DEVELOPMENT OF MEASUREMENT-BASED ADMISSION CONTROL ALGORITHM FOR EDGE CARRIER ETHERNET NETWORK TO REDUCE PACKET LOSS AND IMPROVE BANDWIDTH UTILIZATION

BY

Aduma Fred OKOH

DEPARTMENT OF ELECTRICAL ANDCOMPUTER ENGINEERING
FACULTY OF ENGINEERING
AHMADU BELLO UNIVERSITY, ZARIA

APRIL, 2017

DEVELOPMENT OF MEASUREMENT-BASED ADMISSION CONTROL ALGORITHM FOR EDGE CARRIER ETHERNET NETWORK TO REDUCE PACKET LOSS AND IMPROVE BANDWIDTH UTILIZATION

BY

Aduma FredOKOH(B. Eng, 2009)

P14EGCM8001

okohaf@gmail.com

SUPERVISORY COMMITTEE:

DR. S. M. SANI:	**Chairman**
DR. A. M. S. TEKANYI:	**Member**

A DISSERTATION SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES, AHMADU BELLO UNIVERSITY, ZARIA IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF A MASTER OF SCIENCE DEGREEIN TELECOMMUNICATIONS ENGINEERING

APRIL, 2017

DECLARATION

IAduma Fred OKOH, declare that the work in this Dissertation titled **‘‘Developmentof Measurement-Based Admission Control Algorithm for Edge Carrier Ethernet Network to Reduce Packet Loss and Improve Bandwidth Utilization”** has been carried by me in the Department ofElectrical and Computer Engineering, Ahmadu Bello University, Zaria. The information derived from the literature has been duly acknowledged in the text and a list of references provided. No part of this dissertation was previously presented for another degree at this or any other Institution.

Aduma Fred OKOH _____ _____

(Student) SignatureDate

CERTIFICATION

This dissertation entitled **''DEVELOPMENT OF MEASUREMENT-BASED ADMISSION CONTROL ALGORITHM FOR EDGE CARRIER ETHERNET NETWORK TO REDUCE PACKET LOSS AND IMPROVE BANDWIDTH UTILIZATION"**byADUMA FRED**OKOH**meets the regulations governing the award of degree of Master of Science (M.Sc) in Telecommunications Engineering of the Ahmadu Bello University, Zaria and is approved for its contribution to knowledge and literary presentation.

Dr. S. M. Sani	_____	_____
Chairman, Supervisory Committee	Signature	Date
Dr. A. M. S. Tekanyi	_____	_____
Member, Supervisory Committee	Signature	Date
Dr. Y. Jibril	_____	_____
Head of Department	Signature	Date
Prof. S. Z. Abubakar	_____	_____
Dean School of Postgraduate Studies	Signature	Date

DEDICATION

This work is dedicated to Almighty God, my wife (Mrs. Helen Isiaka-Okoh) and my two lovely children, Fortress and Jotham.

ACKNOWLEDGEMENT

My profound gratitude goes to the Almighty God for his grace, mercy, and loving kindness for the ability he gave me to conclude this programme successfully.

My appreciation goes to my wife, Mrs. Helen Isiaka-Okoh for her encouragement and concern throughout the programme. Many thanks to my uncle, ASP Moses Aduma (Rtd) for the training and upbringing. My thanks also goes to my beloved late parents for giving me a good foundation. May their gentle soul rest in perfect peace, Amen.

My whole-hearted appreciation goes to my supervisor, Dr. S. M Sani for his role in the completion of this dissertation. I truly appreciate and value your esteemed guidance and encouragement from the beginning to the end of this dissertation. May almighty God bless you for your immeasurable effort. My thanks also go to my co-supervisor, Dr. A. M. S Tekanyi for his patience, guidance, and valuable contribution towards the success of this work. May Almighty God rewards you. I sincerely appreciate all the lecturers in the Department who contributed to the success of this dissertation through their constructive criticism. Your contributions are well appreciated.

My esteemed appreciation goes to my friend, Isaac Egbunu who assisted me in carrying out the simulation. My deepest appreciation also goes to my special and beloved friends: Dalhatu Francis Clifford, Maryam Imam, George Azever, Chioma Onwuekwu, Adetiba Dayo and those whose names could not be mentioned. May God reward all of you for show of true friendship.

TABLE OF CONTENTS

Title Page	i
Declaration	ii
Certification	iii
Dedication	iv
Acknowledgement	v
Table of Content	vi
List of Figures	x
List of Tables	xi
List of Abbreviations	xiii
Abstract	xv

CHAPTER ONE: INTRODUCTION

1.1 Background on Admission Control	1
1.2 Justification of the Research	3
1.3 Problem Statement	3
1.4 Aim and Objectives	3
1.5 Scope of the Research	4

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction	5
2.2 Review of Fundamental Concepts	5
2.2.1 Carrier Ethernet	5

2.2.2 Synchronous Digital Hierarchy 6

2.2.3 SDH Frame Structure 8

2.2.4 Wavelength Division Multiplexing 9

2.2.5 Performance Measurement Methods in an IP-Based Network 11

2.2.5.1 *Intrusive Performance Measurement* 12

2.2.5.2*Non-intrusive Performance Measurement* 12

2.2.6 Equivalent Capacity 14

2.2.7 Validation of Developed Algorithm 25

2.2.7.1 *Welch's T-Test* 25

2.2.7.2 Parametric Welch's T-Test 26

2.2.8 IPLR Percentage Reductions and Bandwidth Utilization 28

2.3 Review of Similar Works 29

CHAPTER THREE: METHODOLOGY

3.1 Introduction 34

3.2 Collection of Measured Data 34

3.3 Development of MBAC Algorithm for WDM and SDH System 37

3.4 Simulation of MBAC Algorithm 39

3.5 Validation 39

CHAPTER FOUR: RESULTS AND DISCUSSIONS

4.1 Introduction 41

4.2 Simulation Results Obtained 41

4.3 Evaluation of the IPLR and Bandwidth Utilization 42

4.3.1 Simulation Results of IPLR for WDM System 42

4.3.2 Simulation Results of Bandwidth Utilization for WDM System 44

4.3.3 Simulation Results of IPLR for SDH System 46

4.3.4 Simulation Results of Bandwidth Utilization for SDH System 47

4.4 IPLR Percentage Reductions and Bandwidth Utilization 49

4.5 Validation of Results of T-Test 54

4.5.1 Results of measured Data 54

4.5.2 Results of Simulated Data 55

4.6 Validation of T-Test Results Obtained 56

4.6.1 Bandwidth Utilization T-Test Results 56

4.6.2 IPLR T-Test Results 58

CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS

5.1 Summary 59

5.2 Conclusions 59

5.3 Significant Contributions 60

5.4 Limitations 61

5.5 Recommendations for further work 61

REFERENCE 62

APPENDICES

Appendix A: Average Weekly Measured Data 66

Appendix B: Program Pseudo code 67

Appendix C: Simulated Results 84

LIST OF FIGURES

Fig.2.1: Carrier Ethernet in Mobile Backhaul 6

Fig.2.2: SDH Layer model 7

Fig.2.3: STM-1 frame Structure 9

Fig.2.4: Wavelength Division Multiplexing Architecture 10

Fig.2.5: CWDM and DWDM 11

Fig. 2.6: Two-State Markov Chain 17

Fig.3.1: Measurement Set up for Collection of Measured Data for SDH System 35

Fig.3.2: Measurement Set up for Collection of Measured Data for WDM System 36

Fig.3.3: Flow Chart for Development of MBAC Algorithm 38

Fig.3.4: Block Diagram for Validation Procedure 40

Fig.4.1: Superimposed Graph of IPLR versus Load for WDM System 43

Fig.4.2: Superimposed Graph of Bandwidth Utilization versus Load for WDM System 45

Fig.4.3: Superimposed Graph of IPLR versus Load for SDH System 47

Fig.4.4: Superimposed Graph of Bandwidth Utilization versus Load for SDH System 48

Fig.4.5: Percentage Reduction of IPLR for WDM System 50

Fig.4.6: Percentage Reduction of IPLR for SDH System 51

Fig.4.7: Percentage Improved Bandwidth Utilization for WDM System 52

Fig.4.8: Percentage Improved Bandwidth Utilization for SDH System 53

LIST OF TABLES

Table 2.1: Transmission Rates of Synchronous Multiplexers 8

Table 2.2: Intrusive and Non-intrusive measurement of IP Performance Parameters 11

Table 4.1: Simulation Result for the WDM System 41

Table 4.2: Simulation Result for the SDH System 42

Table 4.3: IPLR against Load WDM System 43

Table 4.4: Bandwidth Utilization against Load WDM System 44

Table 4.5: IPLR against Load SDH System 46

Table 4.6: Bandwidth Utilization against Load SDH System 48

Table 4.7: Percentage Reduction of IPLR for WDM System 49

Table 4.8: Percentage Reduction of IPLR for SDH System 50

Table 4.9: Percentage Improved Bandwidth of IPLR for WDM System 51

Table 4.10: Percentage Improved Bandwidth of IPLR for SDH System 52

Table 4.11: WDM Measured Data 54

Table 4.12: SDH measured Data 55

Table 4.13: WDM Simulated Data 55

Table 4.14: SDH Simulated Data 56

Table 4.15: Results of T-test on bandwidth Utilization for WDM System-Assumed equal variances 57

Table 4.16: Results of T-test on bandwidth Utilization for SDH System-Assumed equal variances 57

Table 4.17: Results of T-test on IPLR for WDM System-Assumed equal variances 58

Table 4.18: Results of T-test on IPLR for SDH System-Assumed equal variances 58

LIST OF ABBREVIATIONS

ACTO	Acceptance Region Tangent at Origin
ACTP	Acceptance Region Tangent at Peak
ATM	Asynchronous Transfer Mode
CWDM	Coarse Wavelength Division Multiplexing
DCN	Data Communication Network
DEMUX	De-multiplexer
DWDM	Dense Wavelength Division Multiplexing
ENNI	External Network-Network Interface
HB	Hoeffding Bound
IP	Internet Protocol
IPLR	IP Packet Loss Ratio
IPOD	IP Operator Domain
INM	Intelligent Network Management
ITU-T	International Telecommunication Union
LAN	Local Area Network
MBAC	Measurement-Based Admission Control
MIB	Management Base Information
MPs	Management Points
MSOH	Multiplexer Section Overhead
MTN	Mobile Telecommunication Network

MUX	Multiplexer
OSS	Operations Support Systems
OTN	Optical Transport Network
PBAC	Parameter-Based Admission Control
PDH	Plesiochronous Digital Hierarchy
QoS	Quality of Service
RSOH	Regenerator Section Overhead
SDH	Synchronous Digital Hierarchy
SONET	Synchronous Optical Network
SPE	Synchronous Payload Envelope
SPSS	Statistical Package for Social Science
STM	Synchronous Transport Module
UNI	User Network Interface
VC	Virtual Container
WDM	Wavelength Division Multiplexing

ABSTRACT

Admission Control is a vital component for end-to-end Quality of Service (QoS) delivery in a packet flow network. It is more effective when implemented at the edge of the network. The edge of a mobile network is dominated withmultiplexer which either uses Synchronous Digital Hierarchy (SDH) or Wavelength Division Multiplexing (WDM) for transport. The edge of the carrier Ethernet network suffers QoS due to packet loss and underutilization of bandwidth. Measurement-Based Admission Control (MBAC) Algorithm which reduced the IP Packet Loss Ratio (IPLR) of WDM system by 30.36% and SDH system by 25.34% was setto be achieved in this work. The developed Algorithm was also set to improve the bandwidth utilization of WDM system by 30.36% and bandwidth utilization of SDH system by 25.34%. The developed Algorithm was implemented using C-sharp (C#) platform. Based on the simulation results, WDM system IPLR was reduced by 30.36% and SDH system IPLR was reduced by 25.34% as compared to the measured data from the network result before implementation of the MBAC algorithm. WDM system bandwidth utilization was improved by 30.36% and SDH system bandwidth utilization was improved by 25.34% as compared to the baseline result before implementation of the MBAC Algorithm. From the t-test results obtained, the probability levels (P-values) of IPLR and bandwidth utilization for both edge systems were less than 0.05. This implied that the developed Algorithm has better performance in comparison with the measured data from MTN Lagos network.

CHAPTER ONE

INTRODUCTION

1.1 Background on Admission Control

Wireless mobile network services have evolved over time from low bandwidth consumption services to high bandwidth consumption services (Cisco, 2016). This has forced network providers to upgrade their mobile backhaul to IP-based because these high bandwidth consumption services (like streaming media, video, and interactive gaming) are IP-based that require IP-based network. These different services require Quality of Service (QoS) which must be supported by a well-defined IP and Ethernet based network. A well-defined IP- and Ethernet-based network which should satisfy the QoS requirements of these services must exercise admission control at the edge and core of its network to control the amount of traffic injected into its network (Heena *et al*, 2011). Admission control is a set of actions, consecutively executed, to decide whether a customer can be admitted into a network segment or not (Bohnert, 2011). It is a congestion avoidance mechanism. The primary role of admission control in a QoS enabled network is to control the amount of traffic injected into the network so that congestion is avoided (Gao*et al*, 2012).

There are different types of admission control methods and their difference is mainly due to the different types of operation and methods of implementation. Some Admission control methods are based on mathematical calculations and statistics indicators, others on measured traffic (Chromy *et al*, 2013). Some of the implementation methods for admission control are (Bohnert, 2011):

i. Integrated admission control placement

ii. Edge based admission control placement

iii. End point admission control placement.

Integrated Admission Control Placement: This kind of Admission control placement is implemented in each router between the source and destination. A new customer request is admitted hop-by-hop from its origin to its destination.

Edge Based Admission Control Placement: This is Admission control implemented in edge (ingress/egress) routers. It helps in regulating traffic load on access links.

Endpoint Admission Control Placement: This Admission control is implemented in customer equipment. It could be on application software, access routers or switches.

In general, admission control methods are divided into two categories: Parameter-Based Admission Control (PBAC) and Measurement-Based Admission Control (MBAC) (Chromy *et al*, 2013). PBAC method is based only on the current active traffic and on the characteristics of the total active traffic. This method may not be optimal, because it does not rely on a new incoming traffic. MBAC method does not specify how source parameters are obtained, but makes measurements on the network running in real time. This method can achieve higher network utilization because serving a new data flow in the network is common. Common criteria of admission methods are the ability to allocate bandwidth for all actual data flows in cases not exceeding the total capacity limit.

This research work aims to develop a robust Measurement Based Admission Control (MBAC) algorithm which will reduce IP Packet Loss Ratio (IPLR) and improve bandwidth utilization on the edge of carrier Ethernet network.

1.2Justification of the Research

Telecommunication industry is ebbing towards all IP-infrastructure. For successful transformation to all IP networks, network providers must tackle network performance issues like packet loss and bandwidth utilization to have a stable QoS. The study identifies QoS due to packet loss and underutilization of bandwidth at the edge of carrier Ethernet network.MBAC algorithm which reduced the packet loss and improved bandwidth utilization of edge carrier Ethernet network was developed. This resulted in optimized network model that justifies the significantcontributions and improvement of QoS.

1.3Problem Statement

The Management-Based Information (MIB) interface of edge equipment has a lower sub-layer (ifHcInUcastPkts) and higher sub-layer (ifHcOutUcastPkts) (ITU-T M.2301, 2002). The lower sub-layer is the layer at which packets arrive at the interface of the equipment and they are delivered to the higher sub-layer.

In most of the literature reviewed, it was discovered that the bandwidth of the higher sub-layer is less than the packets sent by the lower sub-layer. This causes significant value of IP Packet Loss Ratio (IPLR) and under-utilization of higher sub-layer bandwidth, hence, a major problem this work is set to address.

1.4Aim and Objectives

The aim of this research work is to develop a Measurement Based Admission Control (MBAC) algorithm for edge of carrier Ethernet network to reduce packet loss and improve bandwidth utilization.

The research objectives are centered on the following four items:

i. Collection of measureddata from MTN Lagos network for the period of three months so that it can be studied,analysed, and used for developing the MBAC algorithm.

ii. Developing MBAC algorithm for edge of carrier Ethernet network deployed with WDM and SDHto reduce IPLR and improve bandwidth utilization.

iii. Simulating the developed MBAC algorithm using C-sharp (C#) software platform to obtain results for IPLR and bandwidth utilization.

iv. Validating the MBAC algorithm simulated results by comparing them with the data collected from MTN Lagos network.

1.5 Scope of the Research

The scope of the dissertation is reduction of IPLR and maximization of bandwidth utilization on edge of carrier Ethernet network. The study is based on developing a MBAC algorithm for edge of carrier Ethernet network deployed with SDH and WDM to reduce IPLR and increase bandwidth utilization.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This section reviews pertinent theoretical concepts and similar research works. It is divided into two parts:

(i) The review of fundamental concepts relevant to the research and

(ii) Review of similar research works.

2.2Review of Fundamental Concepts

The effectiveness of Measurement Based Admission Control (MBAC) algorithm is determined by the rules or actions it is set to execute in a network environment. This section reviews the existing theories and governing principles for the transport network where the MBAC algorithm was applied, the performance measurement methods and equivalent capacity principle used for the development of the MBAC algorithm. The validation techniques were also considered in this part.

2.2.1 Carrier Ethernet.

Carrier Ethernet is a ubiquitous, standardized, carrier-class service defined by five attributes which are: standardized services, QoS, service management, reliability, and scalability (Harvey, 2010). These attributes distinguish carrier Ethernet from the regular LAN-based Ethernet. Unlike the Local Area Network (LAN) Ethernet which was designed for use within buildings or between buildings in a campus environment, carrier Ethernet is designed to be used over long distances within cities, across regions and between continents (Raj, 2013). The interface types found in the carrier network are User Network Interface (UNI) and External Network-Network Interface (ENNI). The UNI is the physical interface or port that demarcates the subscriber

(customer edge) from the service provider, while the ENNI is also the physical interface or port that separates service providers. The network that provides the Ethernet services is called the Carrier Ethernet Network (CEN) (Raj, 2013). Fig. 2.1 shows a carrier Ethernet in the mobile backhaul (Harvey, 2010).

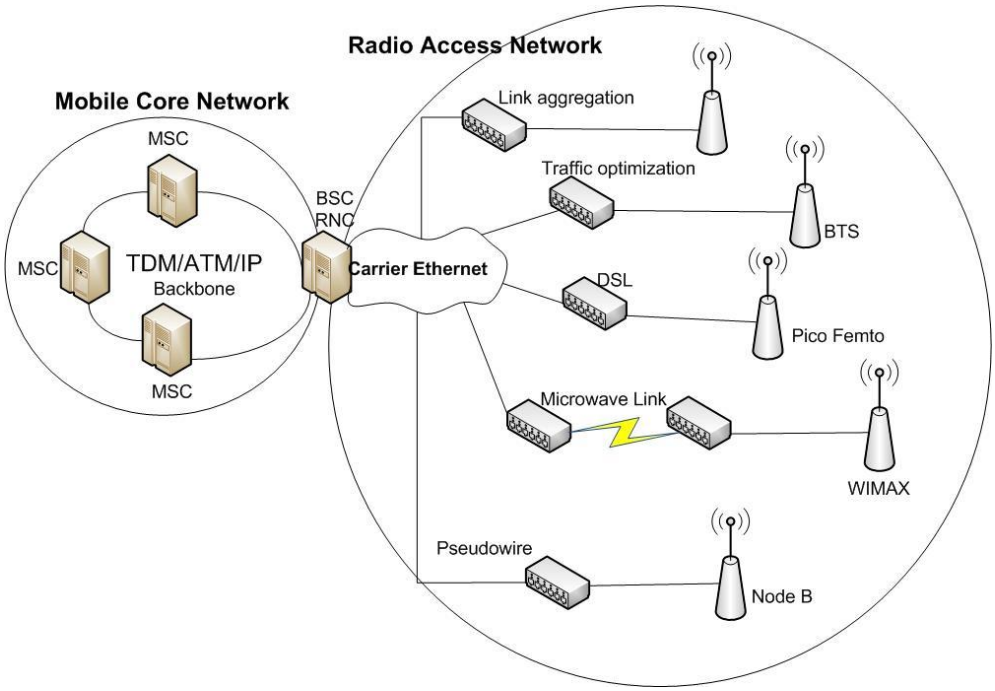


Fig. 2.1: Carrier Ethernet in Mobile Backhaul (Harvey, 2010)

2.2.2 Synchronous Digital Hierarchy

Synchronous Digital Hierarchy (SDH) refers to the group/layers of transmission rates or standards that can transport digital signals/data of different capacities through high bandwidth media like optical fibers or radio waves. It was introduced because of the problems encountered in PDH. It was developed for Europe while its equivalent, Synchronous Optical Network (SONET) was developed for North America. SDH and SONET are the same, but only different in their initial line rate.

The hierarchy starts from the basic SDH rate of 155.52 Mbps, which is greater than the highest transmission rate of the Plesiochronous Digital Hierarchy (PDH) system and goes up to as high as 40Gbps (Jain, 2013). Fig. 2.2 represents the layer model of SDH network.

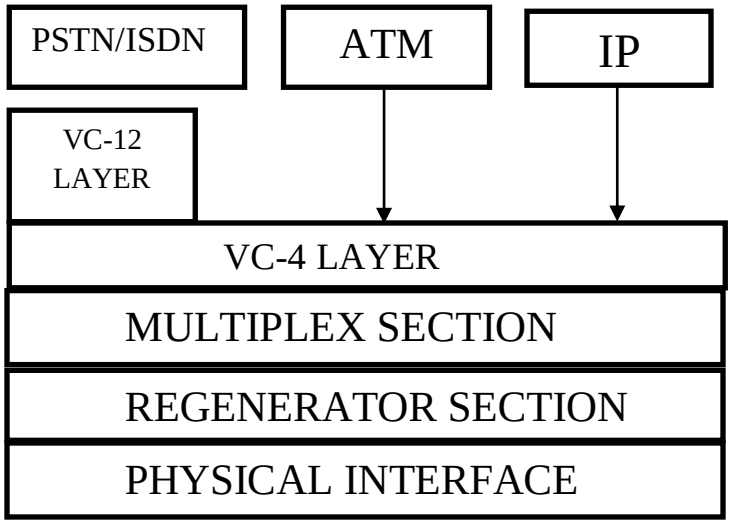


Fig. 2.2: SDH Layer Model (Jain, 2013)

All network technologies and architectures are based on the layered model so that the working of the entire structure in various parts can be abstracted. Hence, the SDH system is modeled in a layered manner to separate the functionality of its various components making understanding of the system easier. The lowest is the physical layer, which represent the transmission medium. This is usually a fiber optic path. Then follows the regeneration section, which is the basic segment of the SDH network. It is the smallest entity that can be managed by the system. Each repeater monitors defects such as the loss of signal, loss of frame, etc. Part of the Regenerator Section Overhead (RSOH) is fully calculated by passing the signal through a regenerator. The next section is the multiplex section, which covers the part of SDH link between multiplexers. The Multiplex Section Overhead (MSOH) is used for this section to detect any error blocks and

defects and generates a special alarm in the forward and backward directions. The two Virtual Container (VC) layers represent a part of the mapping process. Mapping of the signals is done to pack the various incoming tributaries like ATM, IP and PDH into the SDH transport modules. VC-4 mapping is used for packing high capacity signals like ATM signals, whereas VC-12 mapping is for low capacity signals like the PDH (Jain, 2013).

SONET has an initial line rate of 51.84Mbps and SDH level 1 has a bit rate of 155.520Mbps (Vijay, 2011). Table 2.1 shows the transmission rates for these two synchronous multiplexers.

Table 2.1: Transmission Rates of Synchronous Multiplexers (Ian, *et al*, 2010).

SDH	Bit rate (Mbps)	SONET
	52	STS-1
STM-1	155	STS-3
	466	STS-9
STM-4	622	STS-12
	933	STS-18
	1244	STS-24
	18661	STS-36
STM-16	2488	STS-48
STM-64	9953	STS-192

2.2.3 SDH Frame Structure

An SDH frame is called a Synchronous Transport Module (STM). The first level STM frame (STM-1) consists of 2430 bytes, giving an STM-1 transmission rate of 155.520Mbps. It is a 2-dimensional matrix of 9 x 270 bytes and consists of two basic areas (Ian *et al*, 2010): The Transport Overhead (TOH) and the Synchronous Payload Envelope (SPE). The TOH is divided into three parts namely: Regenerator Section Overhead (RSOH), Administrative Unit (AU) pointer, and Multiplexing Section Overhead (MSOH). The RSOH is terminated, that is, disassembled, evaluated and newly generated at each regenerator point, while the MSOH passes the regenerator without being modified and is only terminated at the multiplexers, where the

payload is assembled or disassembled (Jain, 2013). The first 9 columns are made up of the transport overhead while the last 261 columns are for the SPE or payload (Ian *et al*, 2010). The schematic diagram of the STM-1 frame is illustrated in Fig. 2.3

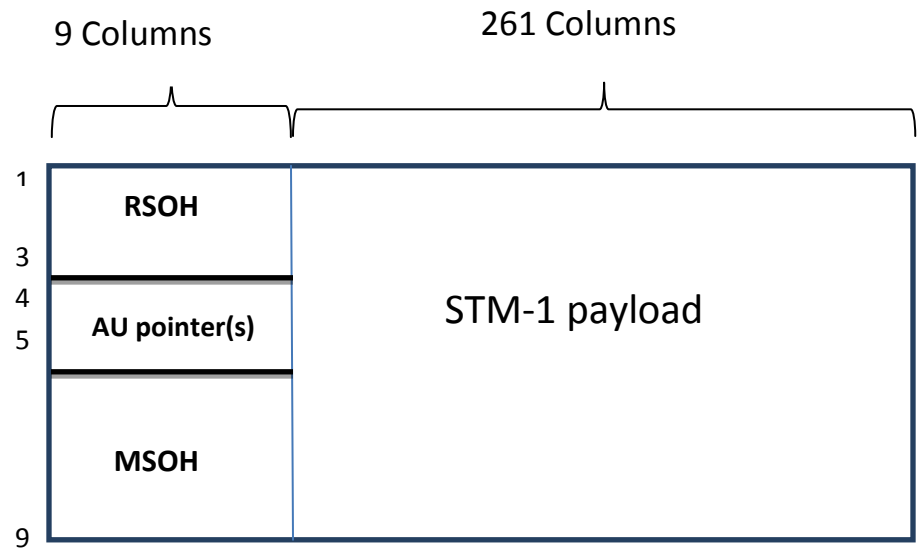


Fig. 2.3: STM-1 Frame Structure (Vijay, 2011)

The use of SDH leads to very flexible network operation (Ian *et al*, 2010). For example, once the frame boundary is found, an individual container can be easily located within the transport frame by its pointer. A container of any size can therefore be dropped from, or added to a passing STM.

2.2.4 Wavelength Division Multiplexing

Wavelength Division Multiplexing (WDM) is the basic technology of optical networking. It is a technique for using a fibre (or optical device) to carry many separate and independent optical channels (Agrawal, 2011). It is a powerful fiber optic transport technology formed by the following three components (Rajiv *et al*, 2010):

i. Photonic layer (WDM),

ii. Digital layer (OTN frame structure), and

iii. Control plane (ASON/WSON).

WDM is the photonic technology that allows the transmission of multiple optical light signals on the same optical fiber cable; it is a multiplexing technique at the optical level (Ajmal & Robert, 2014). There are two key components in a WDM system namely: wavelength multiplexer (MUX) and de-multiplexer (DEMUX). The MUX multiplexes signals at the ingress of the WDM system, while DEMUX de-multiplexes signals at the egress (Agrawal, 2011). Fig. 2.4 illustrates the architecture of the WDM.

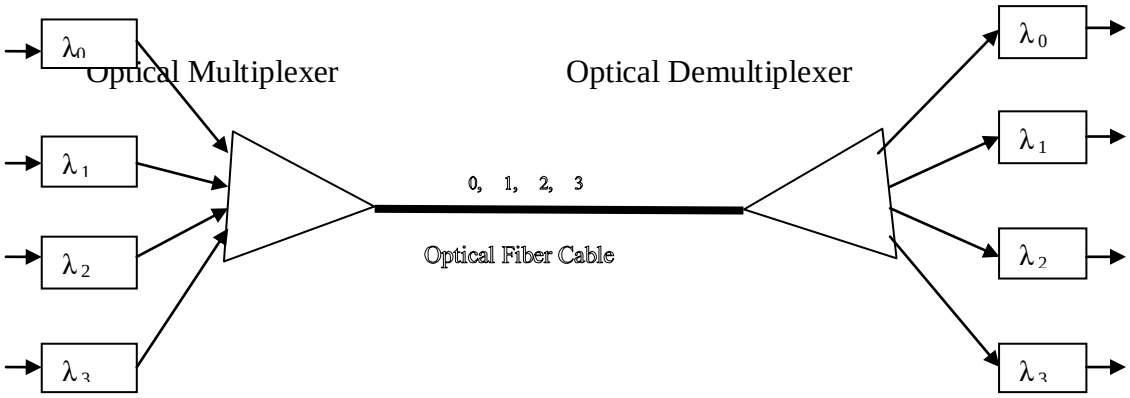


Fig. 2.4: Wavelength Division Multiplexing Architecture (Ajmal & Robert, 2014)

Coarse Wavelength Division Multiplexing (CWDM) and Dense Wavelength Division Multiplexing (DWDM) are the two types of WDM. CWDM uses 16 coarsely spaced wavelengths spanning over very large bandwidth, allowing for the use of low-cost photonic technology, but transmission distances are limited to a few hundred kilometers. Furthermore, CWDM applications are limited to only 8 channels and capacity per channel is also limited to 2.5Gbps. On the other hand, DWDM provides a much narrower wavelength spacing with 160 channels that can be amplified for long distance coverage of more than 300km. Capacity per wavelength for DWDM can be 2.5Gbps or 10Gbps. In the near future, capacity per wavelength of DWDM can be 100Gbps. DWDM is divided into two sub-parts: the C-band and the L-band. The C-band has wavelength range of 1530 – 1565nm, while the L-band has wavelength range of

1565 – 1611nm. Each of these bands allows for a capacity of 80 wavelengths. More wavelengths up to 96 can be transported at a system cost penalty (Ajmal & Robert, 2014). Fig. 2.5 illustrates CWDM and DWDM channels.

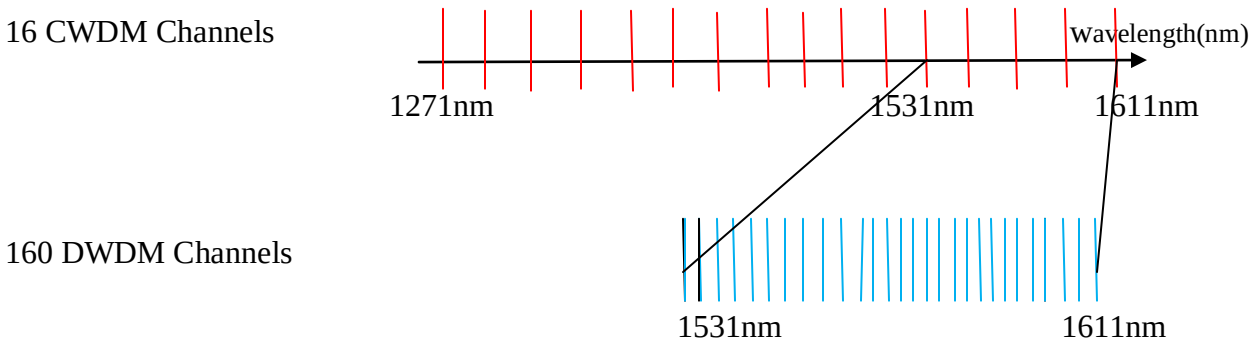


Fig. 2.5 CWDM and DWDM (Ajmal & Robert, 2014)

2.2.5 Performance Measurement Methods in an IP-Based Network.

There are two basic approaches to performance measurement (ITU-T M.2301, 2002). These are intrusive and non-intrusive which equate to the terms active and passive used by the IETF. Some performance parameters can be measured only intrusively, others only non-intrusively and some both intrusively and non-intrusively. Table 2.2 classifies IP performance parameters into intrusive and non-intrusive (ITU-T M.2301, 2002).

Table 2.2: Intrusive and Non-intrusive measurement of IP performance parameters

Parameter	Intrusive	Non-intrusive
IPTD	✓	
IPDV	✓	
IPER	✓	✓
IPLR	✓	✓
IPDR		✓

2.2.5.1*Intrusive performance Measurement*

Intrusive performance measurements are made by inserting test packets interleaved with the normal traffic flows between two Measurement Points (MPs). This kind of measurement allows more detailed investigation of specific performance parameters like one-way delay using time stamped packets, effect of packet size, and number of packets on performance.

It should be noted that intrusive performance measurement causes additional traffic through the network, so care must be taken to ensure that the use of this test does not cause congestion and the subsequent loss of customer's packets. It is also important to note that the test is not carried out when customer traffic is so low to make the results of the test invalid. Intrusive measurement techniques are used when it is difficult or impossible to use non-intrusive measurements.

2.2.5.2*Non-intrusive Performance Measurement*

Non-intrusive performance measurement is carried out when performance parameters are obtained from the network on real time. This kind of measurement has the advantages of minimising impact on customer's traffic and testing every route through the network. Problems on links or routers can also be quickly identified. It should be noted, however, that non-intrusive measurements can realistically be done only within one IP Operator Domain (IPOD) since it may be difficult or undesirable for one operator to access routers in another's IPOD. Non-intrusive performance measurements are done using Management Information Base (MIB). The parameters that make up MIB are:

ifHcInUcastPkts: This is used to define the number of packets delivered by this sub-layer to ahigher sub-layer,that were not addressed to a multicastor broadcast address at this sub-layer. This object is a64-bit version of ifInUcastPkts (McCloghrie *et al*, 2000).

ifHcOutUcastPkts: This define the total number of packets that higher-level protocols requested to be transmitted and were not addressed to amulticast or broadcast address at this sub-layer, includingthose that were discarded or not sent. This object is a64-bit version of ifOutUcastPkts (McCloghrie *et al*, 2000).

ifInNUcastPkts: This is for number of packets delivered by this sub-layer to ahigher sub-layer, which were addressed to a multicast orbroadcast address at this sub-layer (McCloghrie *et al*, 2000).

ifOutNUcastPkts: This parameter defines the total number of packetsthe higher-level protocols requested to be transmitted which were also addressed to amulticast or broadcast address at this sub-layer, includingthose that were discarded or not sent (McCloghrie *et al*, 2000).

ifInDiscards: This term defines the number of inbound packets which were chosen to bediscarded even though no errors had been detected to prevent their being delivered to a higher-layer protocol. Onepossible reason for discarding such a packet could be tofree up buffer space (McCloghrie *et al*, 2000).

ifOutDiscard: This is for number of outbound packets which were chosen to bediscarded even though no errors had been detected to preventtheir being transmitted. One possible reason for discardingsuch a packet could be to free up buffer space (McCloghrie *et al*, 2000).

ifInErrors: This is for packet-oriented interfaces and it defines the number of inboundpackets that contained errors which prevent them from beingdeliver to a higher-layer protocol (McCloghrie *et al*, 2000).

ifOutError: This is for the number of outboundpackets that could not be transmitted because of errors (McCloghrie *et al*, 2000).

ifInUnknownProtos: This could be defined for packet-oriented interfaces and character-oriented or fixed-length interfaces.For packet-oriented interfaces, this is the number of packetsreceived via the interface which were discarded because ofan unknown or unsupported protocol whilefor character-orientedor fixed-length interfaces it is the number of transmission units received viathe interface which were discarded because of an unknown orunsupported protocol. For any interface that does notsupport protocol multiplexing, this counter will always be 0 (McCloghrie *et al*, 2000).

The recommended performance parameters for non-intrusive measurement which are IPER, IPLR and IPDR can be computed from equation (2.1) to (2.5) (ITU-T M.2301, 2002)

IPER = (ifInError + ifInUnknownProtos)/ifInTotal (2.1)

where:

ifInTotal = ifInUcastPkts + ifInNUcastPkts + ifInDiscards + ifInErrors . . .

+ ifInUnknownProtos (2.2)

IPLR = (ifInUcastPkts + ifInNUcastPkts)/ifOutTotal (2.3)

ifOutTotal = ifOutUcastpkts + ifOutNUcastPkts (2.4)

IPDR = (ifInDiscards + ifOutDiscards)/ifOutTotal (2.5)

2.2.6 Equivalent Capacity

The equivalent capacity of a set of connections multiplexed on a link is defined as the amount of bandwidth required to achieve a desired Quality of Service (QoS) (Guerin *et al*, 1991). It is a function of individual connection characteristics and available network resources such as buffers. The equivalent capacity therefore determines the bandwidth allocation levels on network links. It provides a unified metric for link loads which can then be used by Network control functions such as routing and congestion control. The equivalent bandwidth is computed from the

combination of two different approaches: one based on a fluid-flow model and the other on an approximation of the secondary bit rate distribution (Guerin *et al*, 1991).

Fluid-Flow Approximation: In this kind of model, the bit rate generated by several multiplexed connections is represented as a continuous flow of bit rate with intensity varying per the state of an underlying continuous-time Markov chain. This Markov chain is obtained from the superposition of the sources associated with each connection. For example, in the case of multiplexing N identical two-state Markov sources (each capable of a peak bit rate of R_{peak}), the intensity of the aggregate bit rate is $jR_{peak}$when j sources are active. The aggregate bit rate is offered to a buffer which is emptied at a constant rate c. The interest is to determine the smallest value, $\hat{C}$ of c for a given buffer size, $\varkappa$. The value $\hat{C}$ is the equivalent capacity of the multiplexed connections. To determine $\hat{C}$ ensure that a buffer overflow probability (QoS) smaller than ϵ is obtained. This can be achieved by first deriving an expression for a given distribution of the buffer content as a function of the connections characteristics and the service rate. The expression derived will then be inverted to determine the value of the service rate which ensures an overflow probability smaller than ϵ for the available buffer size $\varkappa$.

Markov Process: A Markov process is a random process for which the future (the next step) depends only on the present state and has no memory of how the present state was reached (Richard, 2011). To properly define Markov chain, the following statistical definitions must be considered (Richard, 2011):

i. If I is a countable set, $\{i, j, k, \ldots\}$ then $i \epsilon I$is called a stateand I is called the state-space.

ii. In a probability space (Ω, f, P), where Ω is a set of outcomes, f is a set of subset of Ω and $A \epsilon f$, then P(A) is the probability of A.

iii. For a given sequence of random variables X_0, X_1, . . (taking values in *I*) whose joint distribution is determined by simple rules, then the random variable X with values in *I* is a function $X : \Omega \rightarrow I$

iv. A row vector:

$$\lambda = (\lambda_i : i \epsilon I) \tag{2.6}$$

is called a measured if,

$$\lambda_i \geq 0 \tag{2.7}$$

for all *i*.

v. If,

$$\sum_i \lambda_i = 1 \tag{2.8}$$

then it is a distribution or probability measure.

vi. Transition matrix,*P* is defined as:

$$P = (p_{ij} : i, j \epsilon I) \tag{2.9}$$

with

$$p_{ij} > 0 \tag{2.10}$$

for all *i, j*.

vii. The transition matrix is stochastic matrix if,

$$p_{ij} \geq 0 \tag{2.11}$$

For all $i, j \epsilon I$and

$$\sum_{j \epsilon I} p_{ij} = 1 \tag{2.12}$$

With reference to the statistical definitions in item (i) – (vii), equation (2.13) is a Markov chain with initial distribution, λ and transitional matrix, *P*and given as:

$$(X_n)_n \geq 0 \tag{2.13}$$

if for all

$$n \geq 0 \tag{2.14}$$

$i_0, \ldots i_{n+1} \epsilon\ I,$

$$P(X_0 = i_0) = \lambda_{i0}; \tag{2.15}$$

$$P(X_{n+1} = i_{n+1} | X_0 = i_0, \ldots, X_n = i_n) = P(X_{n+1} = i_{n+1} | X_n = i_n) \ldots$$

$$\ldots P_{inin+1} \tag{2.16}$$

Two-State Markov Chain: Fig. 2.6 illustrates a two-state Markov chain

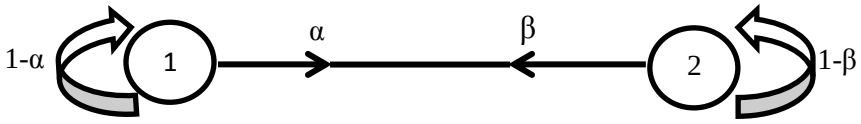


Fig. 2.6: Two-State Markov Chain (Richard, 2011)

1 and 2 are the state one and state two, respectively. The probability of moving from state 1 to state 2 is α, probability of moving from state 2 to state 1 is β, probability of staying at state 1 is 1-α and the probability of staying in state 2 is 1-β.

The transition probability matrix, *P* of a two-state Markov chain is given as (Richard, 2011):

$$P = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix} \tag{2.17}$$

Single source: Using the notations of equation (2.17), a two-state Markov source can be said to be characterized by its peak rate $R_{peak,}$ utilization ρ, and mean burst period b. The distribution of the buffer contents, when such a source is feeding a buffer served by a constant rate server, can be derived using standard techniques for either infinite or finite buffer systems. From this distribution, it is then possible to determine the service rate, or equivalent capacity, $\hat{C}$ needed to achieve a given QoS. The QoS is the buffer overflow probability when the source is active and

transmitting. Assuming a finite buffer of size $\varkappa$, the equation satisfied by the equivalent capacity for an overflow probability of ϵ (the desired QoS) is then easily found to be of the form (Guerin *et al*, 1991):

$$\epsilon = \beta \times \exp\left(-\frac{\varkappa(\mathrm{c}-\rho R_{peak})}{b(1-\rho)(R_{peak}-\mathrm{c})\mathrm{c}}\right) \tag{2.18}$$

where,

$$\beta = \frac{(c-\rho R_{peak})+\varepsilon\rho(R_{peak}-c)}{(1-\rho)\mathrm{c}} \tag{2.19}$$

The equivalent capacity for a single source can then be obtained by solving for c in equation (2.19). It should be noted that, even for the simple single-source case, no explicit expression is available for the equivalent capacity and equation (2.19) must be solved numerically. A natural simplification is available as the term β can be shown to be typically close to 1. Approximating β to be 1 in equation (2.19) provides an explicit upper bound for $\hat{\mathrm{C}}$. The equivalent capacity associated with a single connection in isolation is then taken to be (Guerin *et al*, 1991):

$$\hat{\mathrm{C}} \approx \frac{\alpha b(1-\rho)R_{peak}-\varkappa\sqrt{[\alpha b(1-\rho)R_{peak}-\varkappa]^2+4\varkappa\alpha b\rho(1-\rho)R_{peak}}}{2\alpha b(1-\rho)} \tag{2.20}$$

where,

$$\alpha = \ln(1/\epsilon) \tag{2.21}$$

In the case of a continuous bit stream connection, $\rho = 1$ and $b = \infty$. Taking limits in equation (2.36) yields:

$$\hat{\mathrm{C}} = R_{peak}\,. \tag{2.22}$$

Multiple Sources: In the case of multiple superposed sources, the distribution of the content of a buffer fed by the aggregate bit rate and served by a constant rate server can also be determined. The approach is more complex than for single sources and essentially numerical despite the

existence of powerful simplification techniques (Mitra, 1988). In general, when the input bit rate is characterised by an N-state Markov chain, the distribution of the buffer contents is of the form (Guerin *et al*, 1991):

$$F(\varkappa) = \sum_{i=1}^{N} a_i \Phi_i \, e^{z_i \varkappa} \tag{2.23}$$

where:

z_i and Φ_i are generalized eigenvalues and eigenvectors

a_i's are coefficients

$F(\varkappa)$ is the distribution

z_i and Φ_i are associated with the solution of the differential equation satisfied by the stationary probabilities of the system and a_i's are determined from boundary conditions. The distribution, $F(\varkappa)$ is completely determined from the values of the associated eigenvalues, eigenvectors, and corresponding coefficients.There are typically no explicit expressions for these quantities, which must then be determined numerically. The equivalent capacity corresponding to a set of multiplexed sources can then be obtained using iterative numerical techniques, where at each iteration a new solution to equation (2.23) must be computed. Such a procedure, although exact, is unfortunately not compatible with a dynamic and real-time environment. Approximations must then be developed.

A first natural step is to consider an asymptotic approximation. The system is assumed to have an infinite buffer with a probability of a buffer overflow approximately equal to the probability that the buffer content exceeds the actual buffer size. For large buffers, this quantity can then be further approximated by only considering the contribution of the term corresponding to the largest negative or dominant eigenvalue in equation (2.23). Under such assumptions, the buffer overflow probability is known to be of the form (Kosten, 1986):

$$G(\varkappa) \cong \beta e^{z_0 \varkappa} \tag{2.24}$$

where,z_0 is the largest negative eigenvalue and β is a constant term independent of $\varkappa$.

z_0 can be obtained from relatively simple expressions, while the determination of the constant β is typically more difficult (Kosten, 1986). This can be illustrated with the example of *N* identical two-state Markov sources for which explicit expressions are available. The example will further provide indications to further simplify equation (2.24) to obtain a simple and explicit approximate expression for the equivalent capacity of multiple connections. In the case of *N* identical two-state Markov sources, the following relationship is obtained (Guerin *et al*, 1991):

$$G(\varkappa) = \left(\frac{N\rho R_{peak}}{C}\right)^N \left[\prod_{i=1}^{N-\left\lfloor \frac{C}{R_{peak}} \right\rfloor - 1} \frac{z_i}{z_i + z_0}\right] . \exp\left(-\frac{N(\mathrm{c} - N\rho R_{peak})}{b(1-\rho)(NR_{peak} - \mathrm{c})C\varkappa}\right) \tag{2.25}$$

where:

z_i are all negative eigenvalues and the largest one, z_0 is explicitly given in the exponential term.

C is capacity allocated to the *N* connections, and $\left\lfloor \frac{C}{R_{peak}} \right\rfloor$ is the largest integer smaller than C/R_{peak} .

The expression for β in equation (2.25) is quite complex and involves not only the largest eigenvalue but all the other negative eigenvalues as well. In addition, since the value of the allocated capacity appears as a variable determining the number of coefficients in the product term present in β, simple approximations for β seem difficult to derive. In contrast, the expression for z_0 is relatively a simple function of the unknown allocated capacity. A natural

approach is then to approximate β, at least over a range of connection characteristics by a constant providing a reasonable estimate.

A possible choice, consistent with the single-source approximation is to again approximate β by 1. Although the actual value of β is unfortunately not always close to 1, but there exists a range of connection characteristics for which this assumption is reasonable. For example, in the case of equation (2.25), β is the product of two factors. The first one is the *N*th power of a term smaller than 1, and the second is itself the product of *N* terms all greater than 1. One can then expect that at least for some range, these two terms will compensate for each other and yield a product whose magnitude is reasonably close to 1. It can be verified numerically that this is approximately the case when either the number of connections is small or the actual total equivalent capacity is reasonably close to the overall mean rate. In other cases, approximating β to be 1 can, however, result in a gross overestimation of the equivalent capacity. If β is approximated to 1 for fluid- flow model then a second approximation based on the distribution of the stationary bit rate must be developed.

Setting $\beta = 1$ in equation (2.25), a simple explicit expression can be obtained for the equivalent capacity as a function of individual source characteristics. Letting $\beta = 1$ in equation (2.25) requires a buffer overflow probability of ϵ to be:

$$\epsilon = e^{z_0 \varkappa} \rightarrow z_0 = -\frac{\alpha}{\varkappa} \tag{2.26}$$

where,

$$\alpha = \ln(1/\epsilon) \tag{2.27}$$

In the case of the two-state Markov sources (not necessarily identical), there exists a simple relation between the dominant eigenvalue z_0 and the allocated capacity (Kosten, 1986), which is given as:

$$C = \frac{\sum_{i-1}^{N}\left[\left(\lambda_i + \square_i + R_{peak}^{(i)}\ z_0\right) - \sqrt{\left(\lambda_i + \square_i + R_{peak}^{(i)}\ z_0\right)^2 - 4\lambda_i R_{peak}^{(i)}\ z_0}\right]}{2z_0} \quad (2.28)$$

where, $1/\lambda_i$ and $1/\square_i$ are the respective mean idle and burst periods of source *i* and $R_{peak}^{(i)}$ is the corresponding peak rate.

Combining equation (2.26) and equation (2.28) yields the value of the equivalent capacity, $\hat{C}_{(F)}$ given by the flow approximation for *N* multiplexed sources (Guerin *et al*, 1991) as:

$$\hat{C}_{(F)} = \sum_{i=1}^{N} \hat{c}_i \quad (2.29)$$

where, $\hat{c}_i$ are determined from equation (2.20).

The major advantages of equation (2.29) are its computational simplicity and its explicit dependency on source parameters (Guerin *et al*, 1991). However, although the linearity of equation (2.29) certainly simplifies the accountability of how bandwidth is allocated to connections, it also clearly identifies the limitations of the expression. The linearity explicitly indicates that the simplifying assumption $\beta = 1$ amounts to ignoring the effect of statistical multiplexing. Unless the equivalent capacities of individual connections are themselves close to their mean bit rates, their sum is typically an overestimation of their equivalent capacity. Another approximation is therefore needed to accurately determine the required bandwidth allocation for cases in which statistical multiplexing is significant, hence, the need for stationary approximation.

Stationary Approximation: When several connections with relatively long burst periods are multiplexed, a reasonably accurate estimate of the required bandwidth can be obtained from the stationary bit rate distribution. The equivalent capacity, $\hat{C}_{(s)}$ can be selected to ensure that the aggregate stationary bit rate exceed $\hat{C}_{(s)}$ only with a probability smaller than ϵ, the desired buffer overflow probability. This clearly ensures a buffer overflow probability below ϵ, but is often a

substantial overestimate of the actual bandwidth required as it ignores the "smoothing" effect of the buffer, that is, the buffer allows the input rate to exceed the output rate for "short" periods. In the case of connections with long burst periods, it can be argued that because the aggregate bit rate remains at the same value for relatively long periods, overload situations (when present) are likely to last sufficiently long to result in buffer overflow. It is then reasonable to allocate enough bandwidth to make the probability of an overload condition equal to the desired buffer overflow probability. For a set of multiplexed connections, the value of the equivalent capacity which satisfies such a requirement on the aggregate stationary bit rate is the smallest value of $\hat{C}_{(s)}$ which satisfies (Guerin *et al*, 1991):

$$\Pr(B > \hat{C}_{(s)}) \leq \epsilon \tag{2.30}$$

where,B is the aggregate bit rate and ϵ the desired buffer overflow probability.

The distribution of Bcan be determined from the stationary distribution of the underlying Markov chain formed by the superposition of sources. In the case of identical two-state Markov sources, the probability, P_k that k out of N sources is active is given by a binomial distribution (Guerin *et al*, 1991):

$$P_k = \binom{N}{k}\rho^k(1-\rho)^{N-k} \tag{2.31}$$

The value of $\hat{C}_{(s)}$ is then obtained by computing the smallest integer k' such that:

$$\sum_{k=k'+1}^{N} P_k \leq \epsilon \tag{2.32}$$

The stationary approximation then gives:

$$\hat{C}_{(s)} = k' R_{peak} \tag{2.33}$$

The determination of k' and $\hat{C}_{(s)}$ is possible but in general, the computation of $\hat{C}_{(s)}$ for nonhomogeneous sources can be difficult (Guerin *et al*, 1991). However, in most cases where the

effect of statistical multiplexing is of significance, the distribution of the stationary bit rate can be rather accurately approximated by a Gaussian distribution (Erfani & Malek, 1989).

The assumption of a Gaussian distribution allows the use of standard approximations to estimate the tail of bit rate distribution. In this case, it is to determine the value $\hat{C}_{(s)}$ so that the cumulative tail probability beyond $\hat{C}_{(s)}$ does not exceed ϵ. The value of $\hat{C}_{(s)}$ is then obtained from approximations for the inverse of the Gaussian distribution. A good approximation is given by (Guerin *et al*, 1991) as:

$$\hat{C}_{(s)} \approx m + \alpha' \sigma \tag{2.34}$$

where,

$$\alpha' = \sqrt{-2\ln(\epsilon) - \ln(2\pi)} \tag{2.35}$$

m is the mean aggregate bit rate:

$$m = \sum_{i=1}^{N} m_i \tag{2.36}$$

σ is the standard deviation of the aggregate bit rate:

$$\sigma^2 = \sum_{i=1}^{N} \sigma^2{}_i \tag{2.37}$$

Combining equation (2.20), (2.29), and (2.34) into a single expression gives the equivalent capacity of a set of connections (Guerin *et a*l, 1991). The equivalent capacity, $\hat{C}$ is taken to be the minimum of $\hat{C}_{(F)}$ and $\hat{C}_{(s)}$. That is:

$$\hat{C} = \min\{m + \alpha' \sigma, \sum_{i=1}^{N} \hat{c}_i\} \tag{2.38}$$

where N is the number of multiplexed connections.

The different quantities involved are defined in equation (2.20), (2.29) and (2.34), respectively.

2.2.7Validationof Developed Algorithm

Validation deals with the assessment of the comparison between 'sufficiently accurate' computational results from the simulation and the actual/hypothetical data from the system (Martis, 2006). In other words, it is the process of determining the degree to which a model is an accurate representation of the real-world from the perspective of the intended uses of the model (DoD 2002). This research work adopted the use of statistical validation using Welch's t-test technique and validation of the developed algorithm.

2.2.7.1 *Welch's t-test*

A two-sample test is statistical comparisons between two sets of data. There are several varieties of two-sample tests that exist which are correlation coefficient, Z-test, etc. In this research work, the parametric Welch's t-test is used to compare statistically difference between the two independent means of measured data and simulated result because their results are observed in the two samples. Also, Welch's t-test is a robust statistical procedure that usually provides accurate results even if the assumption of normality is partially violated (Uwaechia *et al*,2013).

The unified guidance (EPA, 2009) describes two versions of the t-test as, the pooled variance Student's t-test and an adaptation of Student's t-test known as Welch's t-test. The pooled variance Student's t-test assumes that the variances are equal in both sample data (in the case of independent samples from normal samples with common variances). The Student's *t*-test was invented precisely to determine when an observed sample difference should be considered significant, especially when the sizes of the samples tend to be small.

When variances of the two-sample data are unequal, Welch's t-test is used. The Welch's t-test is mostly used compared to that of the Student's t-test because it is an approximate to the Behrens-Fisher problem (EPA, 2009). The Behrens–Fisher is the problem of interval estimation and

hypothesis testing concerning the difference between the means of two normally distributed populations when the variances of the populations are not assumed to be equal, based on two independent samples (EPA, 2009).

2.2.7.2*Parametric Welch's t-test*

Some relevant terms applicable to the statistical Welch's *t*-test that are useful inachieving the stated objectives are:

(i) **Levene's Test:** Levene's test is an inferential statistic used to test if the variances are equal ornot equal in both sample data. In Statistical Package for the Social science (SPSS) using the t-test result, power value (P-value) appears as significant value (sig-t). If the P-value is less than thesignificance level (α) of 0.05 then it is assumed that the variances are unequal,otherwise, it is assumed equal variance.It also tests the null hypothesis that the population variances are equal. Thus, when the null hypothesis of equal variances is rejected, it is concluded that there is a difference between the variancesin the population.

(ii) **Means and Variances of two Samples:** $\bar{X}_1$ and $\bar{X}_2$ are the means of the two samples of the obtained data from realGSM network and simulation data from the developed model, $S_1^2$and $S_2^2$are the variances of the two samples data, n_1 and n_2 are the sample sizes of the data and k is the number of degrees of freedom. Then, the average mean $(\overline{X})$ and variance $(s^2)$of the two samples are given by (Dass, 2015) as:

$$\bar{X} = \frac{1}{n}(x_1 + x_2 + x_3 + \ldots\ldots + x_n) = \frac{1}{n}\sum_i^n x_i \qquad (2.39)$$

$$S^2 = \frac{1}{(n-1)}\{(x_1 - \bar{X})^2 + (x_2 - \bar{X})^2 + (x_3 - \bar{X})^2 + \ldots. + (x_n - \bar{X})^2\} = \frac{1}{(n-1)}\sum_i^n (x_1 - \bar{X})^2 \qquad (2.40)$$

$$k = (n_1 + n_2) - 2 \quad , if\ n_1 = n_2 \tag{2.41}$$

(iii) **T-value:** Welch's t-value is defined (EPA, 2009) as follows:

$$\text{T-value} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{2.42}$$

(iv) **K-Degrees of Freedom:** This is the number of values which are involved in the final calculation of a statistic that is expected to vary. The number of degrees of freedom (k) is computed using the equation (EPA, 2009) as:

$$k = \frac{(\frac{s_1^2}{n_1} + \frac{s_1^2}{n_2})^2}{(\frac{s_1^4}{n_1^2(n_1 - 1)} + \frac{s_2^4}{n_2^2(n_2 - 1)}} \tag{2.43}$$

(v) **One-tailed test:** It uses an alternate hypothesis that states that

H_O: μ1> μ2(the measured data mean is greater than simulated data mean).

$H_{A:}$μ1< μ2 (the measured data mean is less than simulated data mean).

where,μ1 and μ2are samples means of the measured and simulated data (Antonio, 2010).

(vi) **Two-tailed test:** It uses null hypothesis that states that

HO: μ1= μ2 (the population means are equal).

HA: μ1≠μ2 (the populations means are not equal) (Antonio, 2010).

(vii) **Null hypothesis $H_{O:}$** If μ1= μ2, fail to accept. This means that there is no significant difference between the two sample data (Antonio, 2010).

(viii) **Alternate hypothesis $H_{A:}$** Ifμ1≠μ2, then reset or fail to reject. This means there is significant difference between the two sample data (Antonio, 2010).

(ix) **Significance Level:** The significance level $\alpha$is set to 0.05, if $\alpha < 0.05$, reject H_O and $\alpha \geq 0.05$ fail to reject H_O. SPSS uses significant value to indicate significance or probability level (p) of all inferential statistics which could be any value from 0 to 1.

In this research, the following decision criteria are made for two-tailed test as (Antonio, 2010):

i. If significant p $<\alpha$, then reject H_O. This means there is significance difference between the existing model and developed model. Hence, developed model is of higher performance.

ii. If significant p $\geq\alpha$, fail to reject H_O. This means there is no significance difference between theexisting model and developed model.

2.8 IPLRPercentage Reductions and Bandwidth Utilization

The percentage reduction of IPLR is determine by

$$\%\, reduction\ of\ IPLR = \frac{x-y}{x} * 100 \tag{2.44}$$

where,

x = IPLR when algorithm is not applied

$y =$ IPLR when algorithm is applied

The percentage improved bandwidth utilization is determined by

$$\%\, improved\ Bandwidth = \frac{N-M}{N} * 100 \tag{2.45}$$

where,

N = computed equivalent bandwidth (computed ifHcOutUcastPkts)

M = system recorded bandwidth (ifHcOutUcastPkts)

2.3Review of Similar Works

This sub-section presents a review of some relevant works carried out and published by other researchers. This gives an indication of the extent of the work in this area of research and the techniques or mechanisms used to resolve the same problem.

Brewer & Ayyagari, (2010) presented work on comparison and analysis of measurement and parameter based admission control methods for Quality of Service (QoS) provision. They implemented versions of both MBAC and PBAC in their network QoS management framework and evaluated MBAC implementation against the more traditional PBAC method in order to compare the efficiency of network utilization between the two methods. They found that for bursty traffic patterns, the MBAC method allowed for better network utilization and for more applications to access the network at the same time than the PBAC method allowed. In their work, edge routers were configured to classify specific packets/flows while core routers applied QoS policy to these flows. This method affected QoS in terms of delay. Classifying packets is a Diffserv method and not MBAC method. More so, MBAC method is strictly based on whether packet/flow should be admitted or rejected. This was not clearly applied in their work.

Alwakeel and Alotaibi, (2011)presented an article on end-to-end measurement based admission control VOIP protocol with loss policy, where they proposed the EMBAC-L scheme for QoS in VOIP networks. They evaluated the performance of the scheme by using different distribution functions in order to find the admission scheme parameters that could provide the required acceptable QoS. They showed that the performance of EMBAC-L scheme provided an enhancement to the original EMBAC scheme. In this work, there was no provision for decision to be taken by the scheme on packets that were rejected.

Ammar*et al*.(2011) submitted work on evaluation and comparison of MBAC solutions, where three MBAC solutions of measured sum, equivalent capacity, and aggregate traffic envelopes were evaluated and compared in terms of packet loss rate and packet queuing delay. The researchers observed all investigated solutions were able to fulfil the target loss rate. On the other hand, only the aggregate traffic envelope was always able to meet the target queuing delay. The parameter set for packet loss rate in this work was not based on any known packet loss rate standard.

Auge *et al*.(2011) presented a paper on measurement-based admission control for flow-aware implicit service differentiation. They proposed MBAC algorithm and evaluated its performance by simulation under different stationary traffic mixes and in a flash crowd scenario. They showed that the algorithm offered a satisfactory compromise between flow performance and link utilization. There was need to clearly state the nature of traffic/flow for which an algorithm was developed but this was not evident in the work.

Heena *et al*.(2011) presented a paper on simulation of measurement based admission control using NS2. They carried out simulation on several network scenarios for video transmission to evaluate the link utilization and adherence to service commitment by comparing results obtained from applying Hoeffding Bound (HB), Acceptance Region Tangent at Peak (ACTP), and Acceptance Region Tangent at Origin (ACTO) methods. Their result showed that HB gave better result when IP network packets were smaller and ACTO and ACTP gave best bandwidth when packet size was more than 1250 bytes. In order to use MBAC methods, there was a need for a measurement process. In this work, measurement process was completely lacking.

Yerima, (2011) implemented and evaluated measurement-based admission control schemes within a converged network QoS management framework. His investigation was undertaken

with real-traffic on Linux-based test-bed, where the admission control schemes were implemented within a policy-based QoS management prototype built with Java programming language. The result showed that MBAC and PBAC schemes inherently achieved better network resource utilization at the expense of higher blocking probability, which translated to reduced service availability. The result further showed that the proposed EWMA-PBAC approach was capable of outperforming the others in terms of network resource utilization but without clarifying service availability. He applied admission control both on the core and edge router. Applying admission control on core and edge router could affect QoS in terms of delay and jitter because traffic have to undergone scrutiny at two different points before it would be allowed to pass or not but applying the admission control on the edge router would reduce the time it takes the traffic to undergone scrutiny before it could pass or not.

Chromy & Behul, (2013) published work on measurement based admission control methods in IP networks. They simulated MBAC algorithms for different traffic which were Voice over IP, constant bit rate and video. They compared Measured Sum (MS), Hoeffding Bound (HB), Acceptance Region – Tangent at Peak (ACTP) and Acceptance Region – Tangent at Origin (ACTO) algorithm in terms of these traffic. Their simulated result showed that ACTP and ACTO algorithms were the best MBAC algorithm for networks with VoIP traffic. The most suitable MBAC algorithm for bandwidth estimation in network with CBR traffic was MS. They also showed that the link utilization for MS was around 95%. CBR was not really a type of traffic but QoS for either voice or video traffic and so should not be classified as traffic for analysis of these MBAC algorithms in this work.

Chromy*et al*.(2013(a)) studied admission control methods in IP networks. They simulated three Measurement Based Admission Control (MBAC) algorithms which they introduced as MBACs,

MBACf and MBACv. From their simulation, they observed that as the number of samples increased, the MBAC methods` memory would increase and time to decide on acceptance or rejection of new flow. They also argued that MBACv is suitable from the point of view equitable allocation of bandwidth than MBACs and MBACf. MBAC is based on measured data but the data used for this work were generated data and this may give different result.

Chromy *et al*.(2013(b)) presented a work on admission control in IP Multimedia Subsystem (IMS) networks. They simulated admission control methods in MATLAB environment. The admission control methods simulated in their work were simple sum, measured sum and Hoeffding bound. Their simulation result showed that in case of higher traffic load, the selection of admission control algorithm was very important and from the effective bandwidth utilization point of view, the Hoeffding bound method was most suitable. The admission methods simulated in their work were Measurement Based Admission Control (MBAC) methods. MBAC was based on measured data. Best simulation result could be obtained from these methods if measured data was obtained, studied, and analysed before considering the nature of data to be used in the simulation but that was not so in this work.

Farooq & Kunz, (2015) presented an article on measurement-based available bandwidth estimation and flow admission control algorithm for ad hoc IEEE 802.15.4 based wireless multimedia networks. They highlighted the factors that must be considered for a flow admission control algorithm in wireless ad hoc networks. They also presented a novel available bandwidth-based flow admission control algorithm (BandEst) for ad hoc IEEE 802.15.4-based networks that considered the identified factors. They stated that one of the main contributions of BandEst was that it proactively considered the complete IEEE 802.15.4's unslotted CSMA-CA MAC layer overhead considering the future data load. One of the major factors of measurement-based

admission control is the decision to either admit or reject traffic. This was not evident in this work.

In view of the limitations observed from the reviewed literature, a Measurement Based Admission Control (MBAC) algorithm that reduces IPLR and increases bandwidth utilization for edge of carrier Ethernet network is developed. Unlike Chromy *et al,* 2013(b) that did notconsider measured data before developing their algorithm, measured data from MTN Lagos network was used for the development of the MBAC algorithm. The developed algorithm was implemented at the edge of the network. This kind of implementation will reduce or eliminate QoS arising from delay and latency which Yerima,2011 did not consider as their algorithm was applied both at the edge and core of the network.

CHAPTERTHREE

METHODOLOGY

3.1 Introduction

This chapter covers the methods adopted to develop Measurement-Based Admission Control (MBAC) algorithm that reduces IP Packet Loss Ratio (IPLR) and increases bandwidth utilization for edge of carrier Ethernet network. The methodologies carriedout include: collection of measured data, development of the MBAC algorithm,and simulation of the developed algorithm using C-sharp (C#) platform as well as validation of the developed algorithm by comparing it with the measured data obtained from mobile network.

3.2 Collection of Measured Data

The data used in this research were obtained from MTN network Lagos, Nigeria. There are two sets of data collected:

i. the data from 1 Gb Ethernet interface of an ETEX10S card. ETEX10S is a packet capability module on Tellabs 6300 multiplexer that supports Ethernet over SDH services. The data collected from the ETEX10S were studied, analysed, and used for the development of MBAC algorithm for SDH system on a carrier Ethernet network.
ii. the data from 1 Gb Ethernet interface (Ethernet line card) of Tellabs 8660 multiplexer. The Tellabs 8660 support Ethernet over WDM services and data collected from it were studied, analysed, and used for the development of MBAC algorithm for WDM system on a carrier Ethernet network.

These multiplexers were accessed through the Tellabs Intelligent Network Management (INM) and the data were pulled from performance management tool. The parameters that constituted these data are listed and defined in section 2.2.5.

The measured data was obtained for period of three months, from January to March, 2016. This was due to difficulty in having access and confidentiality of data by mobile network operators. They were pulled out on normal working days, Monday to Friday excluding public holidays. Appendix A shows the data collected during the stated period.

The number of traffic sources that arrived at the Ethernet interfaces of both the ETEX10sand Tellabs 8660 considered for this work is 35. Fig. 3.1 and Fig.3.2 illustrate the measurement set up for the collection of measured data from the Ethernet interfaces of SDH and WDM system respectively.

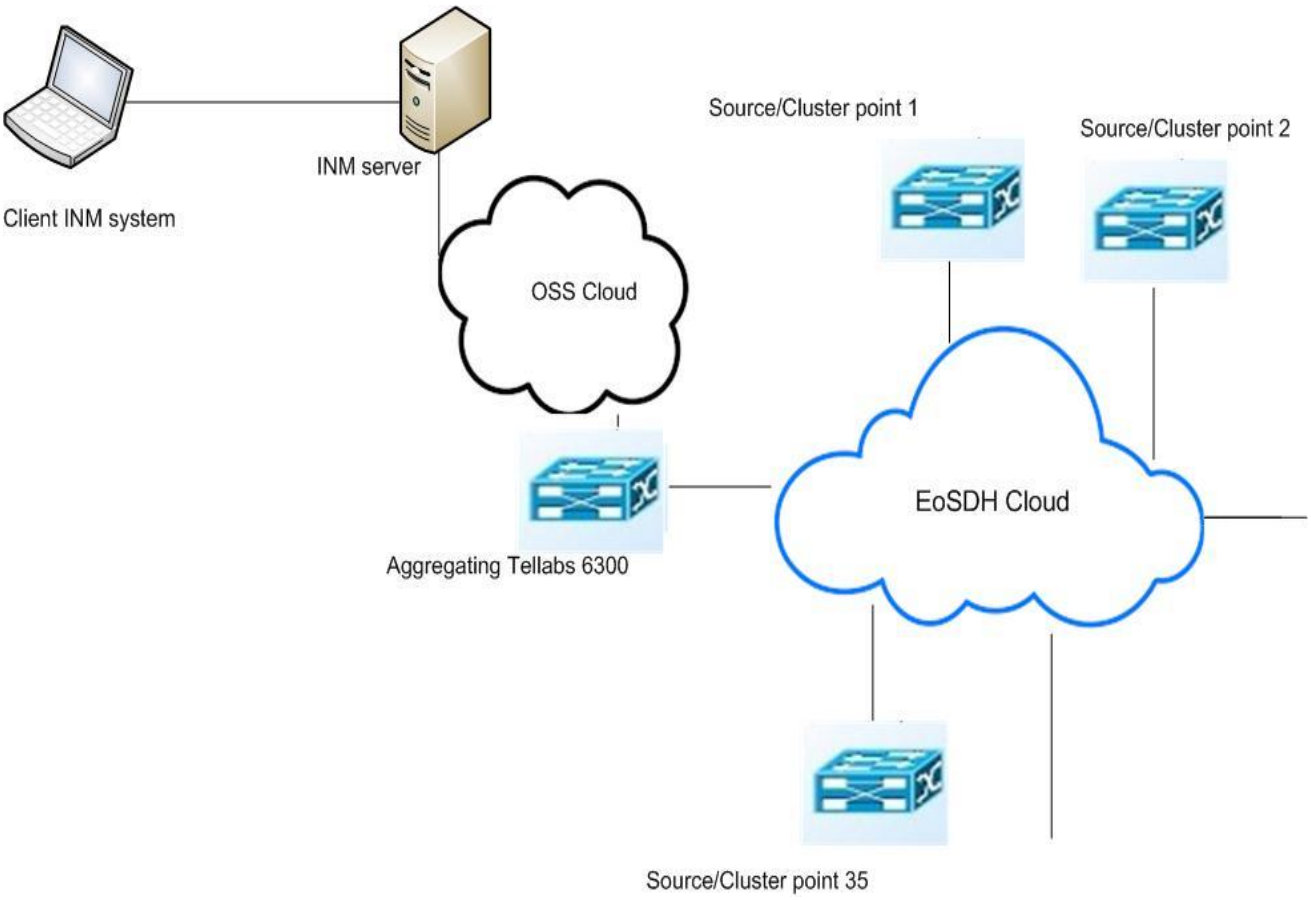


Fig. 3.1: Measurement Set up for Collection of Measured Data for SDH System

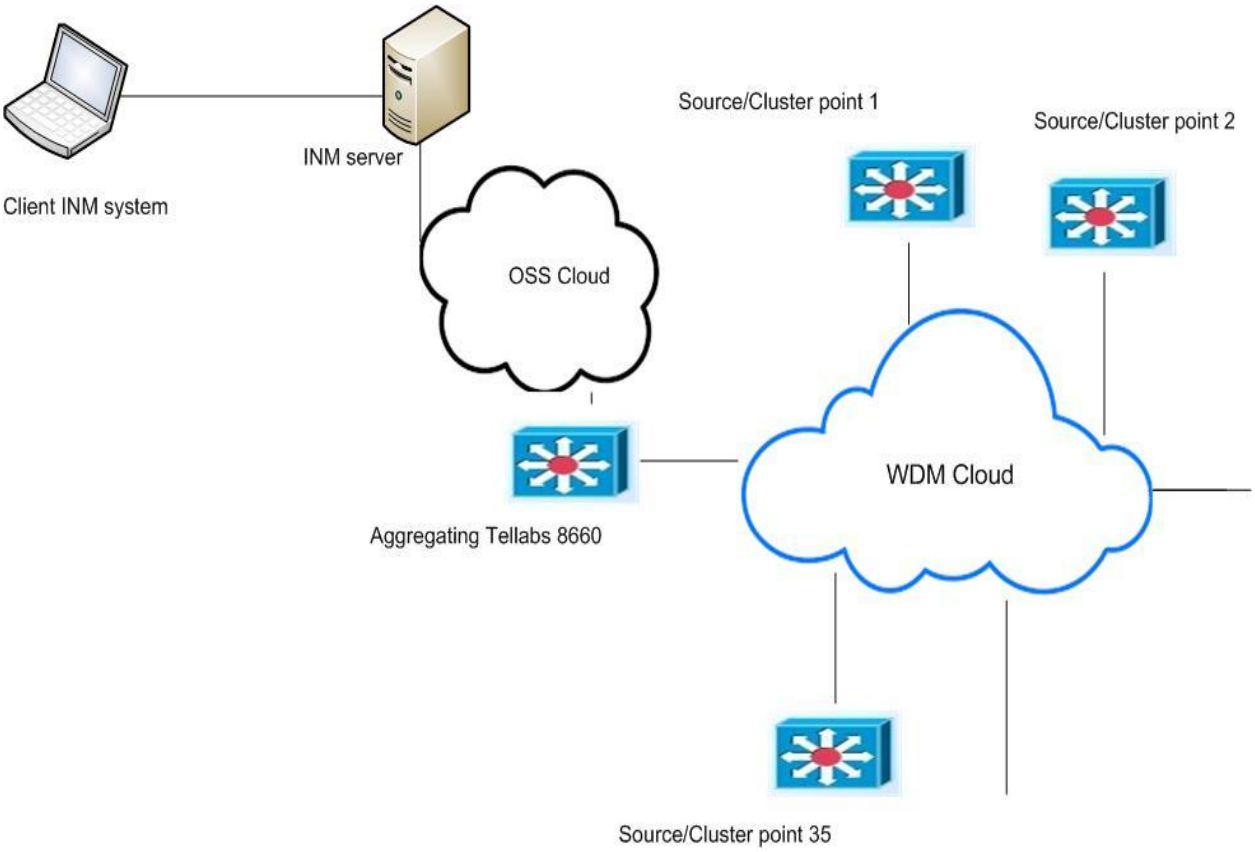


Fig. 3.2: Measurement Set up for Collection of Measured Data for WDM System

In this measurement set up, the multiplexers were reachable through the Data Communication and Network (DCN) to the INM server over the Operations Support Systems (OSS) network. All the equipment set up in fig. 3.1 and fig. 3.2 excluding the client INM system is MTN Lagos standard network. The client INM system was set up and granted access to the INM server for collection of the measured data. The measurement data was pulled from the Tellabs INM installed on the client system. The client system connects to the server on LAN. The packets from the 35 sources, aggregate on the Ethernet interface of the Tellabs 6300 and 8660 multiplexers. These sources are also multiplexers located at the access of the edge network. The aggregating multiplexers serve as the edge multiplexers between the core and access of the network.

3.3 Development of MBAC Algorithm for WDM and SDH System

Equation (2.3) and equation (2.38) was used to developed the MBAC algorithm.Equation (2.3) was used to compute the IPLR while equation (2.38) was used to determine the estimated bandwidth (higher sub-layer). The MBAC algorithm reduced IPLR, optimized the higher sub-layer,and bridge the control gap between the lower sub-layer and higher sub-layer. Fig. 3.3 represents the flow chart that was used to develop the MBAC algorithm for both WDM and SDH system. The implementation of the flow chart is depicted in Appendix B.

At the start of the algorithm, packets from the lower sub-layer (Lsl) and the bandwidth of the higher sub-layer (Hsl) were probed. A decision is then taken for whether Lsl < Hsl or not. If the answer to the decision is no, packets from Lsl will be rejected and the estimated bandwidth, C_{est} that will serve the Lsl will be computed using equation (2.38). The computed C_{est} will be sent to Hsl through a control to set its bandwidth to C_{est}. After the optimization of Hsl, the condition will be tested again, the decision will take yes and C_{est} will be checked to be sure it does not exceed the maximum set physical bandwidth. If C_{est} is less than the maximum bandwidth, IPLR will be computed using equation (2.3) to decide if Hsl should be optimized more to increase Hsl and reduce IPLR else, the packets will be admitted.

The distinct feature present in this MBAC algorithm flow chart but absent in the standard MBAC flow chart is the control which allowed the higher sub-layer to be upgraded. The standard MBAC algorithm flow chart has decisions on whether traffic should be allowed or rejected but does not have control for optimization of a parameter.

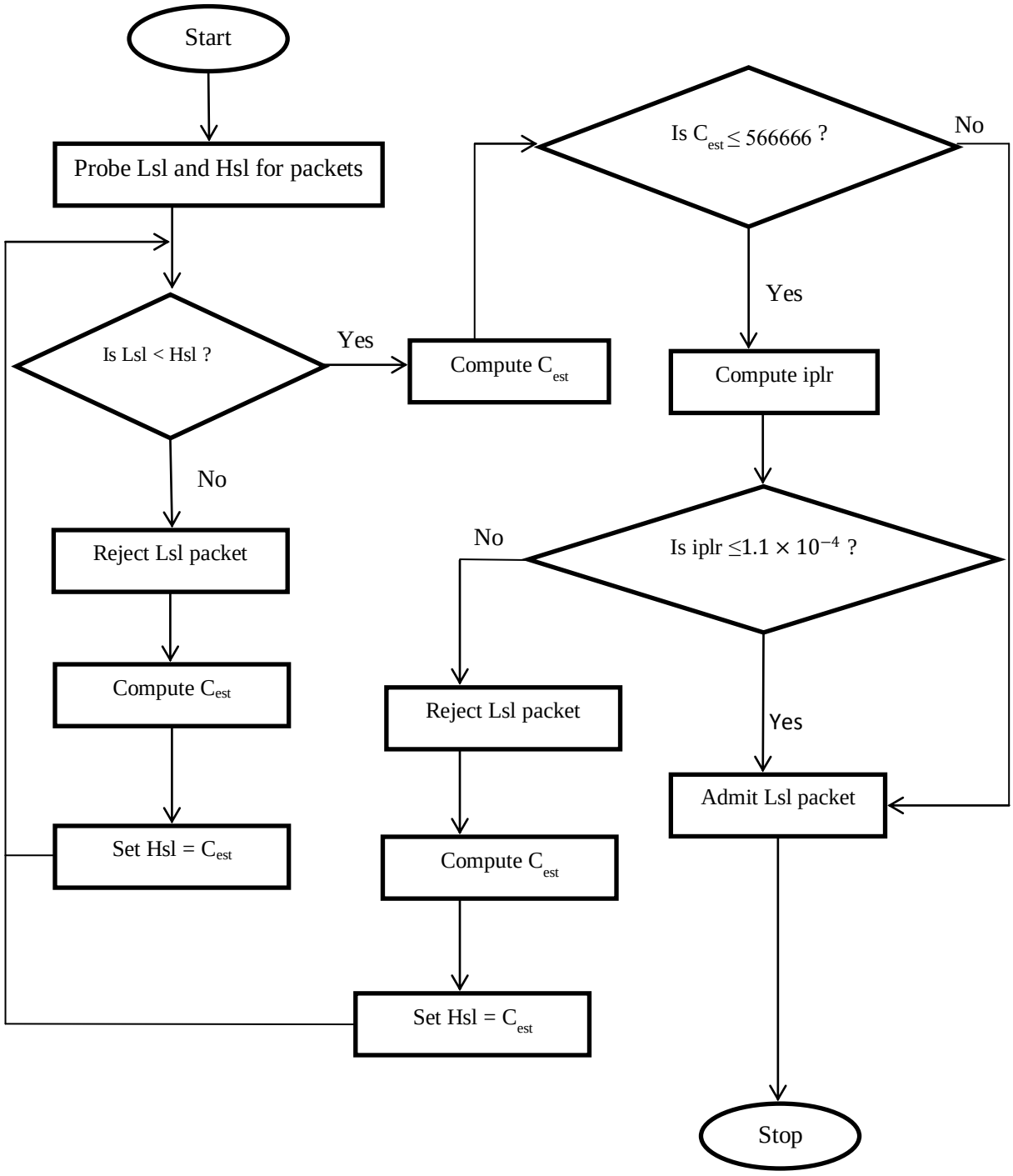


Fig. 3.3: Flow Chart for Development of MBAC Algorithm

3.4 Simulation of MBAC Algorithm

The developed MBAC algorithm in Fig. 3.3 was implemented using C# (C-sharp) programming environment. Lenovo laptop computer system with system properties: processor of Intel(R) Core(TM) i3 2.40GHZ, RAM of 4.00GB of 64 bits windows 8 was used.

3.5 Validation

The validation was carried out by comparing the processed results of measured data from the MTN Lagos network and simulated results of the developed algorithm using metrics IPLR and Bandwidth utilization as shown in Tables 4.11 to 4.14. A statistical validation was carried out using Welch's t-test as appropriate tool implemented on Statistical Package for Social Science (SPSS). The choice of t-test tool over other tools is because it compares the difference between two independent group means.

The validation procedures were executed through Figure 3.4 and the implementation for each model's performance metrics IPLR and Bandwidth utilization for the measured data and simulated results computed and discussed in the next section.

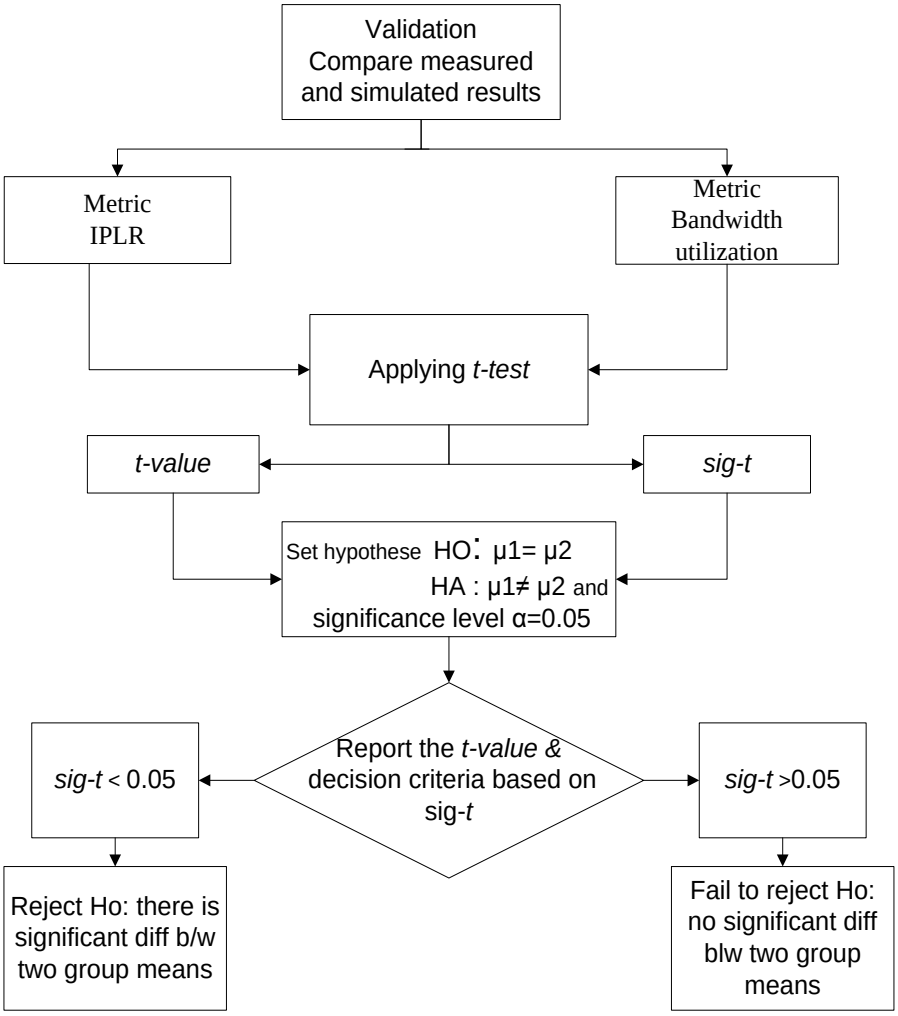


Figure 3.4: Block Diagram for Validation Procedure

CHAPTER FOUR

RESULTS AND DISCUSSIONS

4.1 INTRODUCTION

This chapter comprises of two parts: the results and discussions of the developed algorithm prior to and after the application using graphical plots and the results of the validationtechnique using t-test.

4.2 SIMULATION RESULTS OBTAINED

The simulation was performed after the MBAC algorithm was developed and the input data (lower sub-layer lower bound and lower sub-layer higher bound) was inserted. The lower sub-layer lower bound and lower sub-layer higher bound enabled the generation of the lower sub-layer and higher sub-layer data. The simulation result was carried out for the WDM and SDH system. The summary of the results for the WDM and SDH system are represented in Table 4.1 and 4.2 respectively.

Table 4.1: Simulation Result for the WDM System

Week	ifHcInUcastPkts (Pkt/s)	Bandwidth utilization (Pkt/s)	IPLR
1	390,660	0	0
2	390,869	526,073	0.742993843
3	390,956	526,103	0.743116842
4	390,750	525,927	0.742973835
5	390,695	525,844	0.742986513
6	391,365	526,545	0.743269806
7	391,017	526,175	0.743131088
8	390,804	525,968	0.743018587
9	390,441	525,611	0.742832627
10	390,757	525,890	0.743039419
11	390,574	525,706	0.742951383
12	391,014	526,132	0.743186121
13	390,487	525,677	0.742826869

Table 4.2: Simulation Result for SDH System

Week	ifHcInUcastPkts (Pkt/s)	Bandwidth utilization (Pkt/s)	IPLR
1	390,779	0	0
2	390,396	525,541	0.742845944
3	389,141	524,271	0.742251622
4	389,932	525,047	0.742661133
5	390,578	525,711	0.742951926
6	389,318	524,454	0.742330119
7	390,529	525,671	0.742915245
8	391,023	526,166	0.743155202
9	390,099	525,189	0.742778314
10	388,968	524,114	0.742143885
11	390,551	525,705	0.742909046
12	390,614	525,829	0.742853665
13	388,861	523,974	0.742137969

4.3 EVALUATION OF THE IPLR AND BANDWIDTH UTILIZATION

Evaluation of the IP Packet Loss Ratio (IPLR) and bandwidth utilization of the higher sub-layer for the WDM and SDH systems when the algorithm was applied and when the algorithm was not applied are considered in this section. This is to validate the developed algorithm.

4.3.1 Simulation Results of IPLR for WDM System

The plot of IPLR against load (ifHcInUcastPkts) for WDM system is depicted in Fig. 4.1. It is a superimposed graph of IPLR against load when the algorithm was applied and when it was not. “IPLR_meas” represent the IPLR for the measured data, when the algorithm was not applied and “IPLR_sim” represent the IPLR for the simulated data when the algorithm was applied. Table 4.3 is the extraction from measured and simulated data. Itwas used to plot the graph of Fig. 4.1.

Table 4.3: IPLR against Load WDM System

week	ifHcInUcastPkts (Pkt/s)	IPLR_Meas	IPLR_Sim
1	O-390660	1.043404405	0
2	O-390869	1.042831797	0.742993843
3	O-390956	1.06715654	0.743116842
4	O-390750	1.045113043	0.742973835
5	O-390695	1.049555673	0.742986513
6	O-391365	1.06178088	0.743269806
7	O-391017	1.060295948	0.743131088
8	O-390804	1.057364408	0.743018587
9	O-390441	1.042424129	0.742832627
10	O-390757	1.054763017	0.743039419
11	O-390574	1.064614345	0.742951383
12	O-391014	1.041534704	0.743186121
13	O-390487	1.052074717	0.742826869

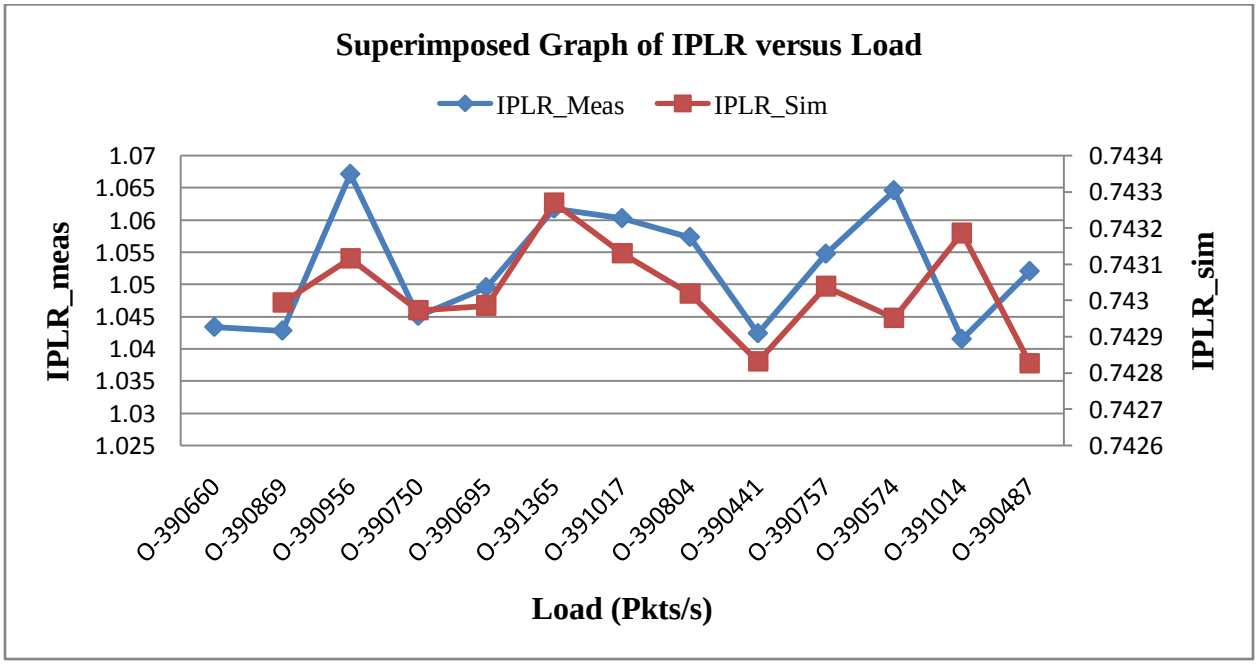


Fig. 4.1: Superimposed Graph of IPLR versus Load for WDM System

The graph covers the 13 weeks of measured data obtained from the network. Measurement Based-Admission Control (MBAC) algorithm is based on a preceding measured value. In the first week, there was no preceding measured value. This is the reason for absence of IPLR_sim in week 1. The real test for the algorithm was observed from the 2nd week. The least value of IPLR_meas was seen in week 12 while the highest value was seen in week 3. Reduction in IPLR was seen when the algorithm was applied. The minimum IPLR_sim achieved was 0.742832627 in week 9.

4.3.2 Simulation Results ofBandwidth Utilization for WDM System

The plot of bandwidth (ifHcOutUcastPkts) against load (ifHcInUcastPkts) is depicted in Fig. 4.2. It is a superimposed graph of bandwidth utilization versus load when the algorithm was applied and when it was not. Table 4.4 is the extraction from measured and simulated data. It was used to plot the graph of Fig. 4.2.

Table 4.4: Bandwidth Utilization against Load WDM System

week	ifHcInUcastPkts (Pkt/s)	Bandwidth utilization_meas (Pkts/s)	Bandwidth utilization_sim (Pkts/s)
1	O-390660	374409	0
2	O-390869	374815	526073
3	O-390956	366353	526103
4	O-390750	373883	525927
5	O-390695	372248	525844
6	O-391365	368593	526545
7	O-391017	368781	526175
8	O-390804	369602	525968
9	O-390441	374551	525611
10	O-390757	370469	525890
11	O-390574	366869	525706
12	O-391014	375421	526132
13	O-390487	371159	525677

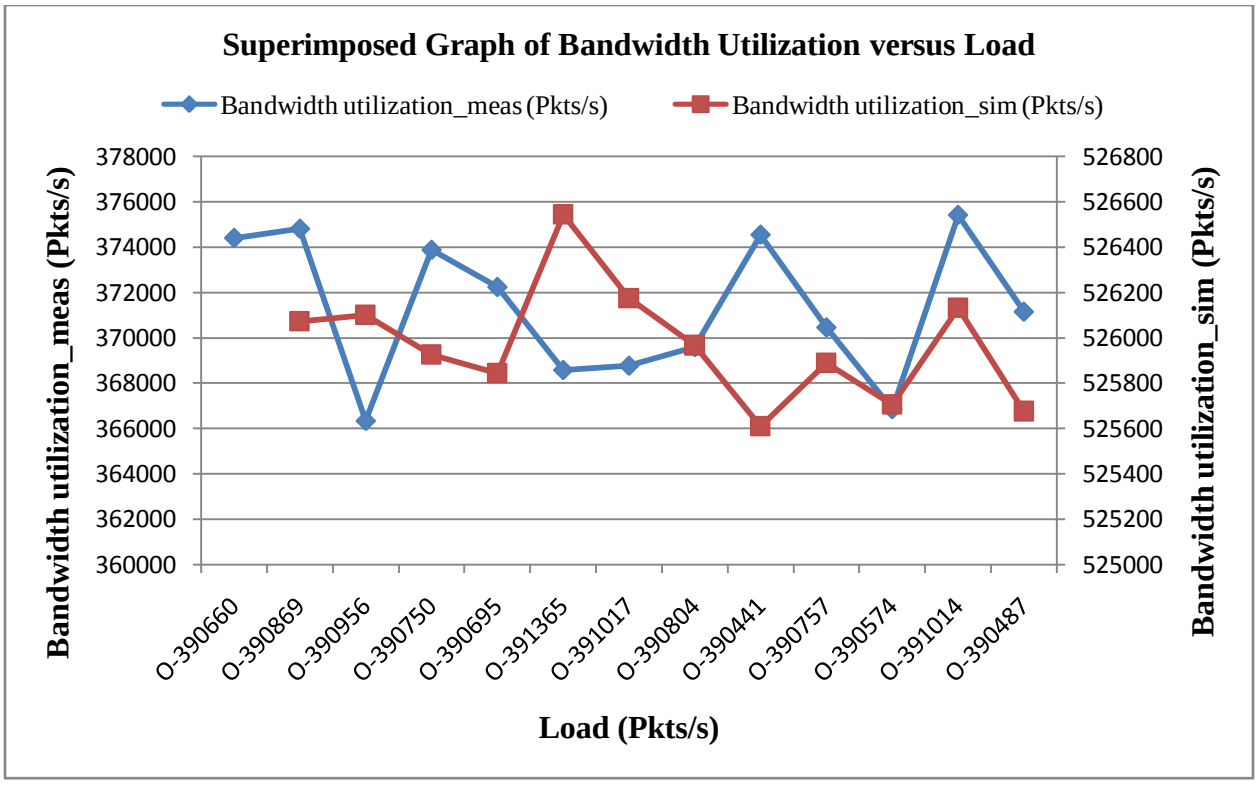


Fig.4.2: Superimposed Graph of Bandwidth Utilization versusLoad for WDM System

The graph covers the 13 weeks of measured data obtained from the network. There was no bandwidth utilization when the algorithm was applied in week 1. As earlier explained in section 4.3.1, it was a result of absence of preceding measured value in week 1. The bandwidth utilization of the measured value was observed to be less than the load throughout the 13 weeks. This was one characteristic observed with the WDM system and this contributed to the large value of IPLR_meas in the graph of Fig. 4.1. The bandwidth increased more than the load when the algorithm was applied for the period of the simulation. The maximum bandwidth (526545 pkts/s) when the algorithm was applied was achieved in week 6.

4.3.3 Simulation Results of IPLR for SDH System

The plot of IPLR against load (ifHcInUcastPkts) for SDH system is depicted in Fig. 4.3. It is a superimposed graph of IPLR against load when the algorithm was applied and when it was not. IPLR_meas represent the IPLR for the measured data, when the algorithm was not applied and IPLR_sim represent the IPLR for the simulated data, when the algorithm was applied. Table 4.5 is the extraction from measured and simulated data. It was used to plot the graph of Fig. 4.3.

Table 4.5: IPLR against Load SDH System

Week	ifHcInUcastPkts (Pkt/s)	IPLR_Meas	IPLR_Sim
1	O-390779	0.990218909	0
2	O-390396	0.992031144	0.742845944
3	O-389141	0.985980835	0.742251622
4	O-389932	0.989346182	0.742661133
5	O-390578	0.990640884	0.742951926
6	O-389318	0.98862612	0.742330119
7	O-390529	0.994043816	0.742915245
8	O-391023	0.995483672	0.743155202
9	O-390099	0.990523854	0.742778314
10	O-388968	0.989803423	0.742143885
11	O-390551	0.993437302	0.742909046
12	O-390614	0.992622936	0.742853665
13	O-388861	0.987708915	0.742137969

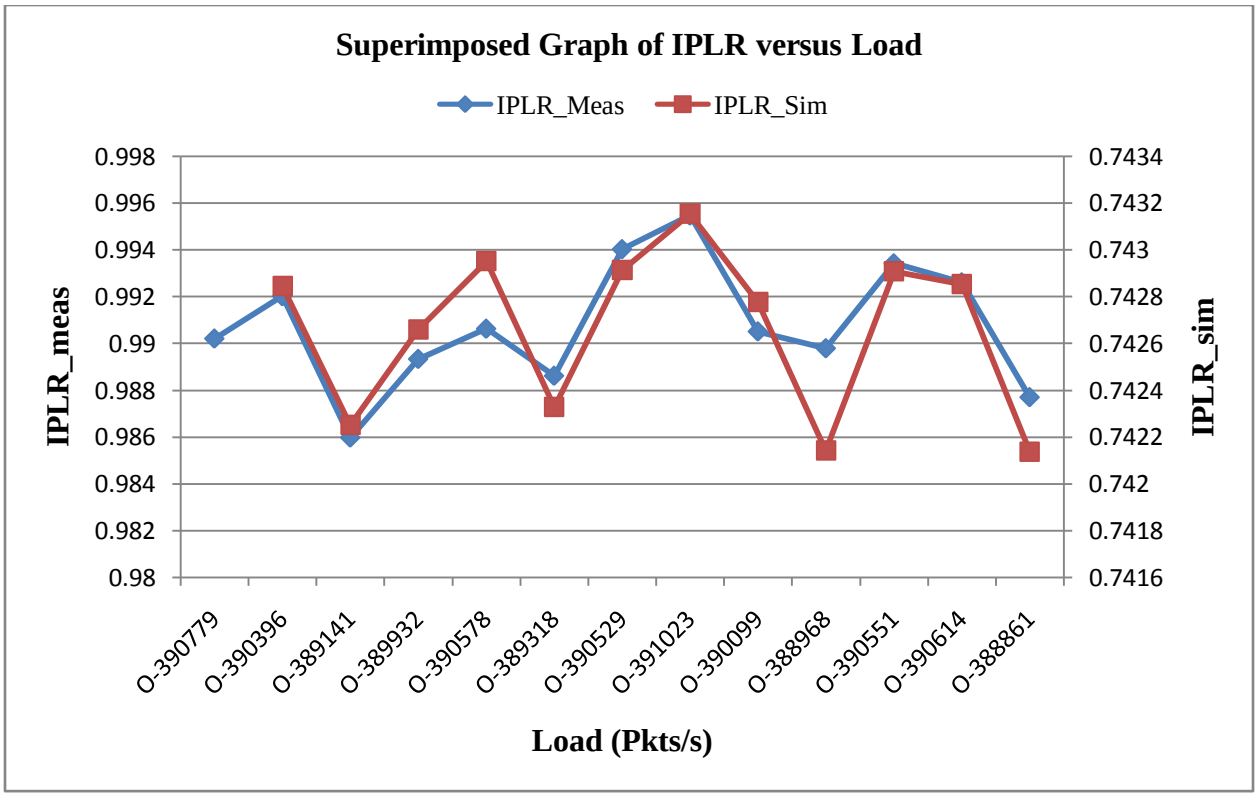


Fig.4.3: Superimposed Graph of IPLR versus Load for SDH System

The graph covers the 13 weeks of measured data obtained from the network. Like the WDM system, there was no preceding measured value in the first week. This was the reason for absence of IPLR_sim in week 1. The real test for the algorithm was observed from the 2nd week. The least value of IPLR_meas was seen in week 3 while the highest value was seen in week 8. Reduction in IPLR was seen when the algorithm was applied. The minimum IPLR_sim achieved was 0.742137969 in week 13.

4.3.4 Simulation Results ofBandwidth Utilization for SDH System

The plot of bandwidth (ifHcOutUcastPkts) against load (ifHcInUcastPkts) is depicted in Fig. 4.4. It is a superimposed graph of bandwidth utilization versus load when the algorithm is applied and when it was not. Table 4.6 is the extraction from measured and simulated data. It was used to plot the graph of Fig. 4.4.

Table 4.6: Bandwidth Utilization against Load SDH System

Week	ifHcInUcastPkts (Pkt/s)	Bandwidth utilization_meas (Pkts/s)	Bandwidth utilization_sim (Pkts/s)
1	O-390779	394,639	0
2	O-390396	393,532	525,541
3	O-389141	394,674	524,271
4	O-389932	394,131	525,047
5	O-390578	394,268	525,711
6	O-389318	393,797	524,454
7	O-390529	392,869	525,671
8	O-391023	392,797	526,166
9	O-390099	393,831	525,189
10	O-388968	392,975	524,114
11	O-390551	393,131	525,705
12	O-390614	393,517	525,829
13	O-388861	393,700	523,974

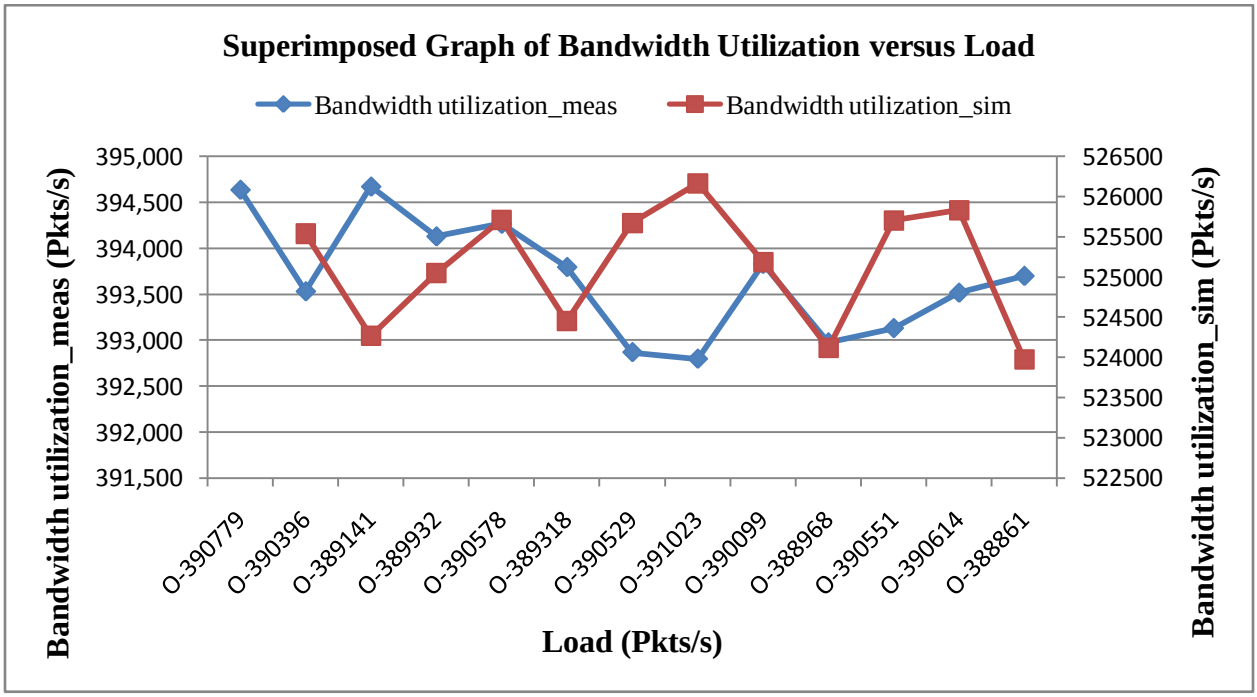


Fig.4.4: Superimposed Graph of Bandwidth Utilization versus Load for SDH System

The graph covers the 13 weeks of measured data obtained from the network. Unlike the WDM system, the bandwidth utilization of the SDH system was greater than its load for the period of the measurement. This was the reason IPLR_meas of the SDH system was less than one. The bandwidth utilization was optimizedwhen the algorithm was applied and this helps in reducing the IPLR. The maximum bandwidth (526166 pkts/s) when the algorithm was applied was achieved in week 8.

4.4 IPLR PERCENTAGE REDUCTIONS AND BANDWIDTH UTILIZATION

Table 4.7 and4.8 represent the percentage reduction of IPLR for the WDM and SDH system respectively. Table 4.9 and 4.10are the percentage improved bandwidth for the WDM and SDH system respectively. Fig. 4.5 and 4.6 illustrate the graphs of percentage reduction of IPLR for WDM and SDH system respectively. Furthermore, Fig. 4.7 and 4.8 illustrate the graph of percentage improved bandwidth for the WDM and SDH system respectively.

Table 4.7: Percentage Reduction of IPLR for WDM System

week	IPLR_Meas	IPLR_Sim	percentage reduction of IPLR (%)
1	1.043404405	0	0
2	1.042831797	0.742993843	28.75228343
3	1.06715654	0.743116842	30.36477648
4	1.045113043	0.742973835	28.90971561
5	1.049555673	0.742986513	29.20942333
6	1.06178088	0.743269806	29.99781595
7	1.060295948	0.743131088	29.91286169
8	1.057364408	0.743018587	29.72918505
9	1.042424129	0.742832627	28.73988558
10	1.054763017	0.743039419	29.5538991
11	1.064614345	0.742951383	30.21403598
12	1.041534704	0.743186121	28.64509287
13	1.052074717	0.742826869	29.39409561

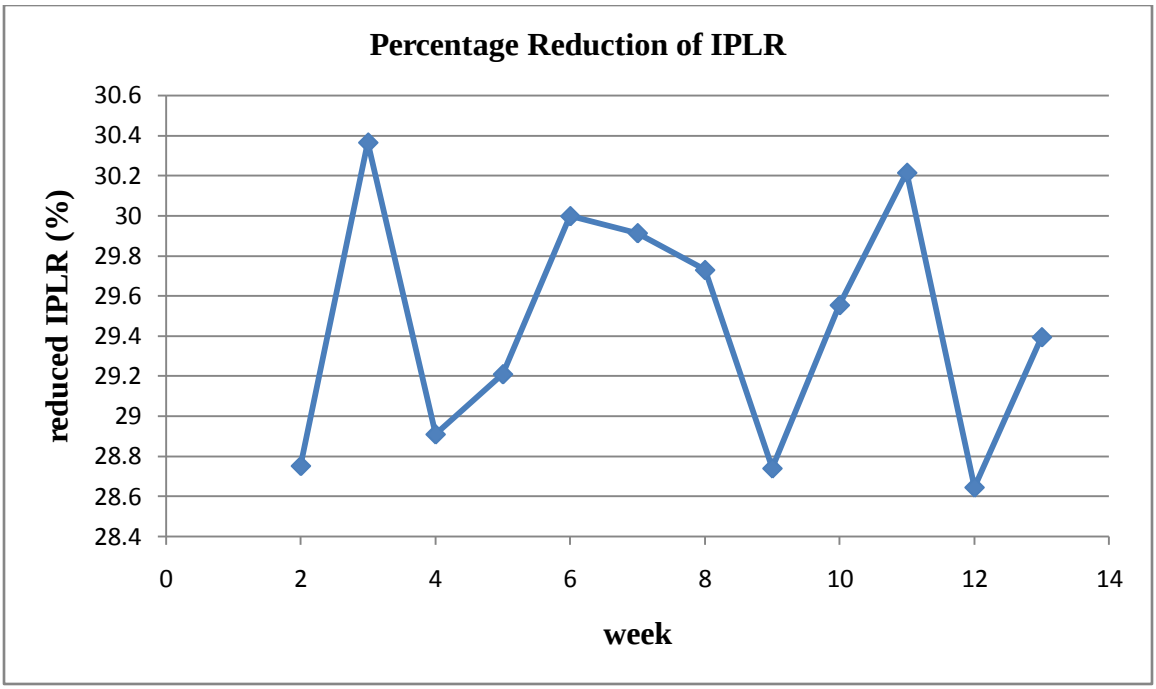


Fig. 4.5 Percentage Reduction of IPLR for WDM System

Table 4.8: Percentage Reduction of IPLR for SDH System

Week	IPLR_Meas	IPLR_Sim	percentage reduction of IPLR (%)
1	0.990218909	0	0
2	0.992031144	0.742845944	25.11868722
3	0.985980835	0.742251622	24.7194676
4	0.989346182	0.742661133	24.93414875
5	0.990640884	0.742951926	25.00290083
6	0.98862612	0.742330119	24.91295709
7	0.994043816	0.742915245	25.26333011
8	0.995483672	0.743155202	25.34732385
9	0.990523854	0.742778314	25.01156726
10	0.989803423	0.742143885	25.0210832
11	0.993437302	0.742909046	25.21832587
12	0.992622936	0.742853665	25.16255285
13	0.987708915	0.742137969	24.86268403

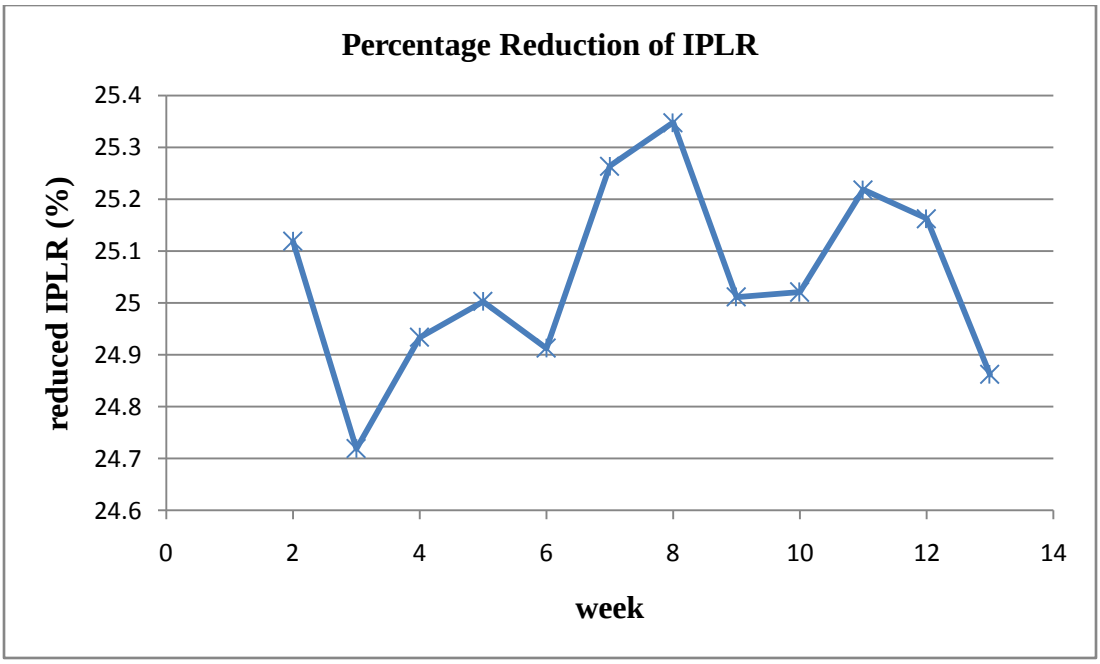


Fig. 4.6 Percentage Reduction of IPLR for SDH System

Table 4.9: Percentage Improved Bandwidth of IPLR for WDM System

week	Bandwidth utilization_meas (Pkts/s)	Bandwidth utilization_sim (Pkts/s)	percentage improved bandwidth (%)
1	374409	0	0
2	374815	526073	28.75228343
3	366353	526103	30.36477648
4	373883	525927	28.90971561
5	372248	525844	29.20942333
6	368593	526545	29.99781595
7	368781	526175	29.91286169
8	369602	525968	29.72918505
9	374551	525611	28.73988558
10	370469	525890	29.5538991
11	366869	525706	30.21403598
12	375421	526132	28.64509287
13	371159	525677	29.39409561

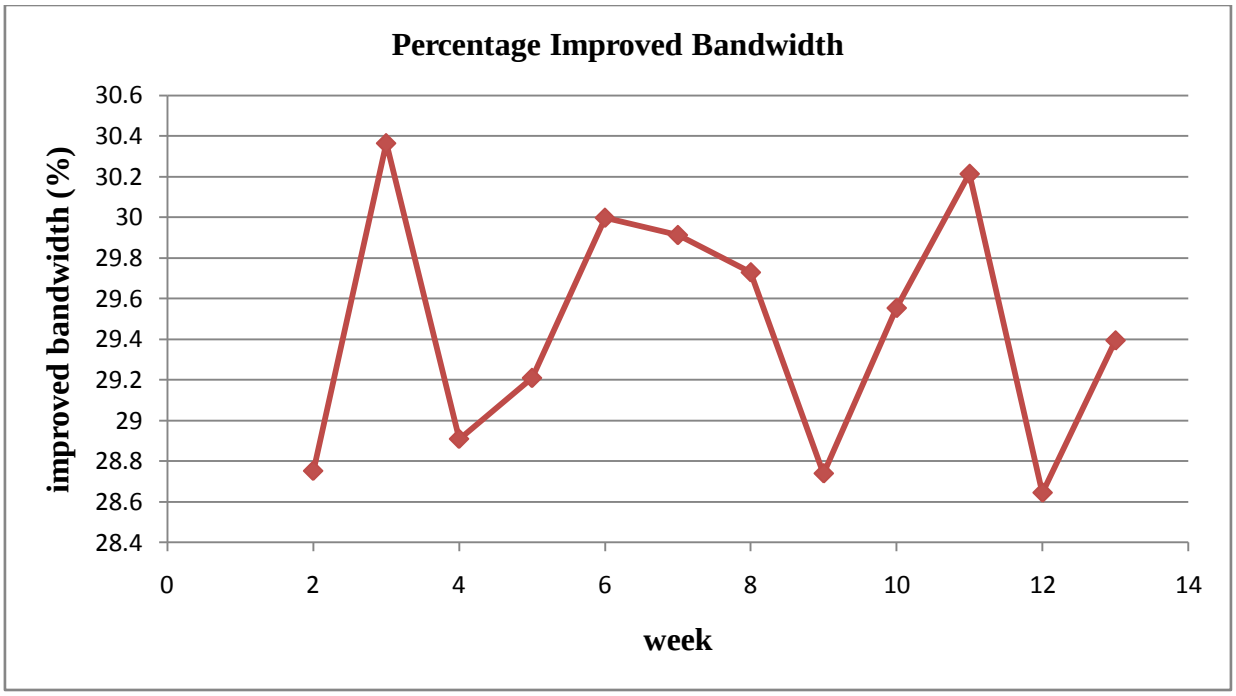


Fig. 4.7Percentage Improved Bandwidth for WDM System

Table 4.10: Percentage Improved Bandwidth of IPLR for SDH System

Week	Bandwidth utilization_meas (Pkts/s)	Bandwidth utilization_sim (Pkts/s)	percentage improved bandwidth (%)
1	394,639	0	0
2	393,532	525,541	25.11868722
3	394,674	524,271	24.7194676
4	394,131	525,047	24.93414875
5	394,268	525,711	25.00290083
6	393,797	524,454	24.91295709
7	392,869	525,671	25.26333011
8	392,797	526,166	25.34732385
9	393,831	525,189	25.01156726
10	392,975	524,114	25.0210832
11	393,131	525,705	25.21832587
12	393,517	525,829	25.16255285
13	393,700	523,974	24.86268403

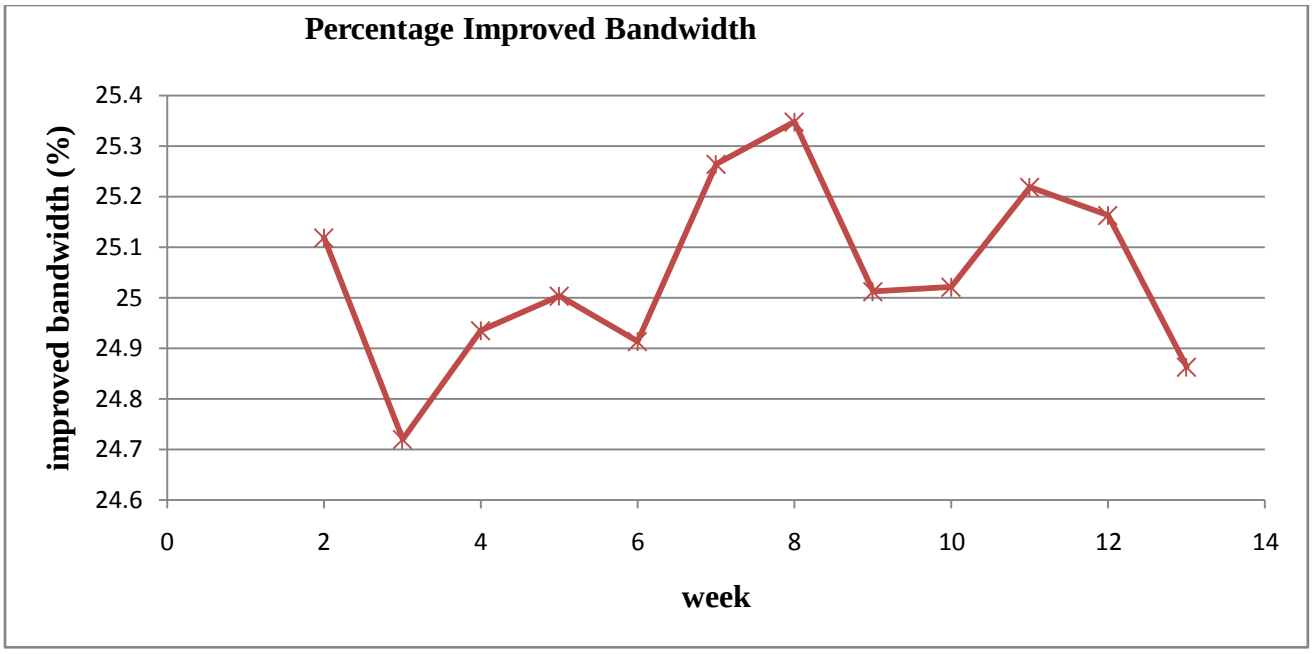


Fig. 4.8 Percentage Improved Bandwidth for SDH System

The highest percentage reduction of IPLR achieved for WDM system within the 13 weeks was 30.36% while that of SDH system was 25.34%. The highest improved percentage bandwidth utilization for WDM system was 30.36% while that of SDH system was 25.34%. An interesting observation was the percentage reduced IPLR of the systems at any point was found to be equivalent to the improved percentage bandwidth utilization;This implied that equation (2.44) equals equation (2.45), as observed throughout the period of the measurement. The independent parameters of equation (2.3) which was used to compute IPLR was part of the parameters for calculation of percentage improved bandwidth utilization. Equation (2.44) equatesequation (2.45) if the percentage reduction of IPLR is calculated from its independent parameters. This relation can be of great use for analysis of IP network.

4.5 Validation of Results of T-Test

The validation of the measured results from MTN Lagos network and simulation results of the developed algorithm was carried out using Welch's t-test of unequal variances implemented on SPSS software tool as discussed in the sub section.

4.5.1 Results of Measured Data

The thirteen weeks of the measured data for WDM and SDH system from MTN Lagos network are shown on Table 4.11 and 4.12 respectively. These results were used to test the validity of developed algorithm using statistical Welch's t-test discussed.

Table 4.11: WDM Measured Data

Week	ifHcInUcastPkts (Pkt/s)	Bandwidth utilization (Pkt/s)	IPLR
1	390,660	374,409	1.043404405
2	390,869	374,815	1.042831797
3	390,956	366,353	1.06715654
4	390,750	373,883	1.045113043
5	390,695	372,248	1.049555673
6	391,365	368,593	1.06178088
7	391,017	368,781	1.060295948
8	390,804	369,602	1.057364408
9	390,441	374,551	1.042424129
10	390,757	370,469	1.054763017
11	390,574	366,869	1.064614345
12	391,014	375,421	1.041534704
13	390,487	371,159	1.052074717

Table 4.12: SDH Measured Data

Week	ifHcInUcastPkts (Pkt/s)	Bandwidth utilization (Pkt/s)	IPLR
1	390,779	394,639	0.990218909
2	390,396	393,532	0.992031144
3	389,141	394,674	0.985980835
4	389,932	394,131	0.989346182
5	390,578	394,268	0.990640884
6	389,318	393,797	0.98862612
7	390,529	392,869	0.994043816
8	391,023	392,797	0.995483672
9	390,099	393,831	0.990523854
10	388,968	392,975	0.989803423
11	390,551	393,131	0.993437302
12	390,614	393,517	0.992622936
13	388,861	393,700	0.987708915

4.5.2 Results of Simulated Data

The simulated results for the WDM and SDH shown in Table 4.13 and 4.14 respectively, were used for the validation test.

Table 4.13: WDM Simulated Data

Week	ifHcInUcastPkts (Pkt/s)	Bandwidth utilization (Pkt/s)	IPLR
1	390,660	0	0
2	390,869	526,073	0.742993843
3	390,956	526,103	0.743116842
4	390,750	525,927	0.742973835
5	390,695	525,844	0.742986513
6	391,365	526,545	0.743269806
7	391,017	526,175	0.743131088
8	390,804	525,968	0.743018587
9	390,441	525,611	0.742832627
10	390,757	525,890	0.743039419
11	390,574	525,706	0.742951383
12	391,014	526,132	0.743186121
13	390,487	525,677	0.742826869

Table 4.14: SDH Simulated Data

Week	ifHcInUcastPkts (Pkt/s)	Bandwidth utilization (Pkt/s)	IPLR
1	390,779	0	0
2	390,396	525,541	0.742845944
3	389,141	524,271	0.742251622
4	389,932	525,047	0.742661133
5	390,578	525,711	0.742951926
6	389,318	524,454	0.742330119
7	390,529	525,671	0.742915245
8	391,023	526,166	0.743155202
9	390,099	525,189	0.742778314
10	388,968	524,114	0.742143885
11	390,551	525,705	0.742909046
12	390,614	525,829	0.742853665
13	388,861	523,974	0.742137969

4.6 Validation of T-Test Results Obtained

The results of the measured and simulated data for metrics IPLR and bandwidth utilization as shown in Table 4.11 to 4.14 for each transport system were inserted on t-test of IBM SPSS statistics tool following the steps depicted in Fig. 3.4. The two results of the metrics IPLR and bandwidth utilization obtained captured the extracted results of Levene's test and t-test indices, as shown in Table 4.15 to 4.18.

4.6.1 Bandwidth Utilization T-Test Results

From Table 4.11 to 4.14,the results of the independent-sample t-test on bandwidth utilization between measured data and simulated results for the WDM and SDH systems are captured as shown in Table 4.15 and 4.16, respectively.

Table 4.15: Results of T-Test on Bandwidth Utilization for WDM System-Assumed Equal Variances

Bandwidth utilization sample	N	Mean	Std. Deviation	t-value	Sig-t (P)
measured data	12	371062.00	3135.88	-170.54	0.00
simulated result	12	525970.92	258.62		

From the results of Table 4.15, t-value = -170.54, P = 0.00. Since P(0.00) <α (0.005). Therefore, reject the null hypothesis, which means that, there is significant difference between the developed algorithm and existing one. This indicates that the developed algorithm is of better bandwidth utilization in comparison with the measured data of MTN Lagos network.

Table 4.16: Results of T-Test on Bandwidth Utilization for SDH System-Assumed Equal Variances

Bandwidth utilization sample	N	Mean	Std. Deviation	t-value	Sig-t (P)
measured data	12	393601.83	585.60	-476.72	0.00
simulated result	12	525139.33	754.65		

From the results of Table 4.16, t-value = -476.72, P = 0.00. Since P(0.00) <α (0.005). Therefore, reject the null hypothesis, which means that, there is significant difference between the developed algorithm and existing one. This indicates that the developed algorithm is of better bandwidth utilization in comparison with the measured data of MTN Lagos network.

4.6.2 IPLR T-Test Results

From Table 4.11 to 4.14,the results of the independent-sample t-test on IPLR between measured data and simulated results for the WDM and SDH systems are respectively captured as shown in Table 4.17 and 4.18.

Table 4.17: Results of T-Test on IPLR for WDM System-Assumed Equal Variances

IPLR sample	N	Mean	Std. Deviation	t-value	Sig-t (P)
measured data	12	1.05	0.01	118.26	0.00
simulated result	12	0.74	0.00		

From the results of Table 4.17, t-value = 118.26, P = 0.00. Since P(0.00) <α (0.005). Therefore, reject the null hypothesis, which means that, there is significant difference between the developed algorithm and existing one. This indicates that the developed algorithm is of better performance in comparison with the measured data of MTN Lagos network.

Table 4.18: Results of T-Test on IPLR for SDH System-Assumed Equal Variances

IPLR sample	N	Mean	Std. Deviation	t-value	Sig-t (P)
measured data	12	0.99	0.00	306.97	0.00
simulated result	12	0.74	0.00		

From the results of Table 4.18, t-value = 306.97, P = 0.00. Since P(0.00) <α (0.005). Therefore, reject the null hypothesis, which means that, there is significant difference between the developed algorithm and existing one. This indicates that the developed algorithm is of better performance in comparison with the measured data of MTN Lagos network.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Summary

This research work has developed a Measurement-Based Admission Control (MBAC) algorithm for edge carrier Ethernet network to reduce IPLR and improve bandwidth utilization. The performance of the algorithm was validated with the measured data obtained from MTN Lagos network, using Welch's t-test of equal variances. The results of IPLR and bandwidth utilization were obtained. The research findings are itemized as:

i) The developed algorithm reduced WDM system IPLR by 30.36% and SDH system IPLR by 25.34% as compared to the baseline result before implementation of the MBAC algorithm.

ii) The developed algorithm improved WDM system bandwidth utilization by 30.36% and SDH system bandwidth utilization by 25.34% as compared to the baseline result before implementation of the MBAC algorithm.

iii) The t-test results obtained showed that the probability levels of WDM and SDH bandwidth utilization was 0.00, which was less than 0.05. This shows that the developed algorithm is of better bandwidth utilization compared with the measured data from MTN Lagos network.

iv) The t-test results obtained showed the probability levels of WDM and SDH IPLR was 0.00, less than 0.05. This shows that the developed algorithm is of higher performance compared with the measured data from MTN Lagos network.

5.2 Conclusion

The MBAC algorithm was developed and implemented on a C# platform. Based on the simulation results, WDM system IPLR was reduced by 30.36% and SDH system IPLR was reduced by 25.34% as compared to the baseline result before implementation of the MBAC algorithm. WDM system bandwidth utilization was improved by 30.36% and SDH system bandwidth utilization was improved by 25.34% as compared to the baseline result before implementation of the MBAC algorithm. From the t-test results obtained, the probability levels (P-values) of IPLR and bandwidth utilization for both edge systems were less than 0.05. This implies that P-value of the metrics reject the null hypothesis which means that the developed algorithm has better performance in comparison with the measured data from MTN Lagos network.

5.3 Significant Contributions

The significant contributions are as follows:

i) The research work reduced IPLR of WDM system by 30.36% and improved bandwidth utilization by 30.36%.

ii) The research work reduced IPLR of SDH system by 25.34% and improved bandwidth utilization by 25.34%.

iii) The research work established that the percentage reduced IPLR is equal to percentage improved bandwidth utilization of edge carrier Ethernet network.

5.4 Limitations

The limitations of this dissertation work are highlighted as follows:

i) The measured data obtained were collected from Monday to Friday. Saturday and Sunday were excluded because these days are not office working days. As a result of this, the measured data weekly average is of five days instead of seven days and these did not represent the true picture of the network studied.

ii) The physical bandwidths of the edge equipment in the network are mostly set to be upgraded when 80-90% of the actual bandwidth has been used. This process hinders the MBAC algorithm the maximum utilization of the physical bandwidth. This also limits the least IPLR that can be achieved by applying the algorithm.

5.5 Recommendations for further work

The following are recommended areas for extension and improvement of this research work:

i) This research can be extended to access and core equipment which are MIB capable.

ii) Research on why physical bandwidths of edge equipment are not 100% utilized before congestion is notice.

iii) Develop an end-to-end model for packet loss management in an IP network.

REFERENCES

Agrawal P. G (2011), "*Fiber optic communication systems",* (4thed). John wiley & sons, Ltd, England. /11, 226-227, 346-348

Ajmal M. and Robert F. (2014), *"Optical Communication and Networking",* lecture note on optical communication and networking, department of Electrical Engineering, Linkoping University. /1-4

Alwakeel S. S and Alotaibi N. M, (2011), "End-to-end measurement based admission control VoIP protocol with loss policy", *Journal of King Saud university-computer and information sciences. 23/37-43.*

Ammar D., Begin T., Guerin-Lassous I., and Noirie L(2011), "Evaluation and comparison of MBAC solutions", IEEE 36th conference, Local computer Networks (LCN), 2011. /4-7

Antonio R. C Paiva. (2010). Hypothesis Testing. University of Utah, scientific Computing and Imaging. ECE 3530 lecture note.pp 1-64

Auge J., Oueslati S., and Roberts J. (2011), "Measurement-based admission control for flow-aware implicit service differentiation", Tele-traffic congress (ITC), 2011. /206-213

Brewer O. T and Ayyagari A (2010), "Comparison and analysis of measurement and parameter based admission control methods for quality of service (Qos) provisioning", Military communications conference, 2010. /184-188

Bohnert T.M (2011), "Admission control for packet Networks" Teletraffic theory II, lecture on Admission control, university of Coimbra, Portugal. /1-38.

Chromy E and Behul T (2013), "Measurement based admission control methods in IP Networks", I. J Information technology and computer science, 2013. 10/1-8

Chromy E., Jadron M., and Behul T. (2013)(a), "Admission control methods in IP Networks", Hindawi publishing corporation-Advances in multimedia. 2013(918930)/1-7

Chromy E., Jadron M., Kavacky M. and Klucik S. (2013)(b), "Admission control in IMS Networks", Information and communication technologies and services, 2013. 11(5)/373-379

Cisco (2016), "Cisco visual Networking index: Global Mobile Data traffic forecast update, 2015-2020-white paper" /1-38. Retrieved from http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html. 02/12/2016.

Dass,H.K. (2015).*Advanced Engineering Mathematics*. S. Chand publisher, England, pp.763-847.

DoD. (2002) DoD Instruction 5000.61: "Modeling and Simulation (Mand S) Verification, Validation, and Accreditation (VVandA)", [online], Defense Modeling and Simulation Office, Office of the Director of Defense Research and Engr.,www.dmso.mil/docslib

EPA 530/R-09-007. (2009).Statistical analysis of ground water monitoring data at RCRA facilities. Environment protection agency office of resource conservation and recovery, pp 368-373.

Erfani S and Malek M (1989), "Dynamic access capacity management in a multiservice packet-mode environment", proceedings of the 1989 Network management and control workshop. /277-285

Farooq M.O and Kunz T (2015), "BandEst: Measurement-Based available bandwidth estimation and flow admission control algorithm for Ad Hoc IEEE 802.15.4-Based wireless

multimedia Networks" Hindawi publishing corporation, international journal od distributed sensor Networks. 2015(539048)/1-15.

Gao K., Ozdemir O., Pados A. D., Batalama N. S., Melodia T., and Drozd L. A. (2012), "Cognitive code-Division channelization with admission control" Hindawi publishing corporation, *Journal of Computer Networks and communications. 2012(510942)/1-9.*

Guerin R., Ahmadi H., and Naghshineh M. (1991), "Equivalent capacity and its application to bandwidth allocation in high-speed Networks",*IEEE journal on selected areas in communications, September, 1991. 9(7)/968-981*

Harvey L (2010), "*Microwave Transmission Networks: Planning, design and deployment (2nd Ed.)",* McGraw-Hill companies, U.S. /15-18

Heena R, Manish G. and Dharm S. (2011), "Simulation of measurement based admission control using NS2",*International Journal of computer applications (IJCA), 2011. /6-8*

Ian A. G. and Peter M. G. (2010) *"Digital Communications"*, (3rd edition) Prentice hall, London. /816-835.

ITU-T M.2301 (2002), "Performance objectives and procedures for provision and maintenance of IP-based networks". /7-11. Retrieved from www.itu.int/rec/T-REC-M.2301-200207-I/en. 20/10/2014.

Jain R. K (2013), *"Principles of Synchronous Digital Hierarchy",* CRC Press, Taylor and Francis group Boca Raton London. /249-309

Kosten L (1986), "Liquid models for a type of information Buffer problems", Delft Prog. Rep.: Math. Eng., Math and Inform. Eng., 1986. 11/71-86

Martis, M. S. (2006), Validation of Simulation Based Models: A Theoretical Outlook" *The Electronic Journal of Business Research Methods, 4(1): 39 -46*

McCloghrie k. and Kastenholz F. (2000), "RFC 2863 – The interfaces Group MIB". /29-45. Retrieved from https://tools.ietf.org/html/rfc2863. 07/08/2014.

Mitra D. (1988), "Stochastic theory of a fluid model of producers and consumers coupled by a buffer", *Advanc. Appl. Prob., vol. 20. /646-676.*

Raj J (2013), "Carrier Ethernet", Lecture on Carrier Ethernet Washington University in Saint Louis. /1-48

Rajiv R; Kumar N. S; & Galen H. S (2010), "*Optical networks: A practical perspective* (3rd*ed"),* Morgan Kaufmann, USA. /11, 26-27, 29, 377-378, 391-392, 433-453.

Richard W (2011), "Markov Chains", lecture on Markov theory, University of Cambridge, 2011./1-15

Uwaechia, A.N., Dajab .D. D, Adeboye B.A.(2013).Performance modelling of Ahmadu Bello University Data Network using a Storage Area Network (SAN). *Design Concept International Journal of Computer Application,65(10):6-13.*

Vijay K. G (2011), *"Wireless Communications and Networking"*, Morgan Kaufmann publisher, U. S/115-152

Yerima S. Y (2011), "Implementation and evaluation of measurement-based admission control schemes within a converged Networks Qos management framework",*International journal of computer Networks and communications (IJCNC). 3(4)/137-152*

APPENDIX A

The average weekly measured data

Table 1.1 Measured data obtained from WDM system

week No. average	ifHcInUcastPkts (/s)	ifHcOutUcastPkts (/s)	ifInNUcast Pkts	ifOutNUcast Pkts	ifInDiscard s	ifOutDiscard s	ifInError s	ifOutError s	ifInUnknown Protos
1	390,660	374,409	0	0	56,123	0	0	0	0
2	390,869	374,815	0	0	71,341	0	0	0	0
3	390,956	366,353	0	0	41,853	0	0	0	0
4	390,750	373,883	0	0	11,768	0	0	0	0
5	390,695	372,248	0	0	90,690	0	0	0	0
6	391,365	368,593	0	0	10,790	0	0	0	0
7	391,017	368,781	0	0	39,013	0	0	0	0
8	390,804	369,602	0	0	56,091	0	0	0	0
9	390,441	374,551	0	0	29,812	0	0	0	0
10	390,757	370,469	0	0	47,871	0	0	0	0
11	390,574	366,869	0	0	89,381	0	0	0	0
12	391,014	375,421	0	0	68,648	0	0	0	0
13	390,487	371,159	0	0	84,981	0	0	0	0

Table 1.2 Measured data obtained from SDH system

week No. average	ifHcInUcastPkts (/s)	ifHcOutUcastPkts (/s)	ifInNUcast Pkts	ifOutNUcast Pkts	ifInDiscard s	ifOutDiscard s	ifInError s	ifOutError s	ifInUnknown Protos
1	390,779	394,639	0	0	0	1	0	0	0
2	390,396	393,532	0	0	0	0	0	0	0
3	389,141	394,674	0	0	0	0	0	0	0
4	389,932	394,131	0	0	0	3	0	0	0
5	390,578	394,268	0	0	0	0	0	0	0
6	389,318	393,797	0	0	0	0	0	0	0
7	390,529	392,869	0	0	0	0	0	0	0
8	391,023	392,797	0	0	0	0	0	0	0
9	390,099	393,831	0	0	0	6	0	0	0
10	388,968	392,975	0	0	0	0	0	0	0
11	390,551	393,131	0	0	0	0	0	0	0
12	390,614	393,517	0	0	0	0	0	0	0
13	388,861	393,700	0	0	0	0	0	0	0

APPENDIX B

Program pseudo code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using CrystalDecisions.Windows.Forms;
using CrystalDecisions.CrystalReports.Engine;
using CrystalDecisions.ReportSource;
using CrystalDecisions.Shared;

namespace MBCA
{
publicpartialclassfrmMBCA : Form
    {
Random random = newRandom();
public frmMBCA()
        {
            InitializeComponent();
        }


privateint RandomNumber(int min, int max)
        {
```

```
return random.Next(min, max);
        }

privatevoid btn_Click(object sender, EventArgs e)
        {
if (opt1.Checked) { exp_WDM(); } else { exp_SDH(); }

        }


privatevoid exp_WDM()
        {

double[,] algoRes = newdouble[1000, 8]; int exp = 13;
int n = 100;
int      intMina      =      Convert.ToInt32(txtlower1.Text);      int      intMaxa      =
Convert.ToInt32(txtlower2.Text);
if (intMina > intMaxa)
            {
MessageBox.Show("Specify lower value for the lower bound", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);

            }
else
            {
int Hsub = 0, inComLsub = 0; int sumSources = 0, intCest = 0;
for (int j = 0; j < exp; j++)
                {
```

```
int intMin, intMax, intSources;
double avg = 0, variance = 0, SD = 0, Cest, Algo_applied_IPLR = 0, Algo_unApplied_IPLR =
0,
                        percent_Bandwidth = 0, percent_improved_IPLR = 0;
                    intMin      =      Convert.ToInt32(txtlower1.Text);      intMax      =
Convert.ToInt32(txtlower2.Text);

                    intSources = 35; // Convert.ToInt32(txtsize.Text);
int[] intArray = newint[35];
for (int i = 0; i < intSources; i++)
                    {
int rnd = RandomNumber(intMin, intMax); intArray[i] = rnd; sumSources = sumSources + rnd;
                    }
////////////////////////////////////////////////////// next coming trafic
if (Hsub == -1 || Hsub == 0)
                    {
                        sumSources = 0;
for (int i = 0; i < intSources; i++)
                        {
int rnd = RandomNumber(intMin, intMax); intArray[i] = rnd; sumSources = sumSources + rnd;
                        }

                        inComLsub = RandomNumber(sumSources - 500, sumSources + 500);
                        Hsub = RandomNumber(sumSources - 25000, sumSources - 15000);

//Hsub = RandomNumber(sumSources - 2000, sumSources - 4000);
//inComLsub = RandomNumber(sumSources + 2000, sumSources - 500);

                    }
```

```
                    avg = sumSources / intSources;
for (int k = 0; k < intSources; k++)
                    {
int val = intArray[k];
double diff = val - avg;
                        variance = variance + (diff * diff);
                    }

                    SD = variance / intSources;
                    SD = Math.Sqrt(SD);
                    Cest = avg + inComLsub + (4.072 * SD);
                    intCest = Convert.ToInt32(Cest);
                    Algo_applied_IPLR = Convert.ToDouble(inComLsub) / intCest;
                    Algo_unApplied_IPLR = Convert.ToDouble(inComLsub) / Hsub;
                    percent_improved_IPLR = ((Algo_unApplied_IPLR - Algo_applied_IPLR) *
100) / Algo_unApplied_IPLR;
                    percent_Bandwidth = ((intCest - Hsub) * 100) / Hsub;
if (intCest < 566666)
                    {
// MessageBox.Show((Hsub / intCest).ToString());
if (Algo_unApplied_IPLR <= 0.00011)
                        {
//admit packet
if (j == 0)
                            {
                                algoRes[j, 0] = sumSources; algoRes[j, 1] = Hsub;
algoRes[j, 2] = 0; algoRes[j, 3] = 0;
                                algoRes[j, 4] = 0; algoRes[j, 5] = 0; algoRes[j, 6] = 0;
algoRes[j, 7] = 1;
```

```
                            }
else
                            {
                                algoRes[j, 0] = inComLsub; algoRes[j, 1] = Hsub;
algoRes[j, 2] = intCest; algoRes[j, 3] = Algo_applied_IPLR;
                                algoRes[j, 4] = Algo_unApplied_IPLR; algoRes[j, 5] =
percent_improved_IPLR; algoRes[j, 6] = percent_Bandwidth; algoRes[j, 7] = 1;
                            }
                            Hsub = -1;
                        }
else
                        {
//////////////////////////////////////////////////Optimize the Higher Sub Layer
if (j == 0)
                            {
                                algoRes[j, 0] = sumSources; algoRes[j, 1] = Hsub;
algoRes[j, 2] = 0; algoRes[j, 3] = 0; algoRes[j, 4] = 0;
                                algoRes[j, 5] = 0; algoRes[j, 6] = 0; algoRes[j, 7] = 1;
                            }
else
                            {
do
                                {  // set Hsl = IntCest
                                    algoRes[n, 0] = inComLsub; algoRes[n, 1] = Hsub;
algoRes[n, 2] = intCest; algoRes[n, 3] = Algo_applied_IPLR;
                                    algoRes[n, 4] = Algo_unApplied_IPLR; algoRes[n, 5] =
percent_improved_IPLR; algoRes[n, 6] = percent_Bandwidth; algoRes[n, 7] = 1;
                                    n = n + 1; Hsub = intCest;
for (int i = 0; i < intSources; i++)
```

```
                                    {
int rnd = RandomNumber(intMin, intMax); intArray[i] = rnd; sumSources = sumSources + rnd;
                                    }
//////////////////////////////////////////////////////////////////////////
                                    avg = sumSources / intSources;
for (int k = 0; k < intSources; k++)
                                    {
int val = intArray[k];
double diff = val - avg;
                                        variance = variance + (diff * diff);
                                    }
                                    SD = variance / intSources; SD = Math.Sqrt(SD);
                                    Cest = avg + inComLsub + (4.072 * SD); intCest =
Convert.ToInt32(Cest);
                                    Algo_applied_IPLR = Convert.ToDouble(inComLsub) /
intCest;
                                    Algo_unApplied_IPLR = Convert.ToDouble(inComLsub) /
Hsub;
                                    percent_improved_IPLR = ((Algo_unApplied_IPLR -
Algo_applied_IPLR) * 100) / Algo_unApplied_IPLR;
                                    percent_Bandwidth = ((intCest - Hsub) * 100) / Hsub;
if (intCest > 566666)
                                    {
                                        algoRes[n, 0] = inComLsub; algoRes[n, 1] = Hsub;
algoRes[n, 2] = intCest; algoRes[n, 3] = 0;// Algo_applied_IPLR;
                                        algoRes[n, 4] = 0;// Algo_unApplied_IPLR;
                                        algoRes[n, 5] = 0;// percent_improved_IPLR;
                                        algoRes[n, 6] = 0;// percent_Bandwidth;
                                        algoRes[n, 7] = 1;
```

```
                                    n = n + 1;
break;
                                }

////////////////////////////////////////////////////////////////////////////
                                }
while (Algo_unApplied_IPLR > 0.0001);
                            }
                            Hsub = -1;
                        }
                    }
else
                    {
if (j == 0)
                        {
                            algoRes[j, 0] = sumSources; algoRes[j, 1] = Hsub; algoRes[j,
2] = 0; algoRes[j, 3] = 0; algoRes[j, 4] = 0;
                            algoRes[j, 5] = 0; algoRes[j, 6] = 0; algoRes[j, 7] = 1;
                        }
else
                        {
                            algoRes[j, 0] = inComLsub; algoRes[j, 1] = Hsub; algoRes[j,
2] = intCest; algoRes[j, 3] = 0;// Algo_applied_IPLR;
                            algoRes[j, 4] = 0;// Algo_unApplied_IPLR;
                            algoRes[j, 5] = 0;// percent_improved_IPLR;
                            algoRes[j, 6] = 0;// percent_Bandwidth;
                            algoRes[j, 7] = 1;
                        }
                        Hsub = -1;
```

```
                    }



                }

////////////////////////////////////////////////////////////////////////////////
                System.Data.DataTable dt = new System.Data.DataTable();
                dt.Columns.Add("ifHcInUcastPkts").DataType = typeof(Int32);
                dt.Columns.Add("ifHcOutUcastPkts").DataType = typeof(Int32);
                dt.Columns.Add("Computed_ifHcOutUcastPkts").DataType = typeof(Int32);
                dt.Columns.Add("IPLR").DataType = typeof(decimal);
                dt.Columns.Add("IPLRnt").DataType = typeof(decimal);
                dt.Columns.Add("Improved_IPLR").DataType = typeof(decimal);
                dt.Columns.Add("Bandwidth").DataType = typeof(decimal);
                dt.Columns.Add("IPLRbase").DataType = typeof(Int32);

////////////////////////////////////////////////////
int numb = algoRes.GetLength(0);
for (int j = 0; j < numb; j++)
                {
if (algoRes[j, 0] > 0)
                    {
DataRow row = dt.NewRow();
for (int i = 0; i < 8; i++)
                        {
                            row[i] = algoRes[j, i];
                        }
                        dt.Rows.Add(row);
                    }
```

```
                }
rep dsrep = newrep();
ReportDocument document = newReportDocument();
                dsrep.Tables["datarep"].Merge(dt);
                document.Load("res.rpt");
                document.SetDataSource((DataSet)dsrep);
                document.SetParameterValue("title", "SIMULATION RESULT FOR WDM SYSTEM");
                viewer.ReportSource = document;

            }

        }
privatevoid exp_SDH()
        {

double[,] algoRes = newdouble[1000, 8]; int exp = 13;
int n = 100;
int      intMina      =      Convert.ToInt32(txtlower1.Text);      int      intMaxa      =
Convert.ToInt32(txtlower2.Text);
if (intMina > intMaxa)
            {
MessageBox.Show("Specify lower value for the lower bound", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);

            }
else
            {
int Hsub=0, inComLsub=0; int sumSources = 0, intCest = 0;
for (int j = 0; j < exp; j++)
```

```
                {
int intMin, intMax, intSources;
double avg = 0, variance = 0, SD = 0, Cest, Algo_applied_IPLR = 0, Algo_unApplied_IPLR =
0,
                        percent_Bandwidth = 0, percent_improved_IPLR = 0;
                    intMin      =      Convert.ToInt32(txtlower1.Text);      intMax      =
Convert.ToInt32(txtlower2.Text);

                    intSources = 35; // Convert.ToInt32(txtsize.Text);
int[] intArray = newint[35];
for (int i = 0; i < intSources; i++)
                    {
int rnd = RandomNumber(intMin, intMax); intArray[i] = rnd; sumSources = sumSources + rnd;
                    }
////////////////////////////////////////////////////// next coming trafic
if (Hsub == -1 || Hsub == 0)
                    {
                        sumSources = 0;
for (int i = 0; i < intSources; i++)
                        {
int rnd = RandomNumber(intMin, intMax); intArray[i] = rnd; sumSources = sumSources + rnd;
                        }

                        Hsub = RandomNumber(sumSources + 2000, sumSources + 4000);
                        inComLsub = RandomNumber(sumSources - 2000, sumSources + 500);

                    }
                    avg = sumSources / intSources;
for (int k = 0; k < intSources; k++)
```

```
                    {
int val = intArray[k];
double diff = val - avg;
                        variance = variance + (diff * diff);
                    }

                    SD = variance / intSources;
                    SD = Math.Sqrt(SD);
                    Cest = avg + inComLsub + (4.072 * SD);
                    intCest = Convert.ToInt32(Cest);
                    Algo_applied_IPLR = Convert.ToDouble(inComLsub) / intCest;
                    Algo_unApplied_IPLR = Convert.ToDouble(inComLsub) / Hsub;
                    percent_improved_IPLR = ((Algo_unApplied_IPLR - Algo_applied_IPLR) *
100) / Algo_unApplied_IPLR;
                    percent_Bandwidth = ((intCest - Hsub) * 100) / Hsub;
if (intCest < 566666)
                    {
// MessageBox.Show((Hsub / intCest).ToString());
if (Algo_unApplied_IPLR <= 0.00011)
                        {
//admit packet
if (j == 0)
                            {
                                algoRes[j, 0] = sumSources; algoRes[j, 1] = Hsub;
algoRes[j, 2] = 0; algoRes[j, 3] = 0;
                                algoRes[j, 4] = 0; algoRes[j, 5] = 0; algoRes[j, 6] = 0;
algoRes[j, 7] = 1;
                            }
else
```

```
                            {
                                algoRes[j, 0] = inComLsub; algoRes[j, 1] = Hsub;
algoRes[j, 2] = intCest; algoRes[j, 3] = Algo_applied_IPLR;
                                algoRes[j, 4] = Algo_unApplied_IPLR; algoRes[j, 5] =
percent_improved_IPLR; algoRes[j, 6] = percent_Bandwidth; algoRes[j, 7] = 1;
                            }
                            Hsub = -1;
                        }
else
                        {
//////////////////////////////////////////////////Optimize the Higher Sub Layer
if (j == 0)
                            {
                                algoRes[j, 0] = sumSources; algoRes[j, 1] = Hsub;
algoRes[j, 2] = 0; algoRes[j, 3] = 0; algoRes[j, 4] = 0;
                                algoRes[j, 5] = 0; algoRes[j, 6] = 0; algoRes[j, 7] = 1;
                            }
else
                            {
do
                                {  // set Hsl = IntCest
                                    algoRes[n, 0] = inComLsub; algoRes[n, 1] = Hsub;
algoRes[n, 2] = intCest; algoRes[n, 3] = Algo_applied_IPLR;
                                    algoRes[n, 4] = Algo_unApplied_IPLR; algoRes[n, 5] =
percent_improved_IPLR; algoRes[n, 6] = percent_Bandwidth; algoRes[n, 7] = 1;
                                    n = n + 1; Hsub = intCest;
for (int i = 0; i < intSources; i++)
                                    {
int rnd = RandomNumber(intMin, intMax); intArray[i] = rnd; sumSources = sumSources + rnd;
```

```
                                    }
////////////////////////////////////////////////////////////////////////////
                                    avg = sumSources / intSources;
for (int k = 0; k < intSources; k++)
                                    {
int val = intArray[k];
double diff = val - avg;
                                        variance = variance + (diff * diff);
                                    }
                                    SD = variance / intSources; SD = Math.Sqrt(SD);
                                    Cest = avg + inComLsub + (4.072 * SD); intCest =
Convert.ToInt32(Cest);
                                    Algo_applied_IPLR = Convert.ToDouble(inComLsub) /
intCest;
                                    Algo_unApplied_IPLR = Convert.ToDouble(inComLsub) /
Hsub;
                                    percent_improved_IPLR = ((Algo_unApplied_IPLR -
Algo_applied_IPLR) * 100) / Algo_unApplied_IPLR;
                                    percent_Bandwidth = ((intCest - Hsub) * 100) / Hsub;
if (intCest > 566666)
                                    {
                                        algoRes[n, 0] = inComLsub; algoRes[n, 1] = Hsub;
algoRes[n, 2] = intCest; algoRes[n, 3] = 0;// Algo_applied_IPLR;
                                        algoRes[n, 4] = 0;// Algo_unApplied_IPLR;
                                        algoRes[n, 5] = 0;// percent_improved_IPLR;
                                        algoRes[n, 6] = 0;// percent_Bandwidth;
                                        algoRes[n, 7] = 1;
                                        n = n + 1;
break;
```

```
                                    }

////////////////////////////////////////////////////////////////////////////////
                                        }
while (Algo_unApplied_IPLR > 0.0001);
                                    }
                                    Hsub = -1;
                                }
                            }
else
                            {
if (j == 0)
                                {
                                    algoRes[j, 0] = sumSources; algoRes[j, 1] = Hsub; algoRes[j,
2] = 0; algoRes[j, 3] = 0; algoRes[j, 4] = 0;
                                    algoRes[j, 5] = 0; algoRes[j, 6] = 0; algoRes[j, 7] = 1;
                                }
else
                                {
                                    algoRes[j, 0] = inComLsub; algoRes[j, 1] = Hsub; algoRes[j,
2] = intCest; algoRes[j, 3] = 0;// Algo_applied_IPLR;
                                    algoRes[j, 4] = 0;// Algo_unApplied_IPLR;
                                    algoRes[j, 5] = 0;// percent_improved_IPLR;
                                    algoRes[j, 6] = 0;// percent_Bandwidth;
                                    algoRes[j, 7] = 1;
                                }
                                Hsub = -1;
                            }
```

```
                }

////////////////////////////////////////////////////////////////////////////
                System.Data.DataTable dt = new System.Data.DataTable();
                dt.Columns.Add("ifHcInUcastPkts").DataType = typeof(Int32);
                dt.Columns.Add("ifHcOutUcastPkts").DataType = typeof(Int32);
                dt.Columns.Add("Computed_ifHcOutUcastPkts").DataType = typeof(Int32);
                dt.Columns.Add("IPLR").DataType = typeof(decimal);
                dt.Columns.Add("IPLRnt").DataType = typeof(decimal);
                dt.Columns.Add("Improved_IPLR").DataType = typeof(decimal);
                dt.Columns.Add("Bandwidth").DataType = typeof(decimal);
                dt.Columns.Add("IPLRbase").DataType = typeof(Int32);

////////////////////////////////////////////////////
int numb = algoRes.GetLength(0);
for (int j = 0; j < numb; j++)
                {
if (algoRes[j, 0] > 0)
                    {
DataRow row = dt.NewRow();
for (int i = 0; i < 8; i++)
                        {
                            row[i] = algoRes[j, i];
                        }
                        dt.Rows.Add(row);
                    }
                }
rep dsrep = newrep();
```

```
ReportDocument document = newReportDocument();
                dsrep.Tables["datarep"].Merge(dt);
                document.Load("res.rpt");
                document.SetDataSource((DataSet)dsrep);
                document.SetParameterValue("title", "SIMULATION RESULT FOR SDH SYSTEM");
                viewer.ReportSource = document;

            }

        }

privatevoid frmMBCA_Load(object sender, EventArgs e)
        {



        }

privatevoid frmMBCA_SizeChanged(object sender, EventArgs e)
        {
            viewer.Width = base.Width - 50; viewer.Height = base.Height - 50;
            groupBox1.Width = base.Width - 40; groupBox1.Height = base.Height - 40;
        }


    }
}
```

APPENDIX C

Table C1: Simulated results for WDM System

Week	ifHcInUcastPkts (Pkt/s)	Bandwidth utilization (Pkt/s)	IPLR
1	390,660	0	0
2	390,869	526,073	0.742993843
3	390,956	526,103	0.743116842
4	390,750	525,927	0.742973835
5	390,695	525,844	0.742986513
6	391,365	526,545	0.743269806
7	391,017	526,175	0.743131088
8	390,804	525,968	0.743018587
9	390,441	525,611	0.742832627
10	390,757	525,890	0.743039419
11	390,574	525,706	0.742951383
12	391,014	526,132	0.743186121
13	390,487	525,677	0.742826869

Table C2: Table C1: Simulated results for SDH System

Week	ifHcInUcastPkts (Pkt/s)	Bandwidth utilization (Pkt/s)	IPLR
1	390,779	0	0
2	390,396	525,541	0.742845944
3	389,141	524,271	0.742251622
4	389,932	525,047	0.742661133

5	390,578	525,711	0.742951926
6	389,318	524,454	0.742330119
7	390,529	525,671	0.742915245
8	391,023	526,166	0.743155202
9	390,099	525,189	0.742778314
10	388,968	524,114	0.742143885
11	390,551	525,705	0.742909046
12	390,614	525,829	0.742853665
13	388,861	523,974	0.742137969