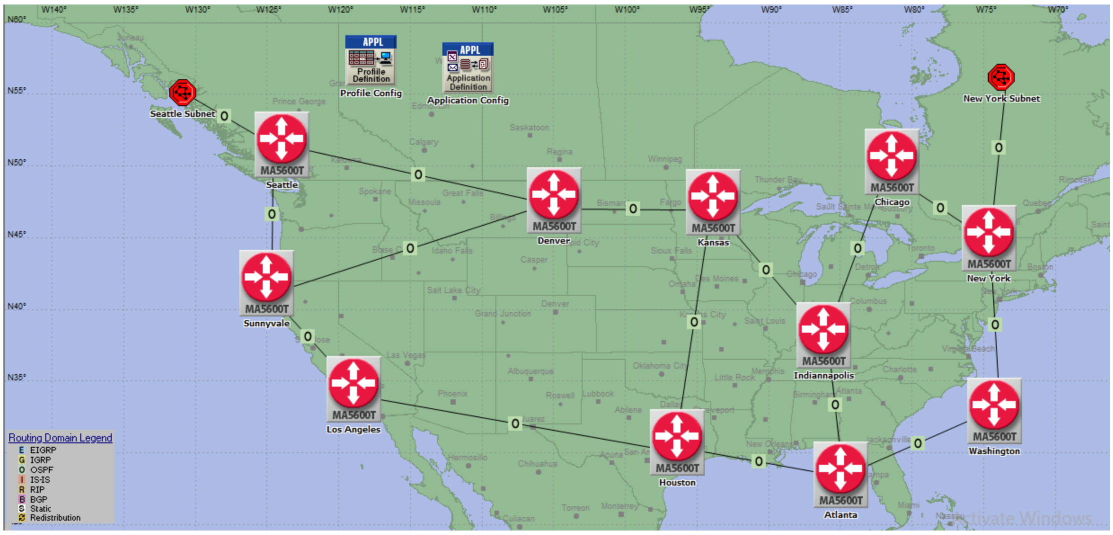

Figure 3.8: Setup with open shortest path first in the network (OSPF).

After the setup was completed, with the types of traffic defined, simulations were then run. The independent variables, buffer size and DFR were varied throughout the simulations. The variations of the variables were done under IP processing information attribute as shown in Figure 3.9. The variables were adjusted until minimum values for buffer size and DFR were found for both low and high throughput environments. The minimum values were points where there was no packet loss during simulation because flows did not leave the sow start This meant there was sufficient buffering in the network and no packet drops were recorded.

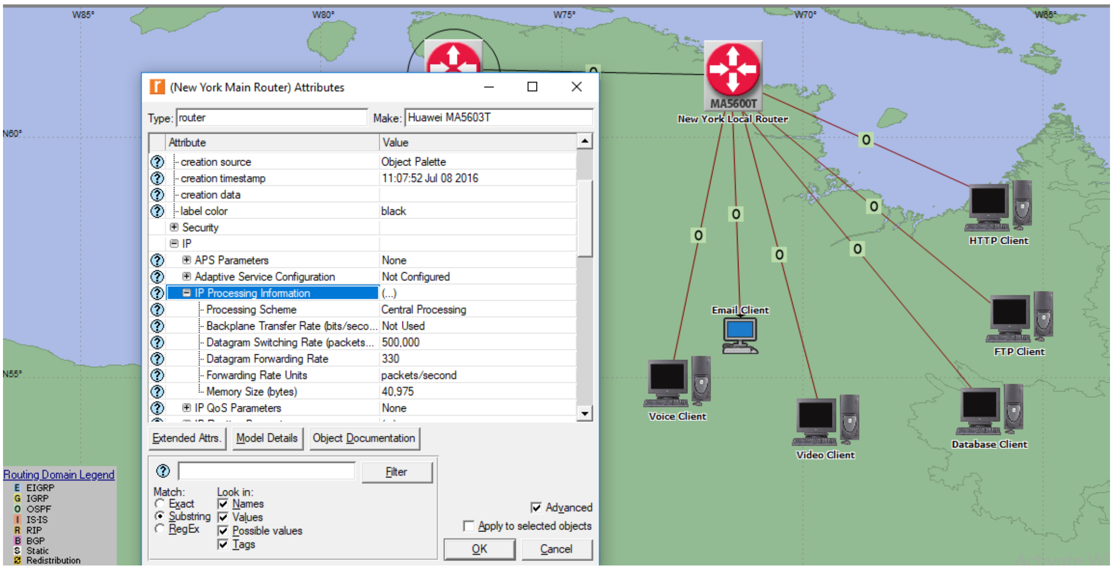

Figure 3.9: Buffer size and datagram forwarding rate configuration.

3.7 Data Analysis Tools

Tools for analysis of the data were Riverbed Modeller, with its inbuilt analysis and graphing capabilities. Graphs produced were analysed within Riverbed and also using Magic Plot Student software. Microsoft excel was also used to analyse the data quantitatively and qualitatively.

Observations made during the simulation also assisted in the analysis of the data that was produced both during the simulation and after the simulation.

3.8 Summary

This chapter outlined the methodology followed to conduct this study and how the simulation was setup. It details the variables which were varied throughout the simulation. The simulations conducted all ran simultaneously during the experiments. The types of traffic both the low and high throughput environment was the same. This was done to keep the traffic uniform.

CHAPTER FOUR

RESULTS AND ANALYSIS

4.1 Introduction

This chapter covers the results of the research that was conducted based on the objectives and research questions in section 1.7, of the study. The results present an alternative model of assignment of buffer in an access network is presented based on simulations using Riverbed Academic Modeller.

4.2 Current Buffer Size Model

The current buffer sizing model used on the Internet is the bandwidth delay product (BDP). BPD or the rule of thumb gives a measure of the required buffer size for a network based on the networks average round trip time (RTT) multiplied by the link capacity for the bottleneck link. As stated in 2.4, the use of TCP for most data communication in the Internet has led to a standardized buffer sizing recommendation based on the BDP, which was obtained experimentally by using at most 8 TCP flows on a 40 Mbps core link in 1994 [97, 112].

BDP is recommended for buffer sizing by Request for Comments (RFCs) and is used by manufactures to assign buffer size. The goal of BDP is to keep links fully utilized, so that the throughput in the network is maximized. In this model, links are kept busy by a buffer size that prevents them from going idle due to congestion in the network. However, TCP “sawtooth” congestion control algorithm is designed to fill any buffers and deliberately causes occasional packet loss to provide feedback to the sender. Therefore, no matter how big we make the buffers at the bottleneck link, TCO will cause the buffer to overflow [18, 19, 32, 161].

The equation for buffer size assignment using BDP is;

$$B = \overline{RTT} \times C \qquad (1)$$

where B is the buffer size, RTT is the average round-trip time of the flow passing through the link and C is the capacity of the link.

4.3 Challenges of Current Buffer Sizing Model

The current buffer sizing model was meant to guarantee a 100% link utilization. The growing link capacities and high-speed networks have proven to be a challenge to the assignment of buffer size based on BDP. The big buffers that result from BDP have led to congestion in the networks, especially in access networks where there is no multipath routing to ease congestion [18, 19, 32].

Current buffer sizing models negatively affect real time traffic due to the lag that big buffers introduce in the network [23, 24].

As an example, for a first generational optical network, with an OC-768 capacity link, a buffer size based on BPD would mean the expected buffer in the network is 1.25 GB. This poses a challenge in router design in terms of buffer space and heating of the machines. The other challenge is the problem of unnecessary packet retransmissions due to packets staying longer in the buffers [18, 21].

These challenges and as extensively covered in the related works mean that there must be a new approach to the problem of buffer sizing. While no one buffer size can be assigned for the whole length of the network, this paper seeks to address these challenges by understanding the linkage between link utilization and buffer size, and packet loss rate based on the input and out rates of packets in a router [23, 24, 32, 74].

As stated in section 1.2, to address the challenges of the current buffer size assignment, there have been alternative buffer size assignment models which have not gained commercial use. The lack of commercial viability is largely due to the challenge of large-scale industrial tests and the commercial feasibility of technologies like all optical routing and buffering [14, 15].

4.4 Results of the Buffer Model Implementation

The model for determining buffer size was implemented using Riverbed Academic Modeller. The focus of the simulation was to find an alternative method for assignment of buffer size in first generation optical network. To achieve this, an access network was connected to a first-generation optical backbone network.

4.4.1 Homogenous Buffer Size with varied Throughput

The second objective of this study was to find out how suitable a homogenous buffer size is for a first-generation optical network running heterogeneous flows. When carrying traffic or TCP flows, a network can have low throughput or high throughput at different times. Therefore, in determining the suitability and feasibility of a homogeneous buffer size for a wide area network, the network was set up in two phases.

The first phase was for the simulation to be run in an environment with low throughput only. This meant that all the traffic types that were being simulated as shown in table 3.2 had a low load on the network. The second was to run the simulation in a high throughput environment. This meant that all the traffic types being simulated as shown in table 3.3, were carrying a high load in the network. Table 4.1 shows the values used in the low throughput environment, with its associated graph in Figure 4.1.

Having a homogenous buffer size and datagram forwarding rate in the network meant that each node in the network had the same buffer size and datagram forwarding rate. All routers in the network to be simulated had the same values for the duration of the simulation.

In these simulations, the buffer size was kept constant in both low throughput and high throughput. Table 4.1 shows a buffer size and DFR in bytes and packets per second respectively. The buffer size was arrived at based on literature that gave an estimate of small buffers that were suitable for traffic in high capacity links. The DFR was arrived based on varying simulations but a minima buffer size for packets in slow start. Packets in slow start do not drop packets, therefore, when simulating, the input parameters were varied, and it was found that the two values for buffer size and DFR were the minimal values at which no packets in the network were dropping.

To maintain buffer size homogeneity, the same minimal values for buffer size and DFR were used with high throughput in the network. Both these simulations had 6 hosts connected to the core network via a common edge or access router.

Table 4.1: Simulation values in low and high throughput environments.

Throughput	Buffer (bytes)	DFR (packets/second)
Low	17975	390
High	17975	390

The values in Table 4.1 were placed into Riverbed Academic Modeller, and a graph on Figure 4.1 is what was produced after simulation. This graph combines the results for low throughput and high throughput, simulations which were done concurrently.

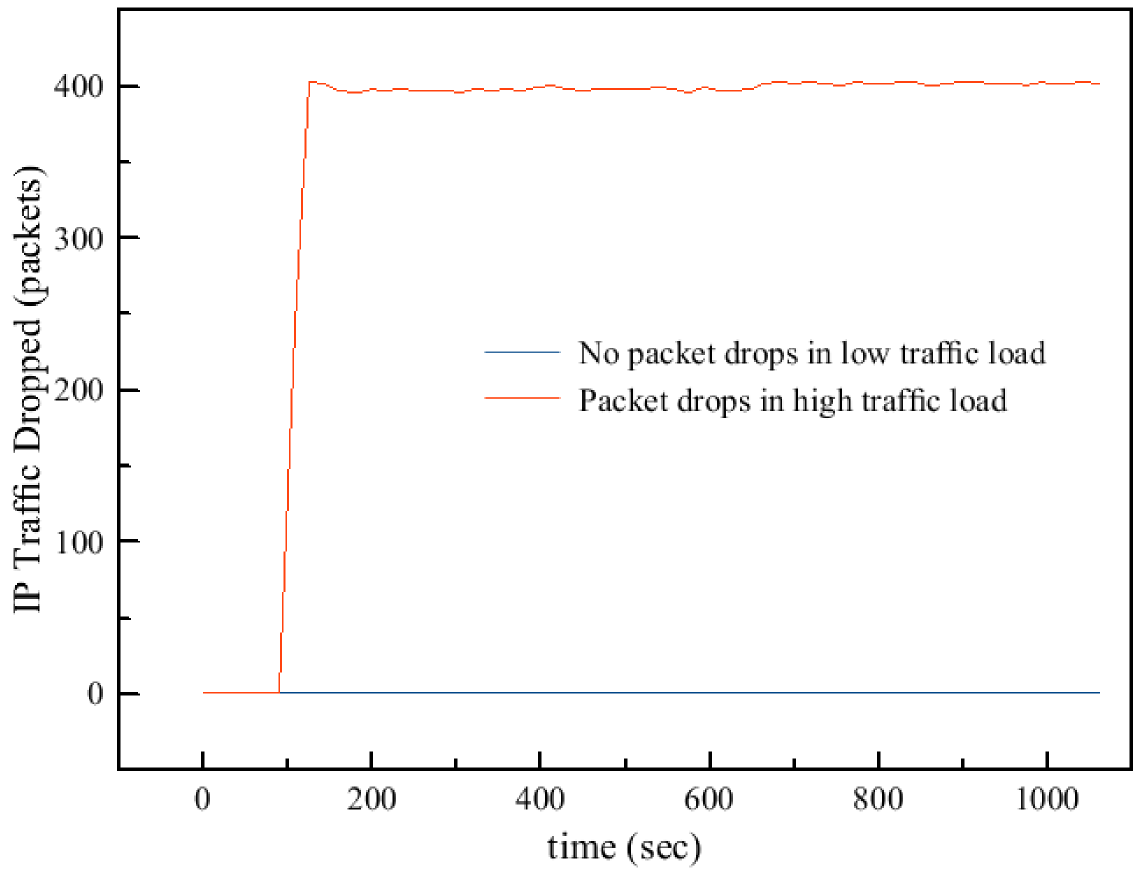

Figure 4.1: Behaviour of traffic in low and high throughput at constant buffer size and DFR [160].

The results in Figure 4.1 show that when the same buffer size and datagram forwarding rate is used in low throughput, there is no packet loss. This means that packets in the network, being generated and received by the 6 hosts in the access network do not leave the slow start state. This result is shown on Figure 4.1 by a straight line that shows packets are not dropping during simulation. When packets are not dropping in the network, it means the buffers are enough and the switching of packets is optimal and for this reason, packets do not leave the slow start phase.

When the same buffer size and datagram forwarding rate is used in a high throughput environment, the network records packet drops. This means that packets leave the slow start phase and the network enters congestion avoidance and thereby drops packets. The packets drops are shown in Figure 4.1 by a graph that moves away from the x-axis almost exponentially. At the start of the simulation, it can be noted that there are no packet drops for about 90 seconds into the simulation. For about 10 seconds, packets start dropping exponentially, from 0 to about 400 packets dropping in less than 10 seconds. Packets maintain a constant dropping rate of 400 packets per second until the end of the simulation.

The simulation in high throughput shows that the buffer size and DFR were not enough for packet routing. Packets reached a constant average drop of 380 packets per second as packets due to the TCP CUBIC implementation used in the simulation. The TCP CUBIC implementation uses an optimized congestion control algorithm for high bandwidth networks with high latency. The window size is a cubic function of time since the last congestion event. This explains the exponential growth of the drop of packets as TCP CUBIC quickly ramps up to the size before the last congestion event.

After the packet drops reach a maximum, packet drop reduces, this is so because TCP CUBIC probes for more bandwidth before the network stabilizes until the end of the simulation. CUBIC allows for more fairness between flows since the window growth is independent of RTT.

4.4.2 Modelling a minimal buffer size for First Generation Optical Network

Based on the minimal values used for low throughput in 4.4.1, this study set out to find proof that indeed there is a minimal value at which traffic in a low throughput network will not leave slow start phase. A phase in the network where no packets will drop. That point will give a minimal buffer size at which routing can take place.

The model was developed in Riverbed as described in section 3.5. There were two scenarios used to determine the minimal buffer size required for this simulation in low and high throughput.

The first scenario was where a constant buffer size and a different DFR was used in low throughput. This is shown in Table 4.2. These values were arrived at by setting network traffic in slow-start. Buffer size was kept constant and DFR varied because buffer size is the main variable in this simulation.

Table 4.2: Simulation values in low throughput environment.

Throughput	Buffer (bytes)	DFR (packets/second)
Low	17975	300
Low	17975	390

The simulation values at shown in Table 4.2 produced an associated graph in Figure 4.2. The figure shows the results that were graphed at the end of the simulation when buffer size was kept constant and DFR was varied. The simulations were run consecutively.

When buffer size was at 17975 bytes and DFR was at 300, the graph shows that there were packet drops in the network. At the start of the simulation, no packets are dropped in the first 90 seconds. After which packets drop exponentially. This is typical of TCP CUBIC which was used in the simulation. The graph fluctuates as TCP CUBIC tries to stabilize as it probes for more bandwidth. CUBIC does not really stabilize as was the case in Figure 4.1 because the buffer size and DFR are not enough for packet routing. The instability occurs throughout the simulation because the constant rate of 300 packets being switched at the router is not enough therefore packets cannot stay in slow start.

When the DFR was increased by an interval of 90, to reach 390 packets per second, while maintain the same buffer size as before, it was noted that there were no packet drops in the network. This is represented by a straight line in Figure 4.2. This meant that packets were able to stay in slow start phase throughout the simulation. The buffer size and DFR were suitable for routing to take place and these were the minimal values at which routing can take place in the network as setup.

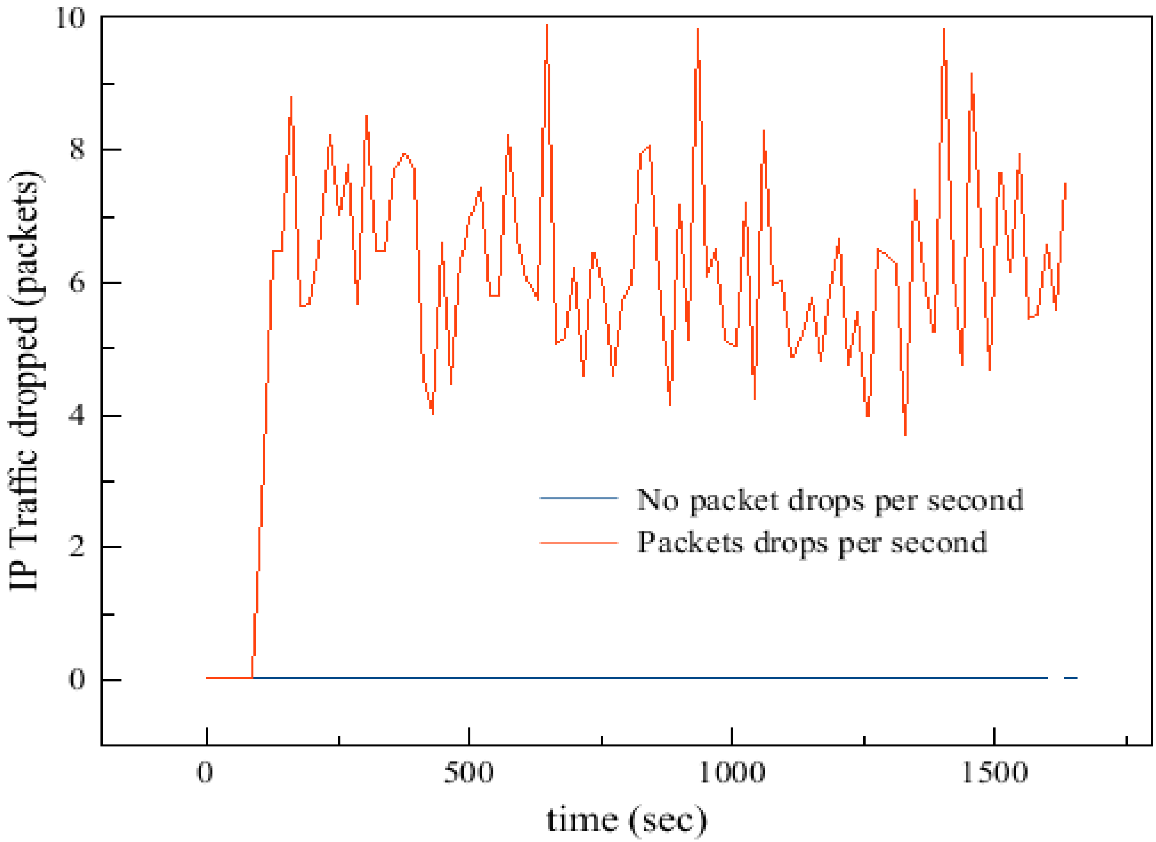

Figure 4.2: Behaviour of traffic in a low throughput environment with 6 hosts [160].

The second scenario was where a simulation with constant buffer size and varied DFR in a high traffic scenario. The buffer size and DFR were arrived at based on the variations that were done with specified increments to both buffer size and DFR until it was confirmed that packets were in the slow start phase. The starting point was buffer size and DFR that was used in Table 4.2, except in a high throughput environment. For high throughput, the values that were taken for simulation were as shown in Table 4.3. This table shows the values that were being simulated to model a minimal buffer size. This is the buffer size at which no packets can be dropped in the network. The associated graph of Table 4.3 is Figure 4.3, which shows the results, in a graphical form, of running two simulations, consecutively.

Table 4.3: Simulation values in high throughput.

Throughput	Buffer (bytes)	DFR (packets/second)
High	40975	300
High	40975	16000

As can be seen from the graph, there were no packet drops in the first 100 seconds of the simulation, after which there was an exponential increase in packet drops in the network.

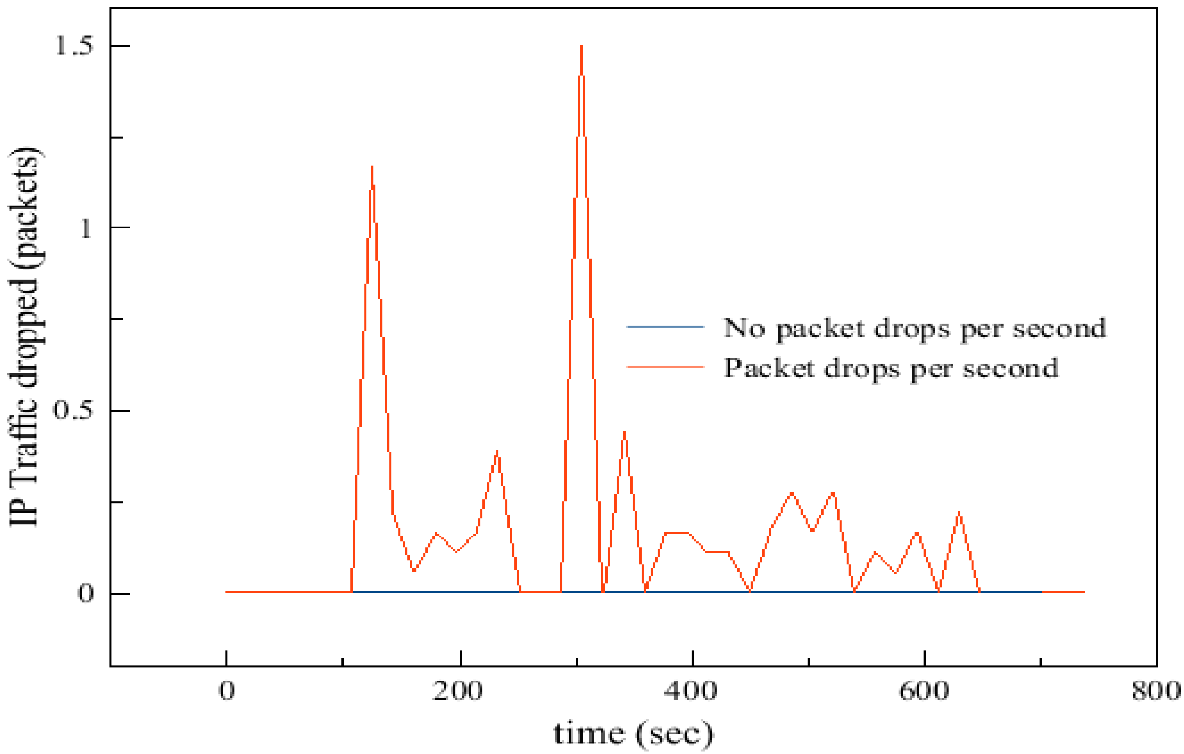

Figure 4.3: Behaviour of traffic in high throughput with 6 hosts [160].

Packet drop reduces, after which it increases close to 180 seconds of simulation. There is a recorded decrease in packet drops after about 250 seconds. This is an interesting point as there are then no packet drops for about 30 seconds, until close to the 300 seconds mark of simulation. After this, the TCP CUBIC algorithm fluctuates until the end of the simulation.

With a constant buffer size, the DFR was changed to a higher value. When the simulation was run, as graphed in Figure 4.3, a horizontal straight line emerges showing that there are no packet drops with this pair of values. That means the packets stayed in slow start phase and did not leave that phase for the duration of the simulation.

From the foregoing, when the simulation did not drop any packets, it means the buffer size and DFR combination was sufficient for the flow of packets in the network. This was true for low and high throughput in the network.

Network congestion was noted once packets started dropping during simulation. This meant that the combination of buffer size and DFR was not enough for packet routing in the network. The number of nodes in the access network was kept at a constant of 6 nodes, representing each of the types of applications that were being simulated.

4.4.3 A model for buffer size assignment

Based on the observed relationship between buffer size and DFR, it was suspected that it can be possible to determine buffer size independent of link capacity. In devising a model, it was important to establish by simulation, whether buffer size can be determined independent of link capacity.

First, a TCP algorithm was chosen, which does not depend on RTT as a metric for determination of congestion in the network. As a guide, in the direction of RTT or link capacity being a metric for buffer size assignment, this study set out to devise a model that can be used to assign buffer size in first generation optical network that was independent of RTT.

Based on the results of simulations in section 4.4.2 and 4.4.3, which showed a relationship between buffer size and DFR, the results were expanded to identify the relationship over a range of values.

As before, the relationship was to be determined in both low throughput and high throughput as networks handle both types of traffic at any given time. The number of nodes in the access network were kept at 6 nodes.

For the simulation in low throughput, the simulation result from Figure 4.2 where no packets were dropped, was picked as a starting point for the paired values of buffer size and DFR. These values were increased at set intervals, as shown in Table 4.4. Buffer size, starting at 17975 bytes, was increased by 500 bytes and DFR was decreased by 1 packet. This was done as it had already been observed in 4.4.1 and 4.4.2 that a change in the values of wither DFR or buffer size created a change in the behaviour of the simulation. Since a minimal buffer size for simulation had already been established in section 4.4.1, the pair values were taken as a starting point for this simulation, where buffer size was increased and DFR was being reduced.

Table 4.4 shows the relationship between buffer size and DFR, as buffer size is being increased and DFR is being reduced. The delay is the network is constant, correct to 2 decimal places. In all the simulations, no packets are dropped because all flows remain in slow start.

Table 4.4 has an associated graph in Figure 4.4, which shows the graphical results of the table. The simulations were carried out for each pair of buffer size and DFR, with average delay recorded in the network.

Table 4.4: Simulation values showing the relationship buffer size and DFR in low throughput.

Buffer (bytes)	DFR (packets/second)	Delay (seconds)
17975	390	0.042711205
18475	389	0.042795787
18975	388	0.04280826
19475	387	0.042826277
19975	386	0.04269029
20475	385	0.042812518
20975	384	0.042912337
21475	383	0.042747008
21975	382	0.042992485
22475	381	0.043157379
22975	380	0.042945778

The graph in Figure 4.4 only considers buffer size and DFR. This is from the values obtained by simulation as outlined in Table 4.4.

The buffer size for this simulation was set at around 1300 bytes. Therefore, Figure 4.4 shows a graph of packets against DFR. From Figure 4.4, it was observed that increasing the buffer size while reducing the datagram forwarding rate revealed an inverse relationship that could be used in the assignment of buffer size in a network. This relationship shows that buffer size can be assigned independent of link capacity.

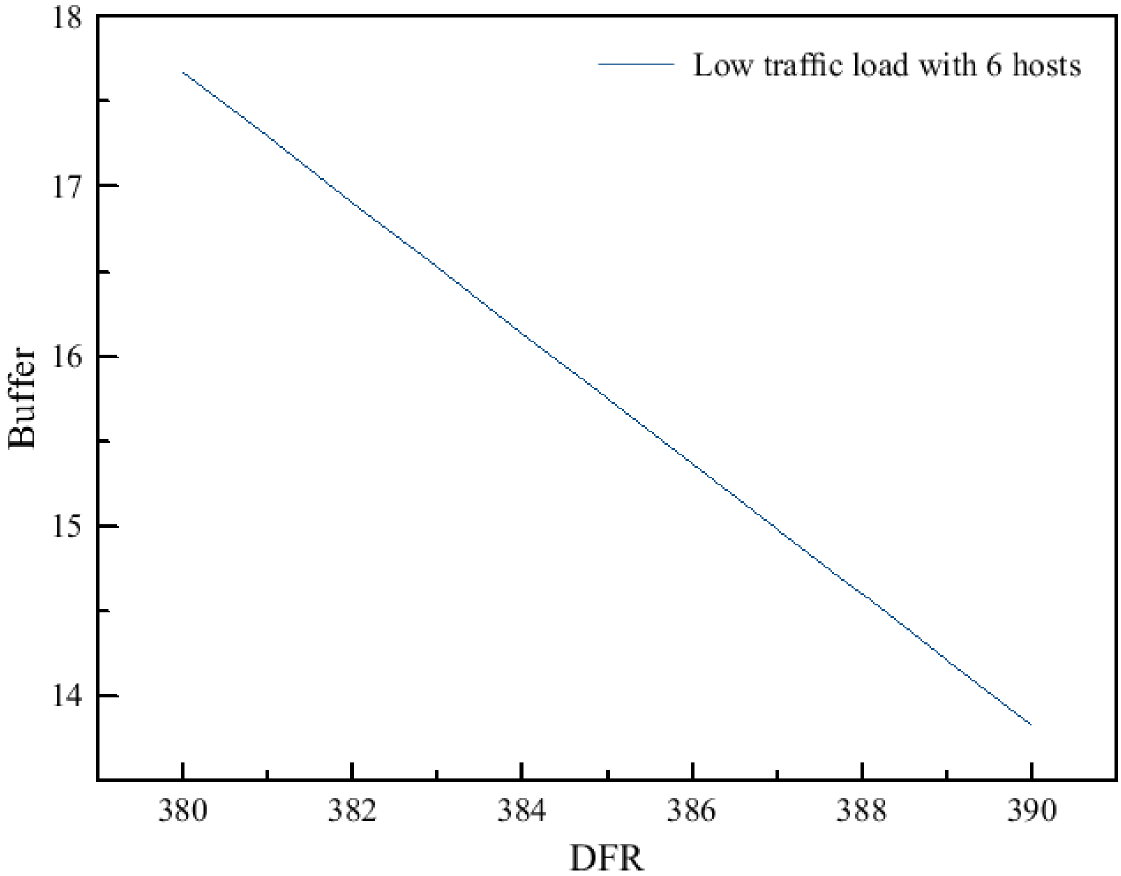

Figure 4.4: The inverse relationship between buffer size and DFR in low throughput.

From this relationship, it holds that the minimum buffer size and DFR for packet routing in a network 17975 bytes and 390 packets per second respectively.

The same process was followed for traffic in high throughput. The process followed was to find a minimum value for buffer size in a network with high traffic volumes. An iterative process was followed to find the minimum buffer for high traffic network. The values are shown in Table 4.5, with the delay in the network showing a constant value correct to 2 decimal places.

Table 4.5: Simulation values showing the relationship buffer size and DFR in high throughput.

Buffer (bytes)	DFR (packets/second)	Delay (seconds)
40975	16000	0.037260521
41475	15900	0.037321333
41975	15800	0.037262802
42475	15700	0.037293674
42975	15600	0.037312844
43475	15500	0.037251109
43975	15400	0.037353057
44475	15300	0.037346795
44975	15200	0.037354849
45475	15100	0.037348653
45975	15000	0.037250752

Buffer size was increased by 500 bytes for each simulation and DFR was reduced by a value of 100 for packets per second. Table 4.5 is then graphed, with buffer size being converted into packets, with an estimate of 1300 bytes to represent 1 packet. The graph does not plot network delay.

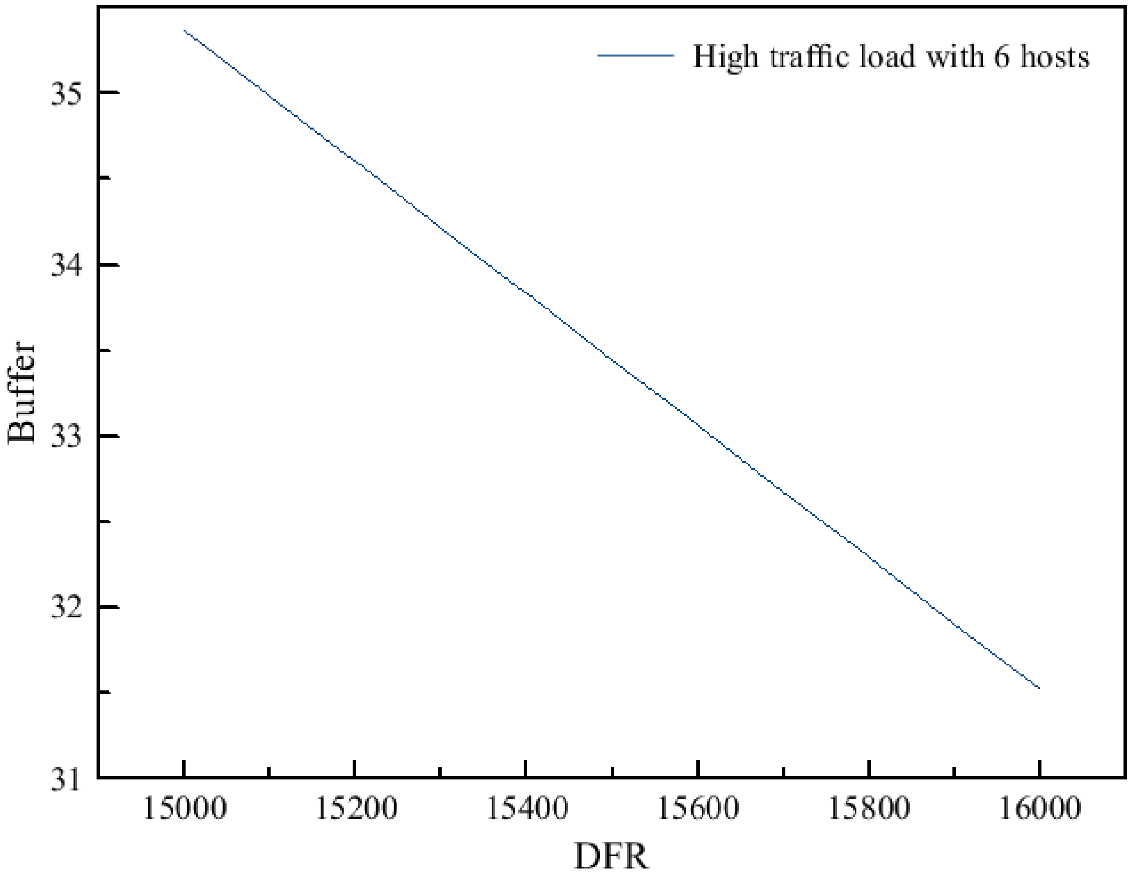

Figure 4.5: The inverse relationship between buffer size and DFR in low throughput.

Figure 4.5 shows an inverse relationship between buffer size and DFR. During this simulation, no packets were dropped. This confirmed that all packets stayed in the slow start phase for the duration of the simulation, hence a similar delay throughout the simulation. The minimum buffer size was found to be 40975 and DFR was found to be 16000 packets per second.

Based on the inverse relationship so established, a model for assigning the minimum buffer size in the network, based on the number of nodes in the access network. With the known variables of buffer size and datagram forwarding rate, a formula was derived based on the inverse relationship, and the constant changes that were being added in the simulation value of buffer size.

$$B = \frac{k}{D} \qquad (2)$$

Where B is the buffer size and D is the datagram forwarding rate and k is the constant of proportionality.

To find the value of k, for a low throughput network, we used the minimum values of B and D from the simulation. The value for B is 1775 bytes while the value for D is 390 packets per second. In the simulation, a packet was 1300 bytes.

Therefore, the value of B in packets becomes 13.83 and D is 390 packets per second. In finding the value of k, equation 2 was replaced with the known variables and the value of k was determined to be 5393.70 and rounded off to 5400. From this, an equation for the assignment of the minimum buffer size was derived to be;

$$B = \frac{5400}{D} + 14 \qquad (3)$$

Where B is the buffer size and D is non-zero and is the datagram forwarding rate. 14 is added to the equation as this is the minimum required buffer, in packets, required for routing to take place for 6 nodes in the access network.

Further simulations were done by doubling the number of hosts in the access network while maintaining the values in Table 4.3. It was found that packets were dropping in the network as traffic could not stay in slow start for the duration of the simulation. A further increase to the maximum number on the router of 48, also showed drop in packets during simulation.

Upon further observation and analysis of the results of the simulation, it was observed that packets were only dropping at the New York local router. This local router was the one which had a direct connection to the nodes in the network.

The router had insufficient buffer for routing to take place. This was the bottleneck link of the network. The buffer size was doubled at the local router where packets were dropping, and the simulation run again, while maintain the parameters in the rest of the network. It was observed that a doubling of the buffer size did not produce packet drops in the network.

The formula for the assignment of buffer size in an access network with more than 24 nodes now becomes;

$$B = \frac{5400}{D} + 28 \quad (4)$$

Where B is the buffer size, D is the datagram forwarding rate and 28 is the minimum buffer size required for routing to take place, where 24 or more hosts are connected in a network.

For a high throughput network, with the established inverse relationship in equation (2) based on low throughput, the same relationship holds for high throughput, having at most 48 nodes in the access network.

The minimal values for routing in a network are buffer size equals to 40975 bytes and datagram forwarding rate equal to 16000 packets per second. The value of k from equal 2 becomes 50400. With a packet equal to 1300 bytes, a model for assignment of buffer size in high throughput become;

$$B = \frac{50400}{D} + 32 \quad (5)$$

Where B is buffer size and D is non-zero and is datagram forwarding rate and 50400 is the constant and 32 is the minimum buffer size required for routing to take place in a network where there are at most 48 nodes in the access network.

4.5 Summary

This chapter addressed the results of the study. The results obtained in this chapter give an estimated model for assignment of buffer size in low and high throughput networks and bring out the relationship between buffer size and DFR. This relationship suggests the assignment of buffer size independent of link capacity.

CHAPTER FIVE

DISCUSSION AND CONCLUSIONS

5.1 Introduction

This chapter gives the discussion, conclusion and recommendations of the research. The model for assignment of buffer size is stated in the chapter and given as a proposed model for buffer assignment in both low and high throughput.

5.2 Discussion

This section discusses the objectives of the research. This study has highlighted why routers need buffers and the importance of TCP in relation to the behaviour of buffer size. All this has been covered in the literature review section. A discussion of the objectives and finding of this research are presented in this section.

5.2.1 To Assess Current Buffer Size Model and Challenges

The first objective of this research was to establish what the current buffer sizing model is for the assignment of buffer size in the Internet. The study established, through secondary data review that bandwidth delay product (BDP) is the model that manufacturers use for the assignment of buffer size. Request for Comments (RFC) guide that BDP be used as a standard when assigning buffer size in the Internet [161].

The current buffer size model was designed in 1994, out of a desire to keep network links fully utilized. Villamizar and Song [19] concluded that because of the dynamics of TCP's congestion control algorithm, a router needed an amount of buffering equal to the average round-trip time of a flow that passes through the router, multiplied by the capacity of the routers' network interface. The study was performed on a 40 Mbps link with few flows. This recommendation has proven to be a change in today's networks. This is due to the rapid increase in the capacity of links which pose a challenge to the design of buffers. With a recommendation of at least 250 ms of buffer in a network, buffer requirements can grow linearly with increases in link capacity. For example, a 10Gbps router line card needs approximately 2.5 Gbits of buffer. This is a huge buffer which creates challenges in both design of the buffer and space.

Several studies have challenged the BDP assignment of buffer size as the recommended buffer for networks. Some studies have proposed small buffers to replace the big buffers while other studies have suggested buffers be made even bigger to accommodate increased traffic in the networks [97, 98, 99].

As highlighted in section 2.5, several methods have been suggested for calculation of buffer size in a network. There is still no consensus on one buffer size for the whole network because it is not ideal to have one buffer size that works for the whole network.

There have been calls for different methods of assigning buffer size because assignment of buffer size depends on several parameters which cannot be universal [111, 117].

Based on the challenges, this study determined to find an alternative model for the assignment of buffer size in first generation optical network. The previous models were not well suited as most models were determining buffer size for the whole network or the core of the network.

5.2.2 To Determine Suitability of a Homogenous Buffer Size

The second objective was to determine whether a homogenous buffer size was suitable for first generation. It is also true that traffic in a network can be low and high in throughput at different times of the day.

The results of the simulations of traffic in low and high throughput, carrying heterogeneous traffic show that a homogenous buffer is not suitable for a first-generation optical network. The results from Figure 4.2 show that when the same buffer size and datagram forwarding arte was used in the network, the low throughput network handled the traffic while when the same parameters were used in a high throughput network, packets were dropped.

The dropping of packets meant that the buffer size and datagram forwarding rate were not enough for packet routing to efficiently take place.

Further simulations showed that access networks must not have the same buffer size as core networks because access network are a bottleneck in routing. This was discovered after increasing the number of nodes in the access network as highlighted in section 4.4.3. In low throughput, when 6 nodes were in the network during simulation, the network did not drop any packets. However, under the same conditions, when access nodes were doubled to ne half of the maximum capacity of the access router, packet drops were observed. Further simulations by doubling the amount of buffer in the access network such that it was twice the minimum buffer for routing, showed that buffer size is indeed dependent on the number of nodes in a network and therefore throughput in the network. Higher throughput with a high buffer size and datagram forwarding ratio value did not pose such a challenge, where packets were dropped during simulation.

Therefore, a homogenous buffer size is suitable only when the buffer size is above 40Kb. This buffer size is ideal in both low and high traffic networks when access nodes are not more than 48.

5.2.3 To Devise and Design a Model for Buffer Size

The third objective of the study was to come up with an alternative model for assignment of buffer size in a first negation optical network. A model in this case is a way in which buffer size can be assigned in the network.

This study comes up with a suggested model for both low and high throughput in the network. Equation three (3) is suggested as a model that can be used for the assignment of buffer size in a low throughput network. Equation four (4) is suggested as a model for the assignment of buffer size in a high throughput network.

5.2.4 Comparison with Other Similar Works

The suggested models for this research differ from all the models discussed in the chapter 2 section 2.4. The difference is that this model does not use RTT or link capacity as a factor in assignment of buffer size.

The advantage of not tying buffer size to link capacity is because of the challenge of rapidly growing link capacities which mean a linear growth in the buffer size required. This can be costly and presents challenges in the design of the router. Big routers also have delay problems and packet retransmission problems.

The challenge with the model presented in this study is a lack of actual testing an a real network so as to put the suggested values to a real world test.

5.2.5 Possible Application

The work in this study can be applied in access network which use first generation optics. First generational optical networks switch their packets in an electronic mode. Such networks are very common in rural and peri-urban Zambia, including most parts of Lusaka. Currently, nearly all buffers use the bandwidth delay product as the standard for assignment of buffer size.

5.3 Conclusion

This study has suggested a buffer sizing method for assignment of buffer size in an access network. Buffer size is dependent on throughput in the network. This study therefore recommends equation (5) as an alternative for the assignment of buffer size in access networks.

5.4 Future Works

Immediate further works for this study will be the development of an interface from which network administrators can fine tune buffer size depending on the needs of their organisation.

The relationship between buffer size and the number of nodes can be studies further for better understanding of the relationship between buffer size and the number of nodes that an access router can support. Further work for this study will require development of an interface which can be used to assign buffer size depending on traffic in the network.

5.5 Summary

This chapter discussed and presented the recommendations and challenges for determining buffer size in first generational optical networks. With network capacity increasing almost yearly, large buffers as are used now only serve to increase latency and unnecessary retransmission of packets. An alternative way of assigning buffer size is presented in this chapter. The proposed model is for an access networks that avoids the use of link capacity as a factor in assignment of buffer size.

REFERENCES

[1] "What is a Computer Network? - Types & Definition - Video & Lesson Transcript | Study.com", Study.com, 2019. [Online]. Available: https://study.com/academy/lesson/what-is-a-computer-network-types-definition-quiz.html. [Accessed: 13- May- 2019].

[2] T. Bakardjieva. Lecture, Topic: “Introduction to Computer Networking”, Institute of Technology, Verna Free University “Chernorizec Hrabar”. [Online]. Available: https://www.vfu.bg/en/e-Learning/Computer-Networks--Introduction_Computer_Networking.pdf [Accessed: 10-May-2019].

[3] M. Benaiah, D. Kumar, B. Deepa. “Computer Networking: A Survey”. *International Journal of Trend in Research and Development*, vol., 2(5), pp. 126-130, Sep-Oct 2015.

[4] J. Bernstein, “Peer to Peer vs. Client-Server Networks,” Online Computer Tips, 19-Jul-2018. [Online]. Available: https://www.onlinecomputertips.com/support-categories/networking/673-peer-to-peer-vs-client-server-networks. [Accessed: 13-May-2019].

[5] “Introducing Basic Network Concepts”, ww3.nd.edu. [Online]. Available:https://www3.nd.edu/~cpoellab/teaching/cse40814_fall14/networks.pdf [Accessed: 10-May-2019].

[6] B. Mukherjee. *Optical WDM Networks*. Davies, CA: Springer Science & Business Media, 2006, pp. 15-39.

[7] S.F. Shaukat, U. Ibrahim, S. Nazir. “Monte Carlo Analysis of Broadband Passive Optical Networks”, *World Applied Sciences Journal*, vol. 12 (8): pp. 1156-1164, 2011.

[8] L. G. Kazovsky, W. Shaw, D. Gutierrez, N. Cheng, S. Wong. “Next Generation Optical Access Networks.” Journal of Lighwave Technology, vol. 25(11), pp. 3428-3442, 2007.

[9] “Introduction to Computer Networking”, tmv.edu.in, [Online]. Available: http://www.tmv.edu.in/pdf/Distance_education/BCA%20Books/BCA%20II%20SEM/BCA-221%20Network%20Fundamentals.pdf .[Accessed 02-Feb-2019].

[8] S. Chamberland. “Designing Reliable IP Networks with an Access/Edge/Core Hierarchical Structure”, *INFOR: Information Systems and Operational Research*, vol. 47(2),

pp. 117-131, 2009.

[10] M. Ilyas, H. T. Mouftah, *The Handbook of Optical Communication Networks*, CRC Press, New York, 2003

[11] What is access network? - Definition from WhatIs.com," WhatIs.com. [Online]. Available: https://whatis.techtarget.com/definition/access-network. [Accessed: 13-May-2019].

[12] G. I. Papadimitriou et al., *Multiwavelength Optical LANs*, Chichester: West Sussex, John Wiley and Sons Ltd, 2003, pp. 12-71.

[13] "What is Packet Buffer? - Definition from Techopedia," Techopedia.com. [Online]. Available: https://www.techopedia.com/definition/2796/packet-buffer. [Accessed: 13-May-2019].

[14] D. Fiems, J.-P. L. Dorsman, and W. Rogiest, "Analysing queueing behaviour in void-avoiding fibre-loop optical buffers," *Performance Evaluation*, vol. 103, pp. 23–40, 2016.

[15] A. Singh, A. K. Tiwari, and R. Srivastava, "Design and analysis of hybrid optical and electronic buffer based optical packet switch," *Sādhanā,* vol. 43, no. 2, 2018.

[16] R. Banner and A. Orda, "Multipath Routing Algorithms for Congestion Minimization," *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 413–424, 2007.

[17] P. Boustead and J. Chicharo, "IP forwarding alternatives in cell switched optical networks," *2000 IEEE International Conference on Communications. ICC 2000. Global Convergence Through Communications. Conference Record.*

[18] L. Sequeira, J. Fernández-Navajas, J. Saldana, J. R. Gállego, and M. Canales, "Describing the Access Network by means of Router Buffer Modelling: A New Methodology," *The Scientific World Journal*, vol. 2014, pp. 1–9, 2014.

[19] C. Villamizar and C. Song, "High performance TCP in ANSNET," *ACM SIGCOMM Computer Communication Review,* vol. 24, no. 5, pp. 45–60, 1994.

[20] A. Vishwanath, V. Sivaraman, and G. N. Rouskas, "Considerations for Sizing Buffers in Optical Packet Switched Networks," *IEEE INFOCOM 2009* - The 28th Conference on Computer Communications, 2009.

[21] A. Dhamdhere and C. Dovrolis, “Open issues in router buffer sizing,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, p. 87, 2006.

[22] J. Sommers, P. Barford, A. Greenberg, and W. Willinger, “An SLA perspective on the router buffer sizing problem,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, no. 4, pp. 40–51, 2008.

[23] J. Saldana, J. Fernandez-Navajas, J. Ruiz-Mas, E. V. Navarro, and L. Casadesus, “The effect of router buffer size on subjective gaming quality estimators based on delay and jitter,” *2012 IEEE Consumer Communications and Networking Conference (CCNC)*, 2012.

[24] F. Chang, W.-C. Feng, W.-C. Feng, and J. Walpole, “Provisioning on-line games,” *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 3, pp. 18–18, 2002.

[25] E. F. Burmeister, D. J. Blumenthal, J. E. Bowers. Optical Buffering for Next-generation Routers, [Online] Available: https://www.ece.ucsb.edu/uoeg/publications/papers/burmeister06oec.pdf [Accessed 03-Jan-2019].

[26] N. Beheshti, Y. Ganjali, R. Rajaduray, D. Blumenthal, and N. McK-eown, “Buffer sizing in all-optical packet switches,” *in Proc. OFC*,2006, Paper OThF8

[27] R. Takahashi, T. Nakahara, K. Takahata, H. Takenouchi, T. Yasui, N. Kondo, and H. Suzuki, “Photonic Random-Access Memory for 40-Gb/s 16-b Burst Optical Packets,” *IEEE Photonics Technology Letters*, vol. 16, no. 4, pp. 1185–1187, 2004.

[28] M. K. Dutta, Comparative Performance Analysis of Fibre Delay Line Based Buffer Architectures for Contention Resolution in Optical WDM Networks, 2016

[29] Dhamdhere, A.; Jiang, H.; Dovrolis, C., "Buffer sizing for congested Internet links," *IEEE Proceedings of INFOCOM, 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*

[30] Prasad, R.S., Dovrolis, C., Thottan, M., “Router Buffer Sizing for TCP Traffic and the Role of the Output/Input Capacity Ratio,” *IEEE/ACM Transactions on Networking*, 2009, Vol. 17(5) pp. 1645 - 1658.

[31] S. A. Soomro, M. M. Shaikh, A. F. Chandio, N. Nizamani, E. A. Buriro. “Impact of Buffer Size on UDP Performance for Real-Time Video Streaming Application,” International Journal of Computer Science and Network Security, vol. 18(6), 2018, pp. 157-160.

[32] R. Tucker, P.-C. Ku, and C. Chang-Hasnain, "Slow-light optical buffers: capabilities and fundamental limitations," *Journal of Lightwave Technology*, vol. 23, no. 12, pp. 4046–4066, 2005.

[33] "Extreme Networks: Congestion Management and Buffering in Data Centre Networks, a Solution White Paper," [Online]. Available: https://people.ucsc.edu/~warner/Bufs/Extreme-Buffer-WP.pdf ,2015, [Accessed: 20-Dec-2018].

[34] M. Enachescu, Y. Ganjali, A. Goel, N. Mckeown, and T. Roughgarden, "Part III," *ACM SIGCOMM Computer Communication Review,* vol. 35, no. 3, p. 83, 2005.

[35] Y.-K. Yeo, J. Yu, and G.-K. Chang, "A Dynamically Reconfigurable Folded-Path Time Delay Buffer for Optical Packet Switching," *IEEE Photonics Technology Letters*, vol. 16, no. 11, pp. 2559–2561, 2004.

[36] C.-H. Chen, L. Johansson, V. Lal, M. Masanovic, D. Blumenthal, and L. Coldren, "Programmable optical buffering using fiber Bragg gratings combined with a widely-tunable wavelength converter," *OFC/NFOEC Technical Digest. Optical Fiber Communication Conference*, 2005., 2005.

[37] A. Agarwal, L. Wang, Y. Su, and P. Kumar, "All-optical loadable and erasable storage buffer based on parametric nonlinearity in fiber," *Journal of Lightwave Technology*, vol. 23, no. 7, pp. 2229–2238, 2005.

[38] A. Liu, C. Wu, Y. Gong, and P. Shum, "Dual-Loop Optical Buffer (DLOB) Based on a 3/spl times/3 Collinear Fiber Coupler," *IEEE Photonics Technology Letters*, vol. 16, no. 9, pp. 2129–2131, 2004.

[39] K. L. Hall and K. A. Rauschenbach, "100-Gbit/s bitwise logic," *Optics Letters*, vol. 23, no. 16, p. 1271, 1998.

[40] E. Burmeister and J. Bowers, "Integrated gate matrix switch for optical packet buffering," *IEEE Photonics Technology Letters*, vol. 18, no. 1, pp. 103–105, 2006.

[41] J. V. Olmos, N. Chi, G. Zervas, D. Simeonidou, S. Yu, I. T. Monroy, and A. Koonen, "Optical node with time-space-and-wavelength domain contention resolution, deflection and dropping capability," *Optics Express*, vol. 14, no. 24, p. 11545, 2006.

[42] D. Blumenthal, (2001). "Photonic packet switching and optical label swapping." *Optical*

Networks Magazine. 2. pp. 54-65.

[43] S. Bregni and A. Pattavina, “Performance Evaluation of Deflection Routing in Optical IP Packet-Switched Networks,” *Cluster Computing*, vol. 7, no. 3, pp. 239–244, 2004.

[44] E. Modiano, "Traffic grooming in WDM networks," *in IEEE Communications Magazine*, vol. 39, no. 7, pp. 124-129, July 2001

[45] R. S. Prasad, C. Dovrolis, and M. Thottan. Router Buffer Sizing Revisited: The Role of the Output/Input Capacity Ratio. *In ACM CoNEXT*, USA, 2007.

[46] Leonid G. Kazovsky, Wei-Tao Shaw, David Gutierrez, Ning Cheng, and Shing-Wa Wong, "Next-Generation Optical Access Networks," *J. Lightwave Technol*. 25, 3428-3442 (2007)

[47] James F. Mollenauer, "Functional Requirements for Broadband Wireless Access Network"', *IEEE 802 Broadband Wireless*, Access Study Group, March 5, 1999.

[48] R. Sanchez, “What is TCP/IP and How Does It Make the Internet Work?,” *HostingAdvice.com*, 17-Nov-2015. [Online]. Available: https://www.hostingadvice.com/blog/tcpip-make-internet-work/. [Accessed: 13-May-2019].

[49] Cui, Cheng, "Study on the Performance of TCP over 10Gbps High Speed Networks" (2013). LSU Doctoral Dissertations. 2102. [Online]. Available: https://digitalcommons.lsu.edu/gradschool_dissertations/2102 [Accessed: 20-Dec-2018]

[50] C. Hunt, *TCP/IP Network Administration*. Sebastopol, CA: OReilly & Associates, Inc., 1998.

[51] R. Jain and K. Ramakrishnan, “Congestion avoidance in computer networks with a connectionless network layer: Concepts, goals and methodology,” *in Computer Networking Symposium, 1988. Proceedings of the. IEEE*, 1988, pp. 134–143.

[52] “Transmission Control Protocol (TCP); Comparison of TCP Congestion Control Algorithms using NetSim,” 2017 tetcos.com. [Online]. Available: https://www.tetcos.com/downloads/TCP_Congestion_Control_Comparison.pdf [Accessed: 01-Nove-2018].

[53] “BBR: Congestion-Based Congestion Control,” *BBR: Congestion-Based Congestion Control - ACM Queue*. [Online]. Available: https://queue.acm.org/detail.cfm?id=3022184.

[Accessed: 13-May-2019].

[54] “Congestion Avoidance in TCP,” *CS558a Syllabus & Progress*. [Online]. Available: http://www.mathcs.emory.edu/~cheung/Courses/558a/Syllabus/6-transport/TCP.html. [Accessed: 13-May-2019].

[55] Administrator, “TCP (Transmission Control Protocol) Congestion Control,” *Noction*, 19-Apr-2019. [Online]. Available: https://www.noction.com/blog/tcp-transmission-control-protocol-congestion-control. [Accessed: 13-May-2019].

[56] H. N. Jasem et al., The TCP-Based New AIMD Congestion Control Algorithm, *IJCSNS* Vol.8 No. 10, October 2008

[57] F. Fund, “TCP congestion control in lossy wireless networks,” *Run my experiment on GENI*, 05-Apr-2017. [Online]. Available: https://witestlab.poly.edu/blog/tcp-congestion-control-in-lossy-wireless-networks/. [Accessed: 13-May-2019].

[58] V. Cerf, V. Jacobson, N. Weaver, and J. Gettys, “BufferBloat: Whats Wrong with the Internet?,” *Queue*, vol. 9, no. 12, pp. 10–20, 2011.

[59] A. Svaraman, https://cs.nyu.edu/courses/fall17/CSCI-UA.0480-009/lectures/lec7.pdf, Lecture Notes, 2017.

[60] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, “Bbr,” *Communications of the ACM*, vol. 60, no. 2, pp. 58–66, 2017.

[61] L. Cai, X. Shen, J. Pan, and J. Mark, “Performance analysis of TCP-friendly AIMD algorithms for multimedia applications,” *IEEE Transactions on Multimedia*, vol. 7, no. 2, pp. 339–355, 2005.

[62] H. Zhang, “Service disciplines for guaranteed performance service in packet-switching networks,” *Proceedings of the IEEE*, vol. 83, no. 10, pp. 1374–1396, 1995.

[63] H. Zhaing et al., “Evaluation of different TCP congestion control algorithm ...” [Online]. Available: http://www2.ensc.sfu.ca/~ljilja/ENSC835/Spring02/Projects/bian_zhang.hilary/Hilary_and_Bian_Report.pdf. [Accessed: 20-Jan-2019].

[64] V. Jacobson, “Congestion Avoidance and Control”, A*CM SIGCOMM Computer Communication Review*, v.18 n.4, p.314-329, August 1988. [Online]. Available:

http://www.sfu.ca/~zbian/courses/cmpt885/congavoid.ps [Accessed: 01-May-2019].

[65] K. Fall and S. Floyd, “Simulation-based comparisons of Tahoe, Reno and SACK TCP,” *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 3, pp. 5–21, 1996.

[66] V. Jacobson. “Modified TCP Congestion Avoidance Algorithm”, Technical report, 30 Apr. 1990. [Online]. Available: ftp://ftp.ee.lbl.gov/email/vanj.90apr30.txt. [Accessed: 20-jan-2019]

[67] Mo, Jeonghoon, Richard J. La, Venkat Anantharam, and Jean Walrand. "Analysis and comparison of TCP Reno and Vegas." *In IEEE INFOCOM'99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now* (Cat. No. 99CH36320), vol. 3, pp. 1556-1563. IEEE, 1999.

[68] Fahmy, S.; Karwa, T.P. (2001): TCP congestion control: Overview and survey of ongoing research. Computer Science Technical Reports, vol. 11, pp. 1-12.

[69] Harjinder Kaur and Gurpreet Singh, TCP Congestion Control and Its Variants,Advances in Computational Sciences and Technology,ISSN 0973-6107 Volume 10, Number 6 (2017) pp. 1715-1723,© Research India Publications, http://www.ripublication.com

[70] Bhatia, K. and Valanjoo, A., 2011, “Progress of Different TCP Variant”, *Journal of Engineering Sciences & Research Technology*, vol.11, pp. 3843- 3848.

[71] “TCP High Speed Variants - eduPERT KB,” *GÉANT federated confluence.* [Online]. Available: https://wiki.geant.org/display/public/EK/TCP High Speed Variants. [Accessed: 13-May-2019].

[72] tcp_bbr: add BBR congestion control, N. Cardwell, V. Jacobson, Y. Cheng, N. Dukkipati, E. Dumazet, S. Yeganeh, commit 0f8782ea14974ce992618b55f0c041ef43ed0b78, Linus Torvalds's Linux Git tree, [September 2016]

[73] Peter L Dordal. An Introduction to Computer Networks, September 2018. [Online]. Available: http://intronetworks.cs.luc.edu/current/ComputerNetworks.pdf [Accessed:07-Oct-2018].

[74] J. Gettys and K. Nichols, “Bufferbloat: Dark Buffers in the Internet,” *Queue*, vol. 9, no. 11, p. 40, 2011.

[75] S. Ha, I. Rhee, and L. Xu, “Cubic: A new TCP-friendly high-speed TCP variant,” *ACM*

SIGOPS Operating Systems Review, vol. 42, no. 5, pp. 64–74, 2008.

[76] V. Jacobson, “Congestion avoidance and control,” *Symposium proceedings on Communications architectures and protocols* - SIGCOMM 88, vol. 18, no. 4, pp. 314–329, 1988.

[77] D. J. Leith, R. N. Shorten, Y. Li and K. Lepikhov, “HamiltonTCP - eduPERT KB,” *GÉANT federated confluence,* 25-Apr-2018. [Online]. Available: https://wiki.geant.org/display/public/EK/HamiltonTCP#. [Accessed: 28-Jun-2018].

[78] K. Lepikhov, “HamiltonTCP - eduPERT KB,” *GÉANT federated confluence*, 25-Apr-2018. [Online]. Available: https://wiki.geant.org/display/public/EK/HamiltonTCP#. [Accessed: 28-Jun-2018].

[79] K. Lepikhov, “WestwoodTCP - eduPERT KB,” *GÉANT federated confluence*, 25-Apr-2018. [Online]. Available: https://wiki.geant.org/display/public/EK/WestwoodTCP. [Accessed: 28-Jun-2018].
[80] https://meetings.internet2.edu/media/medialibrary/2017/05/04/20170425-barry-africa-sig.pdf

[81] “ZAMREN | Zambia,” *UbuntuNet Alliance.* [Online]. Available: https://ubuntunet.net/members/nren/zamren/. [Accessed: 29-Jan-2019].

[82] *ZESCO*. [Online]. Available: http://www.zesco.co.zm/ourBusiness/fibreCom. [Accessed: 29-Jan-2019].

[83] *Study.com.* [Online]. Available: https://study.com/academy/lesson/backbone-networks-types-uses.html. [Accessed: 29-Dec-2018].

[84] “Backbone Networks Analysis,” *Fiber Optic Solutions*, 24-Oct-2016. [Online]. Available: http://www.fiber-optic-solutions.com/analysis-backbone-networks.html. [Accessed: 20-Feb-2019].

[85] *Association for Technology in Music Instruction and The College Music Society in San Diego.* [Online]. Available: https://fredkersten.com/ATMICALIFORNIA/Page1.htm. [Accessed: 29-Jan-2019].

[86] B. Quoitin, "Towards more representative internet topologies." *Universit Catholique de Louvain,* Tech. Rep 2010.
[87] S. Floyd and V. Paxson, “Difficulties in Simulating the Internet,” *IEEE/ACM Trans. Net.,*

vol. 9, no. 4, 2001, pp. 392–403.

[86] M. H. Kabir, S. Islam, J. Hossain and S. Hossain, “Detail Comparison of Network Simulators,” *International Journal of Scientific & Engineering Research*, Volume 5, Issue 10, October-2014 ISSN 2229-5518.

[87] M. S. Hasan, C. Harding, H. Yu, and A. Griffiths, “Modeling delay and packet drop in Networked Control Systems using network simulator NS2,” *International Journal of Automation and Computing*, vol. 2, no. 2, pp. 187–194, 2005.

[88] L. Hogie, P. Bouvry and F. Guinand, "An overview of MANETs simulation." *Electronic Notes in Theoretical Computer Science*, vol. 12, no. 025, pp. 81-101, 2006.

[89] V. Mishra and S. Jangale, “Analysis and Comparison of Different Network Simulators,” *International Journal of Application or Innovation in Engineering and Management (IJAIEM)*, Special Issue for International Technological Conference-2014

[90] E. Weingartner, H. V. Lehn, and K. Wehrle, “A Performance Comparison of Recent Network Simulators,” *2009 IEEE International Conference on Communications*, 2009.

[91] L Raja, “Study of Various Network Simulators,” *International Research Journal of Engineering and Technology*, Vol 5, Dec 2018

[92] G. F. Lucio, M. Paredes-Farrera, E. Jammeh, M. Fleury, and M. J. Reed. “OPNET Modeler and NS-2 - Comparing the Accuracy of Network Simulators for Packet-Level Analysis Using a Network Testbed,” *WSEAS Transactions on Computers*, vol. 2, no 3 :700707, July 2003.

[93] N. Beheshti, E. Burmeister, Y. Ganjali, J. E. Bowers, D. J. Blumenthal, and N. Mckeown, “Optical Packet Buffers for Backbone Internet Routers,” *IEEE/ACM Transactions on Networking*, vol. 18, no. 5, pp. 1599–1609, 2010.

[94] M. Karol, M. Hluchyj, and S. Morgan, “Input Versus Output Queueing on a Space-Division Packet Switch,” *IEEE Transactions on Communications*, vol. 35, no. 12, pp. 1347–1356, 1987.

[95] J. Dai and B. Prabhakar, “The throughput of data switches with and without speedup,” *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064).*

[96] G. Appenzeller, I. Keslassy, and N. Mckeown, "Sizing router buffers," *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM 04, 2004.*

[97] Y. S. Hanay, A. Dwaraki, and T. Wolf, "High-performance implementation of in-network traffic pacing," *2011 IEEE 12th International Conference on High Performance Switching and Routing, 2011.*

[98] R. Langenhorst, M. Eiselt, W. Pieper, G. Grosskopf, R. Ludwig,L. Kuller, E. Dietrich, and H. G. Weber, "Fiber loop optical buffer," *Journal of Lightwave Technology, vol. 14*, no. 3, pp. 324–335, Mar.1996.

[99] A. Vishwanath, V. Sivaraman, and M. Thottan, "Perspectives on router buffer sizing," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 2, p. 34, 2009.

[100] S. B. Jacob and S. Verma, "Router Buffer Sizing in Mixed Traffic," *International Journal of Scientific and Engineering Research*, Vol 4, Issue 8, August 2013.

[101] J. Postel, "Transmission Control Protocol," *Information Sciences Institute*, RFC 793, Sep. 1981.

[102] V. Jacobson, "Congestion avoidance and control," *Symposium proceedings on Communications architectures and protocols - SIGCOMM 88*, pp. 314–329, 1988.

[103] A. Vishwanath, V. Sivaraman, and M. Thottan, "Perspectives on router buffer sizing," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 2, p. 34, 2009.

[104] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," *SIGCOMM Computer Communication Review*, vol. 34, no. 4, Oct. 2004

[105] A. Dhamdhere and C. Dovrolis, "Open issues in router buffer sizing," *SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 87–92, Jan. 2006.

[106] T. Wolf, W. Gong, and Y. Cai, "Burstiness as traffic metric in next-generation optical core networks," *2009 IEEE/LEOS Summer Topical Meeting*, 2009.

[107] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden, "Part III: Routers with very small buffers," *SIGCOMM Computer Communication Review*, vol. 35, no. 3, pp. 83–90, Jul. 2005.

[108] G. Hasegawa, T. Tomioka, K. Tada, and M. Murata, “Simulation studies on router buffer sizing for short-lived and pacing TCP flows,” *Computer Communications*, vol. 31, no. 16, pp. 3789–3798, 2008.

[109] V. Sivaraman, H. Elgindy, D. Moreland, and D. Ostry, “Packet pacing in small buffer optical packet switched networks,” *IEEE/ACM Transactions on Networking*, vol. 17, no. 4, pp. 1066–1079, Aug. 2009.

[110] F. Yakubu, A. Mohammed and I. Abdullahi, "Correlation Analysis of Data Rate and Router Buffer Size on TCP Performance using OPNET Simulator," *International Journal of Computer Applications*, vol 30, no. 9, September 2011.

[111] A. Vishwanath, V. Sivaraman, and M. Thottan, “Perspectives on router buffer sizing,” *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 2, p. 34, 2009.

[112] C. Villamizar and C. Song, “High performance TCP in ANSNET,” *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 5, pp. 45–60, 1994.

[113] G. Appenzeller, I. Keslassy, and N. Mckeown, “Sizing router buffers,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, p. 281, 2004.

[114] G. Appenzeller, Sizing Router Buffers. PhD Thesis, Dept of EE, Stanford University, 2005.

[115] K. Avrachenkov, U. Ayesta, and A. Piunovskiy, “Optimal choice of the buffer size in the Internet routers,” *Proceedings of the 44th IEEE Conference on Decision and Control*, Spain, 2005.

[116] L. Andrew, T. Cui, J. Sun, M. Zukerman, K.-T. Ko, and S. Chan, “Buffer sizing for nonhomogeneous TCP sources,” *IEEE Communications Letters*, vol. 9, no. 6, pp. 567–569, 2005.

[117] D. Y. Eun and X. Wang, “Achieving 100% Throughput in TCP/AQM Under Aggressive Packet Marking with Small Buffer,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 945–956, 2008.

[118] G. Raina and D. Wischik, “Buffer sizes for large multiplexers: TCP queueing theory and instability analysis,” *Next Generation Internet Networks*, 2005.

[119] D. Wischik, “Buffer requirements for high-speed routers,” *31st European Conference*

on Optical Communications (ECOC 2005), 2005.

[120] D. Wischik and N. Mckeown, “Part I,” *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 3, p. 75, 2005.

[121] G. Raina, D. Towsley, and D. Wischik, “Part II,” *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 3, p. 79, 2005.

[122] M. Enachescu, Y. Ganjali, A. Goel, N. Mckeown, and T. Roughgarden, “Part III,” *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 3, p. 83, 2005.

[123] M. Enachescu, Y. Ganjali, A. Goel, N. Mckeown, and T. Roughgarden, “Routers with Very Small Buffers,” *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications,* 2006.

[124] N. Beheshti, Y. Ganjali, R. Rajaduray, D. Blumenthal, and N. Mckeown, “Buffer sizing in all-optical packet switches,” *2006 Optical Fiber Communication Conference and the National Fiber Optic Engineers Conference*, 2006.

[125] N. Beheshti, Y. Ganjali, A. Goel, and N. Mckeown, “Obtaining High Throughput in Networks with Tiny Buffers,” *2008 16th International Workshop on Quality of Service*, 2008.

[126] S. Gorinsky, A. Kantawala, and J. Turner, “Link Buffer Sizing: A New Look at the Old Problem,” *10th IEEE Symposium on Computers and Communications (ISCC05),* Spain, 2005.

[127] S. Gorinsky, A. Kantawala, and J. Turner, “Simulation Perspectives on Link Buffer Sizing,” *Simulation*, vol. 83, no. 3, pp. 245–257, 2007.

[128] A. Dhamdhere, H. Jiang, and C. Dovrolis, “Buffer sizing for congested internet links,” *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, USA, 2005.

[129] R. Morris, “TCP behaviour with many flows,” *Proceedings 1997 International Conference on Network Protocols*, USA, 1997.

[130] R. Morris, “Scalable TCP Congestion Control,” In *IEEE INFOCOM,* Israel, 2000.

[131] C. M. Kellett, R. N. Shorten, and D. J. Leith, “Sizing Internet Router Buffers, Active Queue Management, and the Lure Problem,” *Proceedings of the 45th IEEE Conference on*

Decision and Control, 2006.

[132] Y. Zhang and D. Loguinov, “ABS: Adaptive buffer sizing for heterogeneous networks,” *Computer Networks*, vol. 54, no. 14, pp. 2562–2574, 2010.

[133] A. Dhamdhere and C. Dovrolis, “Open issues in router buffer sizing,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, p. 87, 2006.

[134] G. Vu-Brugier, R. S. Stanojevic, D. J. Leith, and R. N. Shorten, “A critique of recently proposed buffer-sizing strategies,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 1, p. 43, 2007.

[135] Y. Ganjali and N. Mckeown, “Update on buffer sizing in internet routers,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 5, p. 67, 2006.

[136] M. Shifrin and I. Keslassy, “Modeling TCP in Small-Buffer Networks,” *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet Lecture Notes in Computer Science*, pp. 667–678, 2008.

[137] R. S. Prasad, C. Dovolis, and M. Thottan, “Router Buffer Sizing Revisited: The Role of Output/Input Capacity Ratio,” *In ACM CoNEXT*, USA, 2007.

[138] G. Appenzeller, I. Keslassy and N. McKeown, “Sizing Router Buffers,” *Stanford HPNG Technical Report TR04-HPNG-06-08-00*

[139] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson, “Netalyzr: Illuminating the Edge Network,” *Proceedings of the 10th annual conference on Internet measurement - IMC 10*, 2010.

[140] M. M.al-Quzwini, “Design and Implementation of a Fiber to the Home FTTH Access Network based on GPON,” *International Journal of Computer Applications*, vol. 92, no. 6, pp. 30–42, 2014.

[141] “What are the restrictions for Riverbed Modeler Academic Edition 17.5,” *Riverbed Support: Knowledge Base.* [Online]. Available: https://supportkb.riverbed.com/support/index?page=content&id=S24443. [Accessed: 13-May-2019].

[142] L. L. C. Revolvy, “‘Abilene Network’ on Revolvy.com,” *Revolvy*. [Online]. Available: https://www.revolvy.com/page/Abilene-Network. [Accessed: 13-May-2019].

[143] “Internet2,” Abilene Network Upgrade to 10 Gbps Complete | *Internet2 News*. [Online]. Available: https://www.internet2.edu/news/detail/1978/. [Accessed: 13-May-2019].

[144] A. Bryman, E. Bell, and B. Harley, *Business research methods: 4th Edition*. Oxford, United Kingdom: Oxford University Press, 2015. p.27

[145] M. N. K. Saunders, P. Lewis, and A. Thornhill, *Research methods for business students: 6th Edition*. New York: Pearson, 2012.

[146] H. R. Bernard, Research Methods in Anthropology. Lanham: AltaMira Press, 2011, p.7

[147] “Dr. Kostas Alexandridis, GISP - Google Scholar Citations.” [Online]. Available: http://scholar.google.com/citations?user=9g3P0cUAAAAJ&hl=en. [Accessed: 29-Feb-2019].

[148] H. Collins, *Creative Research: The Theory and Practice of Research for the Creative Industries*. Bloomsbury Academic & Professional, 2018, p.38.

[149] G. Lancaster, G. Lancaster, and G. Lancaster, *Research methods: a concise introduction to research in management and business consultancy.* Oxford: Butterworth-Heinemann, 2008.

[150] J Wilson, *Essentials of Business Research A Guide to Doing Your Research Project*. Sage Pubns Pvt Ltd, 2010.

[151] H. Collins, *Creative Research: The Theory and Practice of Research for the Creative Industries*. Bloomsbury Academic & Professional, 2018.

[152] M. N. K. Saunders, P. Lewis, and A. Thornhill, *Research methods for business students*. New York: Pearson, 2019.

[153] N. Beheshti, Y. Ganjali, M. Ghobadi, N. Mckeown, and G. Salmon, “Experimental study of router buffer sizing,” *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement conference - IMC 08*, 2008.

[154] N. L. Leech and A. J. Onwuegbuzie, “A typology of mixed methods research designs,” *Quality & Quantity*, vol. 43, no. 2, pp. 265–275, 2007.

[155] J. W. Creswell and J. D. Creswell, *Research design: qualitative, quantitative, and*

mixed methods approaches. Los Angeles: Sage Publications, Inc., 2018.

[156] R. B. Johnson, *Mixed methods research design and analysis with validity: A Primer.* Department of Professional Studies, University of South Alabama, USA, 2014.

[157] W. L. Neuman, *Social Research Methods: Qualitative and Quantitative Approaches*, Seventh Edition, Pearson Education Limited, 2014, England

[158] J. W. Creswell and T. C. Guetterman, *Educational research: planning, conducting, and evaluating quantitative and qualitative research.* New York, NY: Pearson, 2019.

[159] A. Bhattacherjee, *Social science research: principles, methods, and practices*. Tampa, Florida: Anol Bhattacherjee, 2012.

[160] B. M. Bwalya and S. Tembo, "Performance Evaluation of Buffer Size for Access Networks in First Generation Optical Networks," *International Journal of Internet of Things.* [Online]. Available: http://article.sapub.org/10.5923.j.ijit.20170603.02.html. [Accessed: 20-Nov-2018].

[161] CISCO Line Cards [Online]. Available: http://www.cisco.com/en/US/products/hw/modules/ps2710/products_data_sheets_list.html. [Accessed: 30-Mar-2019].