



香港城市大學
City University
of Hong Kong

Department of Electronic Engineering

FINAL YEAR PROJECT REPORT

BEngIE-2009/10-MZ-02

Performance Evaluation of Optical Burst Switching Networks

Student Name: PENG Yu

Student ID:

Supervisor: Prof. ZUKERMAN, Moshe

Assessor: Dr. Eric Wong, W M

Bachelor of Engineering (Honours) in
Information Engineering

Student Final Year Project Declaration

I have read the student handbook and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I declare that the work submitted for the final year project does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Student Handbook.

Project Title : Performance Evaluation of Optical Burst Switching Networks

Student Name : PENG Yu

Student ID:

Signature:

Date : 19/04/2010

No part of this report may be reproduced, stored in a retrieval system, or transcribed in any form or by any means – electronic, mechanical, photocopying, recording or otherwise – without the prior written permission of City University of Hong Kong.

TABLE OF CONTENTS

1.	ABSTRACT	- 1 -
2.	ACKNOWLEDGMENTS.....	- 2 -
3.	OBJECTIVES	- 3 -
4.	INTRODUCTION	- 4 -
5.	BACKGROUND STUDY	- 7 -
5.1	OBS Fundamentals	- 7 -
5.2	OBS Assembly Algorithm.....	- 9 -
5.3	OBS Burst Reservation Protocols	- 12 -
5.4	OBS Contention Resolution Schemes.....	- 14 -
5.5	Long Range Dependent Traffic Model	- 16 -
5.6	Pareto Distribution.....	- 19 -
5.7	TCP over OBS	- 20 -
5.8	NS2 Simulator	- 21 -
6.	METHODOLOGY	- 22 -
6.1	Network Topology	- 23 -
6.2	Modeling.....	- 26 -
6.2.1	Open Loop Model	- 26 -
6.2.2	TCP Model.....	- 28 -
6.2.3	Combined Close Loop Model.....	- 29 -
6.3	Simulation.....	- 29 -
6.3.1	Simulation for Open Loop Modal.....	- 29 -
6.3.2	Simulation for Closed Loop Model.....	- 32 -
7.	RESULT AND DISCUSSION	- 37 -

7.1	Open Loop Validation.....	- 37 -
7.2	Closed Loop Validation.....	- 42 -
7.3	Approximation for Large Scale Network.....	- 43 -
8.	CONCLUSION	- 45 -
9.	APPENDIX.....	viii
	Appendix I: open loop simulation details	viii
	Appendix II: Pareto traffic generator	x
10.	GLOSSARY	xii
11.	REFERENCES.....	xiv

LIST OF FIGURES

Figure 1 Burst Assembly ^[9]	- 7 -
Figure 2 Burst Disassembly ^[9]	- 7 -
Figure 3 Logical flow of open loop simulation	- 11 -
Figure 4 Control signalling and resource reservation ^[9]	- 12 -
Figure 5 JET reservation scheme ^[9]	- 13 -
Figure 6 Number of Poisson arrivals under different time scaling ^[20]	- 17 -
Figure 7 Number of arrivals of self-similar traffic across time scaling ^[20]	- 18 -
Figure 8 An optical network topology with a bottleneck DWDM switch	- 23 -
Figure 9 ON/OFF model of TCP sources	- 24 -
Figure 10 Cross section of the same network topology in Fig. 8	- 25 -
Figure 11 Interconnected TCP model and open-loop model	- 26 -
Figure 12 User interface for open loop simulation - input.....	- 31 -
Figure 13 User interface for open loop simulation - result.....	- 32 -
Figure 14 Raw data from close loop simulation	- 34 -
Figure 15 Report for closed loop simulation	- 34 -
Figure 16 Animation for closed loop simulation	- 35 -
Figure 17 Animation for closed loop simulation	- 36 -
Figure 18 Open Loop simulation with 2 input ports , 1 out put port and buffer size equals to 0	- 37 -
Figure 19 Open Loop simulation with 2 input ports , 1 out put port and buffer size equals to 10	- 38 -
Figure 20 Open Loop simulation with 80 input links , 20 output links and buffer size equals to 0	- 39 -

Figure 21 Open Loop simulation with 80 input links , 20 output links and buffer size equals to 10	- 39 -
Figure 22 Open loop simulation accross different value of shape parameter	- 41 -
Figure 23 Closed loop validation.....	- 42 -
Figure 24 Bottleneck utilization approximation	- 44 -
Figure 25 Open loop simulation logical flow - CQueue.....	ix
Figure 26 Open loop simulation logical flow - Main function	x

LIST OF TABLES

Table 1 Comparsion between simulations with 80 input links, 20 output links, buffer size equals to	40 -
---	------

1. ABSTRACT

Optical Burst Switching (OBS) is a promising switching scheme for wavelength division multiplexing (WDM) networks, which achieves the balance between Optical Packet Switch (OPS) and Optical Circuit Switching (OCS). During the past decade, numbers of papers have studied the performance of OBS. However, most of the analyses in those papers did not consider the existence of self-similarity. Compare with the traffic measurement, the burstiness of Internet traffic has been underestimated in many queuing models, such as Poisson process. This project investigates the performance of TCP over OBS by focusing on the buffer requirement at a bottleneck WDM switch without neglecting the traffic self-similarity. In particular, this project consists of both qualitative and quantitative parts. In the qualitative part, the background about OBS and Long Range Dependency (LRD) traffic has been studied and discussed. In the quantitative part, a two-part analytical model proposed by Wong et al. is used, one open loop model to approximate the packet loss rate at bottleneck and another closed loop model to evaluate the performance with TCP. Based on the analytical model, two simulations are implemented in C++ and ns2 respectively to investigate the buffer requirement in the TCP over OBS.

2. ACKNOWLEDGMENTS

I would like to thank for everyone who helped me with my final year project, especially for Prof. ZUKERMAN, Dr. Eric Wong, Tony Cui, Li Ji, Li Fan, and my dear final year project teammates.

3. OBJECTIVES

- To get an insight into the Optical Burst Switching (OBS) network.
- To evaluate the TCP performance over OBS with heavy tailed flows.
- To investigate the buffer requirement at the bottleneck core switch in OBS network.

4. INTRODUCTION

The exponential growth of the Internet traffic today leads to the dramatically increasing demand on the network bandwidth. To meet this increasing demand, it is widely accepted that optical fibre will be the replacement of metal cables in the information communication in the near future, since optical fibre enables signal transmission at very high bit rate. Moreover, optical fibre is highly suitable for long distance communication, because light propagates with much less attenuation through optical fibre compared with signal through electrical cables. However, the potential of optical fibre has not been fully realized until the wavelength division multiplexing (WDM) ^[1] was introduced. WDM is a technology which enables multiple optical signals transmitted through a single optical fibre simultaneously by assigning different wavelengths to each signal ^[2]. With WDM technology, the total effective speed of a single optical fibre can be hundreds gigabits per second (Gbps).

There are mainly three switching schemes that have been proposed for WDM network, namely Optical Circuit Switching (OCS), Optical Packet Switching (OPS) and Optical Burst Switching (OBS) ^[3]. In OCS network, an end-to-end light path is reserved in advance for every connection. As a result, there is not much transmission delay and no loss in the network. However, both the utilization of communication channel is low ^[3, 4] and it also lacks of

flexibility. Moreover, when the connection holding time is short, the overhead for connection set-up and release leads to poor performance^[2]. In OPS network, network traffic is carried by packets. In this case, no end-to-end connection reservation is required, so the utilization of communication channel is high. Since no dedicated light path is reserved in this scheme, the header for routing purpose will be electrical while data are forwarded in optical domain. In this case, the complexity due to the electrical and optical conversion will be very high, and it may require expensive fibre delay line. To achieve the benefits from both OPS and OCS meanwhile avoiding their drawbacks, OBS was introduced to the WDM network. In OBS network, traffic is carried by burst, which is aggregated by number of packets arriving at the same edge router with the same destination^[6, 7]. A burst control packet is sent in advance to reserve the wavelength at each hop for each data burst. Since the wavelength reservation is only for each burst, which is not a whole end-to-end connection, the utilization of the communication channel is high in OBS^[3]. Moreover, since numbers of packets are aggregate into one burst with one control message, it no longer need to do the table look up for each packet at each hope, which significantly reduce the complexity of the network. Those are the reasons why OBS is becoming one promising switching scheme for future WDM network.

Since OBS has become a quite hot research field these years, there are numbers of papers studying about the performance of OBS network. However, most of the analyses in previous papers applied queuing models

such as Poisson model, which underestimates the traffic self-similarity. Based on the research of recent year, many studies showed that the Internet traffic exhibits self-similarity, which is due to the flow size in the Internet being heavy-tailed. Without considering the traffic self-similarity the analyses done before are not accurate, some may have big differences from the real situation.

According to the Cisco White Paper, “The sum of all forms of video (TV, VoD, Internet, and P2P) will account for close to 90 percent of consumer traffic by 2012.”^[8] Although the traffic generated by the application using User Datagram Protocol (UDP), such as streaming media, has increased a lot in the last few year; Transmission Control Protocol (TCP) based applications, which need reliable connection-oriented end-to-end transport service, such as world wide web, email, peer-to-peer file sharing, still lead to a huge amount of traffic in the Internet. That is why understanding and evaluating the performance of TCP over OBS is critical and important.

In this project, we will get an insight into OBS network and evaluate the performance of TCP over OBS network with heavy-tailed flows. By investigating the TCP Newreno, this project demonstrate that with sufficient number of wavelength channel in the output fibre, a high throughput could be achieved at a bottleneck core switch with few or even no buffers.

5. BACKGROUND STUDY

5.1 OBS Fundamentals

OBS network consists of two types of node, namely edge nodes and core nodes. Edge nodes are responsible for the aggregation of packets and dissembling of burst while core nodes are in charge of burst switching and control information updating.

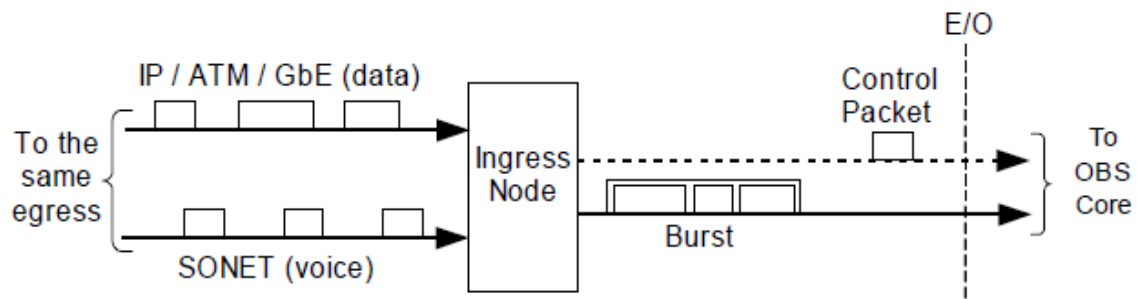


Figure 1 Burst Assembly^[9]

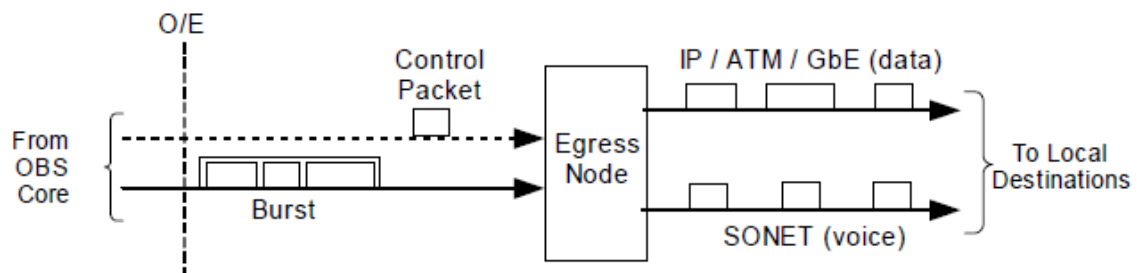


Figure 2 Burst Disassembly^[9]

In the OBS network, as you can see from Figure 1 and Figure 2, various data packets from different resources (e.g. IP, ATM) with the same destination are aggregated into a burst at the ingress router, which later will be disassembled at egress router. Each burst will be switched and routed at core node individually. For each burst, the control signaling is done by a separate Burst Control Packet (BCP) contains burst information including burst length information, destination address and QoS requirement, which will be transmitted through a dedicated control channel, which will be sent in advance of the transmission of the burst to reserve the resource at the next hop. The data burst will follow the control packet without waiting for the acknowledgement. If the required resource is available, the burst will be forwarded to the next hop transparently. Otherwise, the whole burst or a fraction of it will be dropped.

Note that OBS is in some way a combination of OCS and OPS, which achieve the benefits from both OPS and OCS while avoiding their shortcomings.

There are two main aspects that OBS differs from OCS ^[9]. First, the resource reservation in OBS is not end-to-end reservation; a burst will be forwarded after its control packet without waiting for the reservation acknowledgement. A burst may not complete its route and be blocked at half

way. In OCS, a dedicate end-to-end path is reserved before the transmission start, which guarantees there is no loss during transmission. Second, in OBS, the resources are reserved for the duration which is needed for only switching and transmission of each burst while in OCS, the resources are available for the entire duration of end-to-end transmission. Thus OCS achieves very low utilization due to the overhead for connection establishment and release when the connection holding time is very short.

OBS also differs from OPS two three main aspects. First, instead of the in-band signalling in OPS, OBS performs out-band signalling. In OPS, the control information is contained in the header which transmitted with the data information together. Second, OBS burst are not buffered at core node, but packets in OPS are. Thus the complexity of OPS network will be higher than OBS due to the buffering and table look up for every packet at core nodes.

5.2 OBS Assembly Algorithm

As mentioned above, burst assembly in OBS is the process that aggregate different packets from upper layer, such as IP and ATM, into bursts at the edge router of OBS network. There is a burst scheduler in charge of the generation of burst and its corresponding burst control packets. According to the previous study, there are mainly three types of assembly algorithms,

namely time-based, burstlength-based and mixed time-burstlength-based ^{[11,}
^{12]}.

For the time-based scheme, all the bursts are assembled according to a pre-set timer and it restarts every time when a new burst is generated. In this case, the value of time T should be decided with consideration. It should not be too large or too small. If the pre-set value is too large, the queuing delay for all buffered packet will be intolerable. Conversely, if the time interval set to be too small, too many small burst will result in higher complexity in the network, which will become more likely to be OPS. Thus the time-based scheme will result in undesirable burst length.

For the burstlength-based scheme, a threshold is introduced to restrict the minimum burst length. When a packet arrives at the edge router which leads to sizes of all currently stored packets exceeds minimum burst length. In this case, there is no guarantee on the delay of packets in the network.

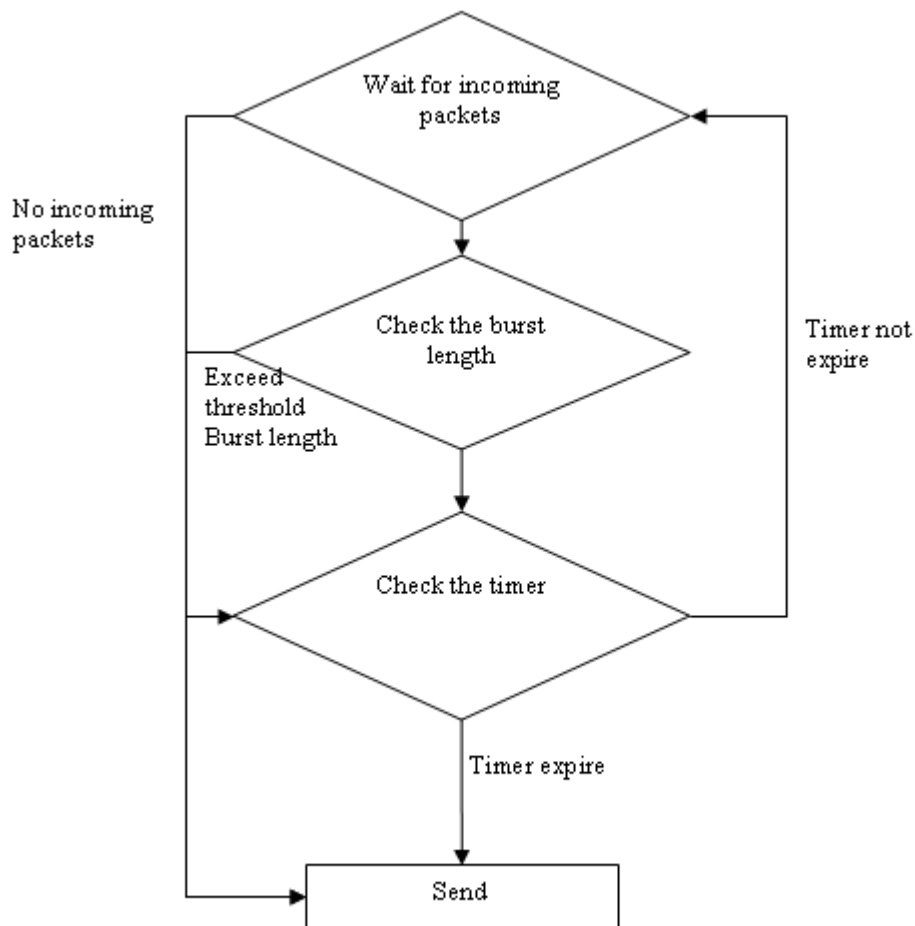


Figure 3 Logical flow of open loop simulation

To optimize the performance of both schemes in OBS, a mixed time-burstlength-based scheme was proposed, as can be seen in Figure 3. A burst is generated either a pre-set timer is expired or the burst length exceeds the desirable threshold. However, there is another question, during the time period after a burst control packet is sent and before the burst is forward, the packets may arrive continually, aggregating those packets into the same burst will not be reasonable since the resources need has already been made, while leaving those packets for next burst will result in increase of average delay. A

burst length prediction maybe applied to minimize the overall delay in the OBS network ^[9].

5.3 OBS Burst Reservation Protocols

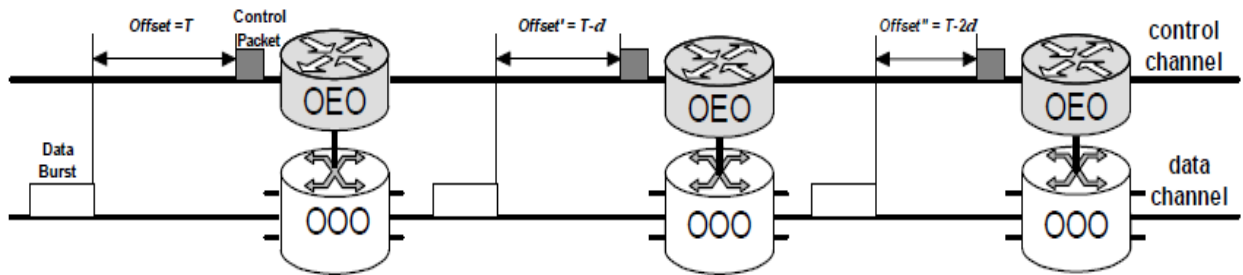


Figure 4 Control signalling and resource reservation^[9]

For the purpose of forwarding bursts over the OBS network, a resource reservation protocol should be applied to ensure all the resources needed are well reserved before the corresponding data burst arrives at the next hop. The basic idea about the one-way resource reservation in OBS network is showed in Figure 4. The offset time T is the time interval between BCP and corresponding burst and which will be updated after passing each core nodes.

There are several reservation schemes been previously proposed, such as Just-In-Time (JIT), Just-Enough-Time (JET), Horizon schemes [6, 13-16]. In this project, I will mainly compare the JIT and JET scheme here.

JIT scheme is considered to be a scheme based on Tell-And-Wait (TAW) in ATM network, which means if the output wavelength is available, it will be immediately reserved when a reservation request arrives. This scheme is very simple, whereby the core node is not aware of the burst length, which is considered to be its biggest advantage. However, during the offset time between the control packet and data burst, although the reserved output wavelength is free, it cannot be used for other purpose, which results in low efficiency.

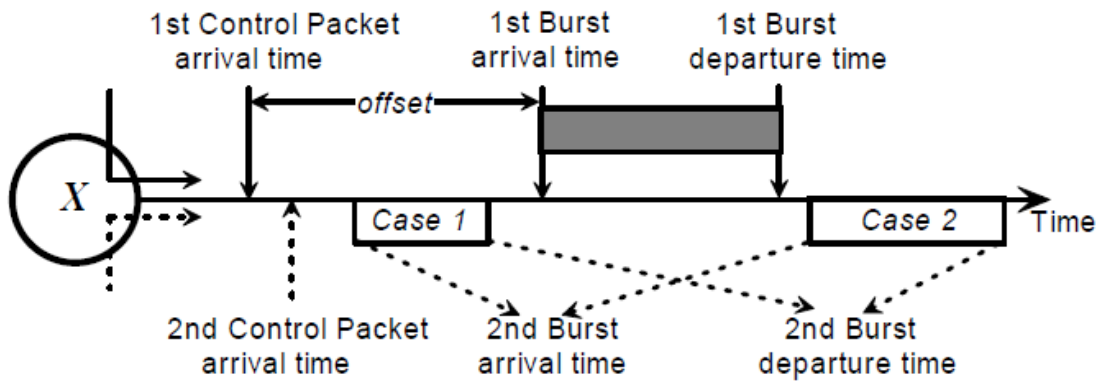


Figure 5 JET reservation scheme^[9]

In the JET scheme, a burst control packet reserves an output channel for a period of time equal to the burst length, starting at the expected burst arrival time. The JET scheme is believed that has a higher efficiency, since during

the offset time, the wavelength can also use for switching of other data burst, as can be seen at Figure.4. However, the drawback of this JET is this scheme is quite complex, since every node need to know the arrival time of the burst and the duration of its transmission, and those information need to be updated at each hop.

In the comparison in [16], the simplicity of JIT scheme may balance its shortcomings of low efficiency. Compare with other protocols, hardware implementation of JIT has already been realized.

5.4 OBS Contention Resolution Schemes

By applying the one way reservation protocols, the nodes in OBS network forward burst without receiving the reservation acknowledgements. It is necessary to have resource contention resolution to handle switching contention among burst. In the OBS network, contention is handled based store-and-forward techniques, which requires buffer in the core node or wavelength conversion technique. The scheme designed to resolve contention in several domains ^[6]:

a) Wavelength domain

By applying wavelength conversion, burst can be transmitted to a different wavelength channel when the original one is not available. In this case, no burst will be delayed unless all the output wavelength

channels are busy. However it is expensive, since a wavelength converter is needed at each hop in OBS.

b) Space domain

The solution under space domain also referred as deflection routing, a data burst can be route to another output wavelength and then switch to an alternate path to the destination. This scheme is relatively efficient it reduces the requirement of buffer in the core node. However, the bursts may arrive at destination out of order since the number of hops experienced by each burst may not equal.

c) Time domain

Contention resolution scheme under time domain can be achieved by using Fiber Delay Line (FDLs). A burst can be delayed for certain period of time when there is a contention in the network. FDLs are very expensive and can only delay the burst for fixed time for a given length. Moreover, FDLs cannot have random access compare with the electronic buffers.

d) Burst Segmentation

The burst segmentation tries to minimize the data discarded by rescheduling or dropping portions of the burst. In preemption-based scheme, the conflicting bursts will be segmented based on the burst priority. Although burst segmentation can achieve high efficiency and low burst loss, it requires complex algorithms and regeneration of the burst control packet will increase the complexity of signalling.

5.5 Long Range Dependent Traffic Model

Based on the traffic measurements, one of the important characteristics of today's Internet traffic is it exhibits Long Range Dependence (LRD) and self-similarity ^[17]. Those previous analyses without consider the LRD and self-similarity maybe not accurate, since the LRD and self-similarity increase the data loss and delay, and decrease the network utilization. The observation also showed that the self-similarity is caused by the flow sizes on the Internet are heavy-tailed ^[19], a small proportion of the very long burst carries a large proportion of traffic.

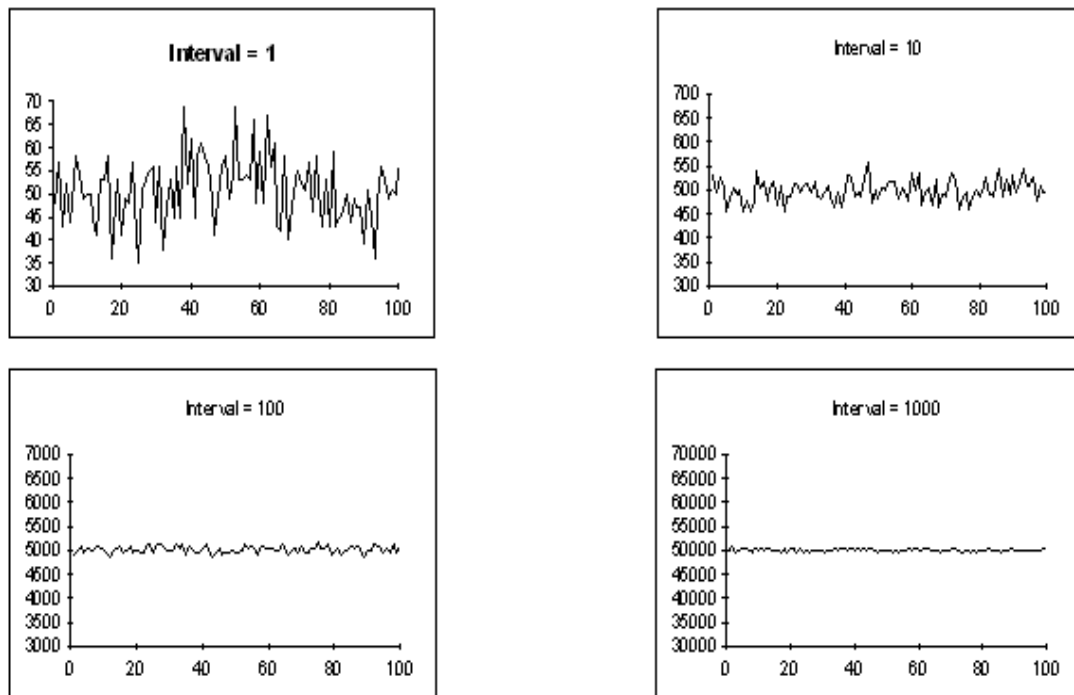


Figure 6 Number of Poisson arrivals under different time scaling^[20]

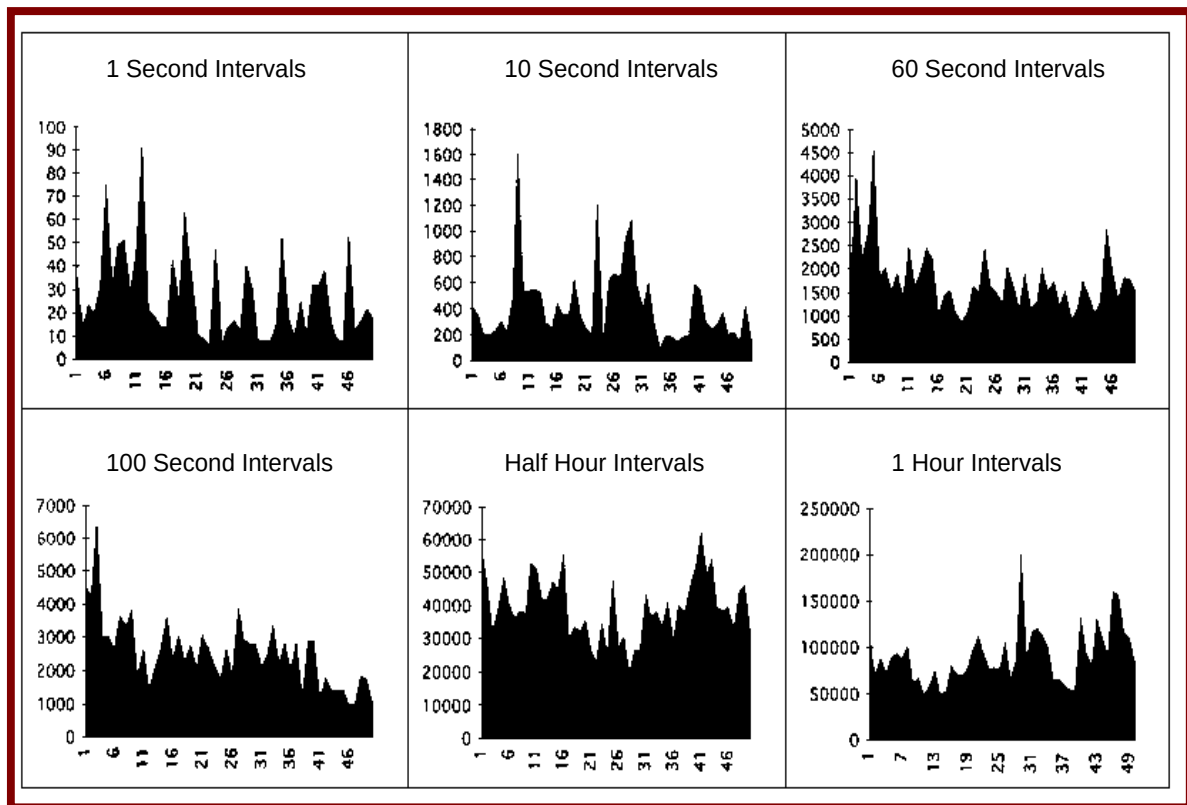


Figure 7 Number of arrivals of self-similar traffic across time scaling ^[20]

The explanation of traffic self-similarity is, the traffic that exhibits the same property at different levels of time scaling ^[18]. As you can see the Poisson model has been widely used before from Figure 6, which shows the number of arrivals within one time interval. When the time interval becomes larger, the burstiness of the traffic becomes smaller and smaller. However, in reality, the Internet traffic exhibits self-similarity, which is illustrated by Figure 7. No matter how large the time interval is, the burstiness of data remains.

The comparison showed the burstiness of Poisson processes decreasing when the time scale becomes larger. The burstiness of Internet traffic remains

across different time intervals, which do not agree with the Poisson model. However, the simulation about the self-similar traffic is always considered to be a challenge, because it is not accurate for a simulation with finite time to reflect the effects due to a random variable with infinite variance.

5.6 Pareto Distribution

It has been shown that self-similar or LRD traffic can be generated by several sources of Pareto distributed ON and OFF periods. The heavy-tailed flows are generated in the ON periods while the sources are silent in OFF periods [20].

The Pareto distribution is a power law distribution. It has two parameters as follows, the shape parameter α and the minimal value of random variable β . Following is the cumulative distribution function of Pareto distribution.

$$F(x) = 1 - (\beta / x)^\alpha \quad (1)$$

And the probability density function

$$\begin{cases} f(x) = \alpha (\beta / x)^{\alpha+1} / \beta, \text{ for } x > \beta, \alpha > 0 \\ f(x) = F(x) = 0, \text{ for } x \leq \beta \end{cases} \quad (2)$$

The shape parameter decides the mean and the variance of the random variable. When $\alpha \leq 1$, the mean value of the distribution is infinite; when $1 < \alpha \leq 2$, the variance is infinite while the mean is a constant; when $\alpha > 2$, both mean and variance are finite. For the self-similar traffic, since it has constant mean and infinite variance, α should stay between 1 and 2.

5.7 TCP over OBS

Although it is widely believed that, most of the traffic in the Internet in the future will be transmitted under UDP, such as media stream, grid computing, the TCP is still the most widely used protocol for reliable data transmission in today's Internet and will be in the near future. TCP-based application, such as world wide web, email, peer-to-peer file sharing, still account for a huge amount of data in the Internet. Thus understanding the performance of TCP over OBS is critical.

The sending rate of a TCP source is sensitive to the packet loss in the network. The sender will slow down when there is a high rate of packet loss, which avoids the congestion of network due to too many resending packets. OBS networks have its own unique characteristics that affect the TCP throughput. For example, the burst assembly delay will increase the Round Trip Time (RTT) which results in the decrease of TCP throughput^[21]. Since the transmission unit is enlarged from packet to burst, when there is a burst contain packets from different TCP sources lost, multiple TCP will adapt

their rate of transmission. Thus the characteristics of OBS will have an obvious impact on TCP performance.

5.8 NS2 Simulator

Ns2 is an object-oriented network simulator for discrete event, which is widely used at networking research for its good performance and flexibility for simulation of various protocols over wired or wireless network. However, as a discrete event simulator, Ns2 takes care of each packet in the network, which makes simulations under Ns2 consume a lot of computer resources, such as CPU time and computer memory. For simulation of small-scale network, the Ns2 may be a good simulator. But for large scale network, the simulation may be computational prohibitive.

6. METHODOLOGY

In the past, in order to achieve high throughput, it had been widely accepted that we need large buffers at core switch. However, large buffers lead to long queue in the network which enlarges the average delay. On the other hand, buffering is very expensive in optical future network, since we can only do it by FDL or O-E-O conversion at each core node. Thus it is advantageous to find ways to reduce queuing delay and still achieve high throughput.

A queuing model proposed by Dr. Eric Wong and Prof. Zukerman ^[22] indicate that there is only small maybe non-zero buffer needed when there are numbers of wavelength in the switch with wavelength conversion applied. Later, Eric W.M. Wong et al. ^[23] showed the interaction between buffers requirement and number of wavelength channels with Poisson model. And the result in this paper demonstrated that when there is sufficient number of wavelength at DWDM switch, TCP over OBS can achieve 90% utilization when there are 10 packets per output port and 70% utilization with zero buffers. In this project, I used the same network topology and analytical model in ^[23] to investigate the buffer requirement at bottleneck core switch with heavy-tailed flows. In order to show the same analytical model is also applicable to heavy-tailed flows, I implemented both “open loop” and “close loop” simulation to validate the modelling.

6.1 Network Topology

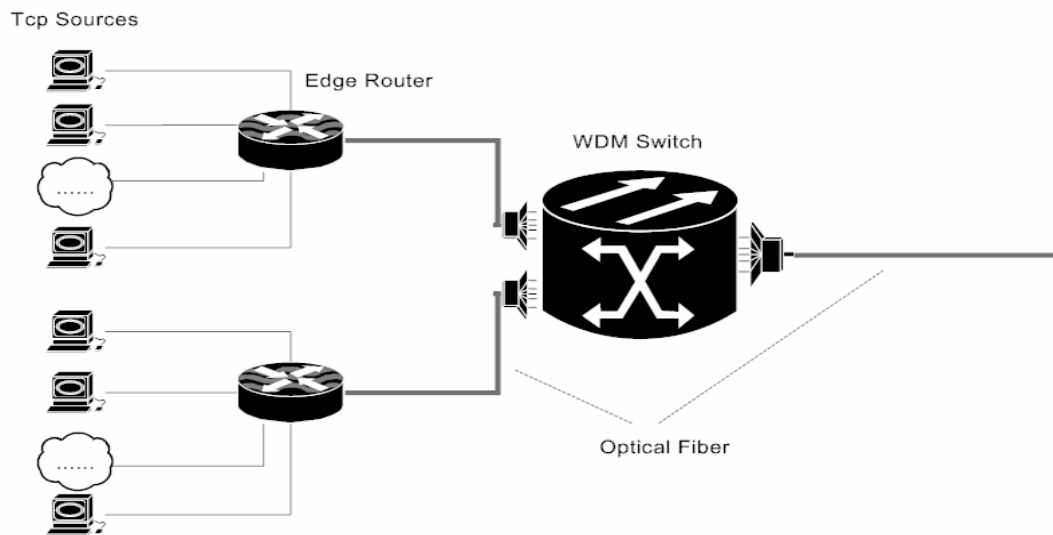


Figure 8 An optical network topology with a bottleneck DWDM switch

I considered using the same network topology in ^[23]. There are numbers of TCP sources transmit packets through access link to an OBS edge router, where the packets head to the same destination are aggregated into bursts. The edge router is considered to have large enough electronic buffer which leads to negligible probability of overflowing. And this assumption is quite reasonable in optical networks, since the electronic buffers will be quite cheap in the future. From the edge router, packets are aggregated into burst and forward to a DWDM switch through input optical fibre. Moreover, the DWDM switch here is considered to be a non-blocking switch, and all the buffers are located at output ports, which can be achieved by using a group of FIFO delay lines.

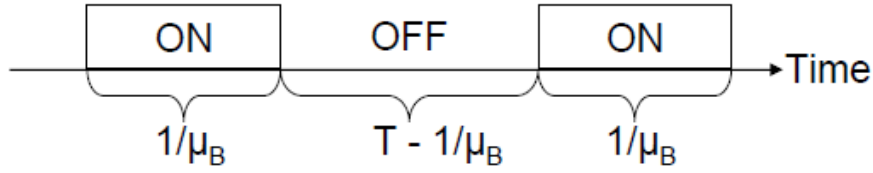


Figure 9 ON/OFF model of TCP sources

In order to generate heavy-tailed flows, I considered each TCP sources generates packets following a ON/OFF period shown in Figure 9., and the number of packets transmitted during ON period and the length of OFF period are taken from Pareto distribution with shape parameter $1 < \alpha \leq 2$. If it is not mentioned, the value of shape α parameter equals to 2. TCP sources are also assumed to always have packets to send here and the sender and receiver buffers for TCP sources are assumed to be large enough. Moreover, and there this no limitation about the TCP window size for each source, and which is also reasonable for future network.

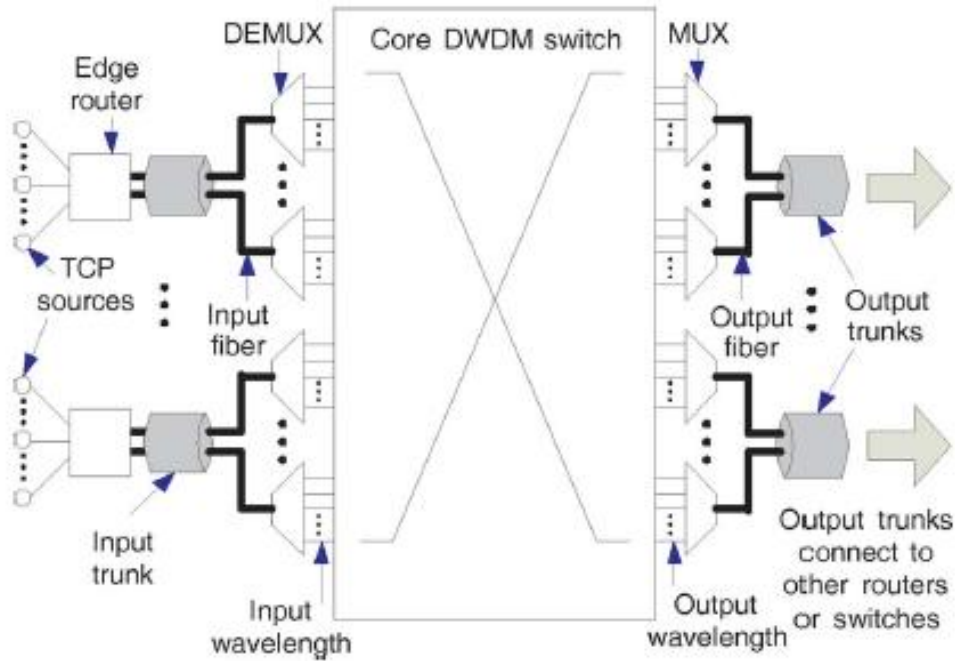


Figure 10 Cross section of the same network topology in Fig. 8

In the later part, the following terminology and notation will be used. And you can also refer those notations with a more detailed topology diagram in Figure 10 above. An optical fiber connecting two nodes called a trunk. There are multiple links, refer as wavelengths as well, in each truck. And every input and output link connects to an input port and output port at switch respectively. Assume that the total number of TCP sources is larger than the number of wavelength in a single input truck, and the total number of input links in all input trunks is larger or equals to the number of output wavelengths in a single output trunk.

For simplicity, this project will only focus on a single bottleneck output trunk, which all the TCP sources will transmit packets to it.

6.2 Modeling

In this project, the same analytical model in [23] will be used and validated. The analytical model of this project consists of two parts: One open loop approximation model to get the packet loss probability with input of traffic load; one closed loop fixed-point to reflect the impact of TCP. The closed loop is a model combined with interconnected open loop queuing model and a standard TCP Model [24]. And the open loop model can be applied independently to the non-TCP network. The relationship between them can be showed in the following figure.

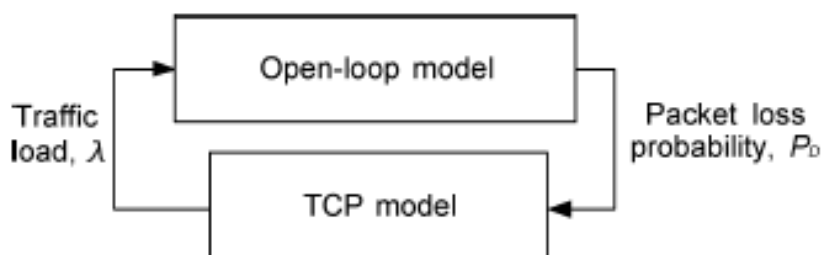


Figure 11 Interconnected TCP model and open-loop model

Following are some parameters used in the “open loop” queuing model and TCP models. λ refers as mean arrival rate, and $1/\lambda$ is the mean idle time between two consecutive packets. μ refers as mean service rate, and $1/\mu$ is the mean packet service time. N is the total number of TCP flows. M is the total number of incoming wavelengths. B represents the buffer size in terms of number packets. P_D is the long-term average packet loss probability.

6.2.1 Open Loop Model

As mentioned above, the open loop model will estimates the packet loss probability P_D by taking traffic load λ as input with parameters μ , M , K , B . As showed in [23], the Packet Engset with Buffer model (PEB) will be a more accurate model to do approximations, the detail comparison among several different models can be also found in [23]. In this project, the PEB model will also be used for loss probability approximation. PEB model is a single dimension Markov chain ^[23]. In the single dimension Markov chain, the state i equals to the number of packets in buffer or being serving. p_i is the probability in state i .

However, the single dimension Markov chain cannot monitor the discarding packets, which allows the packets to arrive immediately after the discard packet arrives. To make it more accurate, the more accurate mean idle time can be obtained from

$$\frac{1}{\lambda^*} = \frac{1}{\underline{\lambda}^*(\lambda, P_D)} \equiv \frac{1}{\lambda} + \frac{P_D}{\mu}. \quad (3)$$

From [23], we can get the packet loss probability over total arrival rate is

$$P_D = \frac{(M - K)\lambda^* p_{K+B}}{\Lambda}. \quad (4)$$

The probability of being at P_{K+B} can be achieved by solving the steady equation

$$\frac{p_{i+1}}{p_i} = \begin{cases} (M-i)\frac{\lambda^*}{(i+1)\mu} & 0 \leq i \leq K-1 \\ (M-K)\frac{\lambda^*}{K\mu} & K \leq i \leq K+B-1 \end{cases} \quad (5)$$

and the normalization equation

$$\sum_{i=0}^{K+B} p_i = 1. \quad (6)$$

6.2.2 TCP Model

According to the [24], the sending rate r of TCP source (packets/second) can be obtained from

$$r = \frac{\sqrt{1.5/P_D}}{RTT} \quad (7)$$

with parameter of end-to-end Round Trip Time.

For multiple TCP sources, if all of them have the same packet loss probability, the total aggregation rate for all TCP sources is:

$$R_{agg} = \frac{N\sqrt{1.5}/\sqrt{P_D}}{\overline{RTT}_H} \quad (8)$$

where $\overline{RTT}_H$ is the mean round trip time.

And the total packet arrival rate to the open loop is

$$\Lambda = \frac{M}{\lambda^{-1} + \mu^{-1}} \quad (9)$$

By substituting the (9) into (8), where $R_{agg} = \Lambda$, we can get the relationship between PD and λ :

$$T(p) = \frac{N\mu\sqrt{1.5}}{M\mu RTT_H\sqrt{p} - N\sqrt{1.5}}. \quad (10)$$

6.2.3 Combined Close Loop Model

After we got the relationship between PD and λ from both open loop model and TCP model, we can get the solution of closed loop model by solving the fixed point equations of (10) (3) and (4) by performing binary search. A detailed algorithm of solving those equations can be found in [23], as well as prove of unique existence of the fixed-point solution.

6.3 Simulation

In order to validate the analytical model that is also applicable to heavy-tailed flows, I developed the simulations for both close loop model and open loop model. For the open loop model, the simulation is done by C++. For closed loop model, I used an advanced network simulator NS2 to process the simulation under TCP.

6.3.1 Simulation for Open Loop Modal

The open loop simulation is implemented using C++, which is a simulation program with five inputs plus one optional input: number of incoming links, number of server, buffer size in terms of packets, ON/OFF ratio for each source (λ/μ) and data output name (optional).

In the simulation, there are some differences between “real” switch and queuing model in the simulation. I listed the main characteristics of the queuing model in the simulation below:

1. When a free port is available, the arrival packet is progressively directed to the output port.
2. When there are no output ports and no free buffers available, the incoming packet will be discarded.
3. A buffer space is busy means that it is holding a whole packet or part of it. It is possible that a packet only has one bit left in the buffer space. The buffer space is still regarded as busy so that the buffer space cannot accept any other packet
4. When there is one free buffer and all the output ports are not available, a packet on arrival is forwarded to the buffer.
5. While a departure release an output, a packet is sent from the buffer to the output port when the buffer is not empty.
6. In the simulation, no packet can be written to the input of the buffer while another packet is reading from output buffer.
7. An output is released while no packets in the buffer, the number of busy buffer reduce one.

Note that the sixth characteristic is different from the packet switch, while in an optical FIFO buffer, one packet can be written to the input of the buffer while another packet is read from the output of the buffer.

The simulation is divided into three main classes. There are CQueueServer, CQueue and main body for simulation. CQueueServer is a class to simulate the output port of the switch. CQueue is the class to simulate the whole queuing system. And the main body for simulation is for heavy-tailed flow generation. For the detail of the implementation of open loop simulation, you can refer to the Appendix I.

The screenshot shows a software window titled "Simulation" with a blue header bar and a red close button. The window is divided into two main sections: "Input" and "Results".

Input Section:

- No. of Incoming Links:** A text box containing the value "2".
- No. of Servers:** A text box containing the value "1".
- Buffer size (in packets):** A text box containing the value "1".
- On-off Ratio for each source:** A text box containing the value "1".
- Simulation packets:** A text box containing the value "1000000".
- Name of Save File:** A text box containing the value "data.xls".

Results Section:

- Loss Probability:** A text box containing the value "0".
- PEB Approximation:** A text box containing the value "0".
- Open Loop Simulation with Poisson Process:** An empty text box.
- Open Loop Simulation with heavy-tailed flow:** An empty text box.
- Simulate Button:** A large rectangular button with the text "Simulate" inside.

Figure 12 User interface for open loop simulation - input

The final interface of my open loop simulation is showed in Figure 12. By entering the parameter needed in upper box, then press the simulate button, the result will be displayed in the lower box. The simulation time is depend on how many packets to simulate and the complexity of the network. As showed in Figure 13, the result obtained from the simulation with heavy-tailed flows is compared with the simulation result with Poisson model and PEB approximation. And the result will be presented in the later part of this report.

Input	
No. of Incoming Links	100
No. of Servers	10
Buffer size (in packets)	5
On-off Ratio for each source	1
Simulation packets	1000000
Name of Save File	data.xls

Results	
PEB Approximation	0.799925
Open Loop Simulation with Poisson Process	0.785506
Open Loop Simulation with heavy-tailed flow	0.799370

Simulate

DONE

Figure 13 User interface for open loop simulation - result

6.3.2 Simulation for Closed Loop Model

For closed loop, the TCP need to be introduced to the simulation. The complexity for develop the simulation by programming language we usually

used such as JAVA, C++ and C will be quite high. Thus in the closed loop simulation, I used an advanced network simulator NS2. NS2 is a widely used network simulator for study TCP/IP network and wireless network. However, it lacks components to simulate optical network. The link and queue need to be extended to develop the optical link and optical queue. In this case, I used the same optical components object in [23], which is originally developed by University of California.

For the heavy-tailed flow generation, I used the The Pareto On/Off Traffic Generator embodied in the original NS2, which has been widely used for generating traffic that exhibits long range dependency. For the NS2 commands to create a Pareto traffic Generator, you can refer to Appendix II.

After the simulation is done, you can get the information about every individual packet from NS2, such as source, destination and packet size. The data format is displayed at Figure 14.

```

----- null}
90 r -t 0.418008489882538 -s 0 -d 2 -p ack -e 40 -c 1 -i 5 -a 1 -x {1.3 7.0 0 ----- null}
91 + -t 0.418008489882538 -s 2 -d 7 -p ack -e 40 -c 1 -i 5 -a 1 -x {1.3 7.0 0 ----- null}
92 - -t 0.418008489882538 -s 2 -d 7 -p ack -e 40 -c 1 -i 5 -a 1 -x {1.3 7.0 0 ----- null}
93 h -t 0.418008489882538 -s 2 -d 7 -p ack -e 40 -c 1 -i 5 -a 1 -x {1.3 7.0 -1 -----
null}
94 r -t 0.420886153217435 -s 2 -d 0 -p pareto -e 1040 -c 1 -i 3 -a 1 -x {6.0 1.2 0 -----
null}
95 + -t 0.420886153217435 -s 0 -d 1 -p pareto -e 1040 -c 1 -i 3 -a 1 -x {6.0 1.2 0 -----
null}
96 - -t 0.420886153217435 -s 0 -d 1 -p pareto -e 1040 -c 1 -i 3 -a 1 -x {6.0 1.2 0 -----
null}
97 h -t 0.420886153217435 -s 0 -d 1 -p pareto -e 1040 -c 1 -i 3 -a 1 -x {6.0 1.2 -1
----- null}
98 r -t 0.421718153217435 -s 2 -d 0 -p pareto -e 1040 -c 1 -i 4 -a 1 -x {6.0 1.2 0 -----
null}
99 + -t 0.421718153217435 -s 0 -d 1 -p pareto -e 1040 -c 1 -i 4 -a 1 -x {6.0 1.2 0 -----
null}
100 r -t 0.422550153217435 -s 0 -d 1 -p pareto -e 1040 -c 1 -i 3 -a 1 -x {6.0 1.2 0 -----
null}
101 + -t 0.422550153217435 -s 1 -d 0 -p ack -e 40 -c 1 -i 6 -a 1 -x {1.2 6.0 1 ----- null}
102 - -t 0.422550153217435 -s 1 -d 0 -p ack -e 40 -c 1 -i 6 -a 1 -x {1.2 6.0 1 ----- null}
103 h -t 0.422550153217435 -s 1 -d 0 -p ack -e 40 -c 1 -i 6 -a 1 -x {1.2 6.0 -1 -----

```

Figure 14 Raw data from close loop simulation

In order to get achieve meaningful result from the simulation data, a simple

Linux bash script need to be implemented for data processing. By running my

own script, a simulation report was generated as shown in Figure 15.

```

1|
2 sack buf:0 ER:2 src_ER:5 paced:0 outWL:2 inWLc:25 outWLc:10 time:50 3439 171.000 5373 179.000
953 17.895000000000003 5.0429753122536649
3 sack buf:5 ER:2 src_ER:5 paced:0 outWL:2 inWLc:25 outWLc:10 time:50 20171 1008.000 31115
1037.000 898 98.152500000000003 32.860054492498925
4 sack buf:10 ER:2 src_ER:5 paced:0 outWL:2 inWLc:25 outWLc:10 time:50 28836 1441.000 43318
1443.000 833 139.77000000000001 58.182032020203607
5 sack buf:100 ER:2 src_ER:5 paced:0 outWL:2 inWLc:25 outWLc:10 time:100 142011 2366.000 190325
2379.000 990 235.34875000000002 46.155369219761319

```

Figure 15 Report for closed loop simulation

A

Animation can be generated automatically in NS2 from the simulation raw data. In Figure 16, 17, there are two frames I captured from one simulation of small-scale network with only 4 TCP sources. Nodes 4-7 are four TCP sources generating heavy-tailed flows. Nodes 2, 3 are edge routers and Node 0 is the DWDM switch. You can clearly see the packets queuing in the buffer from Figure 17 and red feedback TCP ACK packets in Figure 16. In the simulation, the number of TCP sources are usually larger than 10, the reason for using 4 sources here is for better viewing. Moreover, the nice animation showed here can also prove my closed loop simulation is correct.

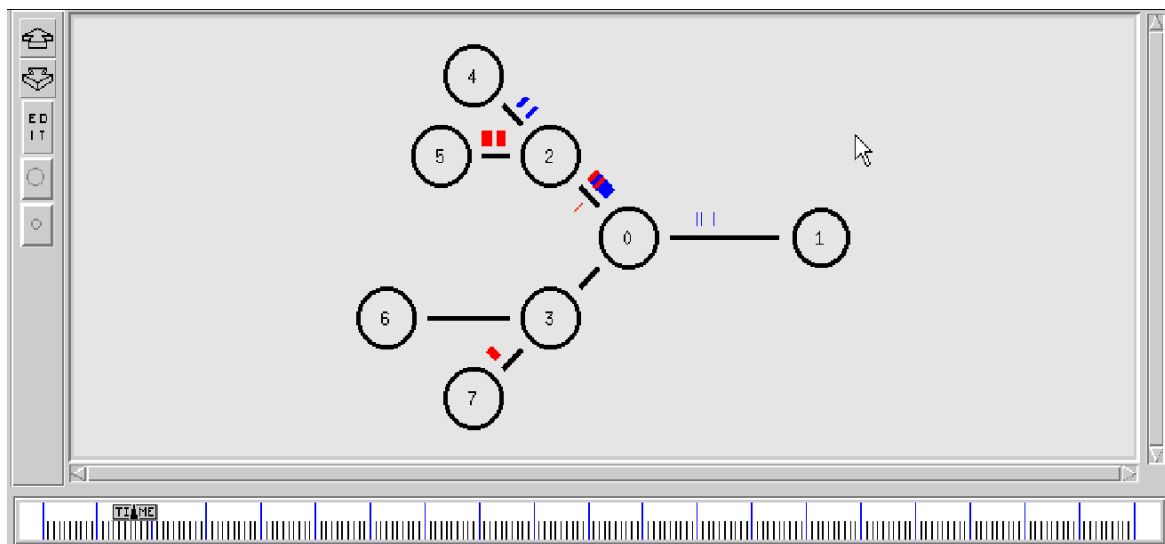


Figure 16 Animation for closed loop simulation

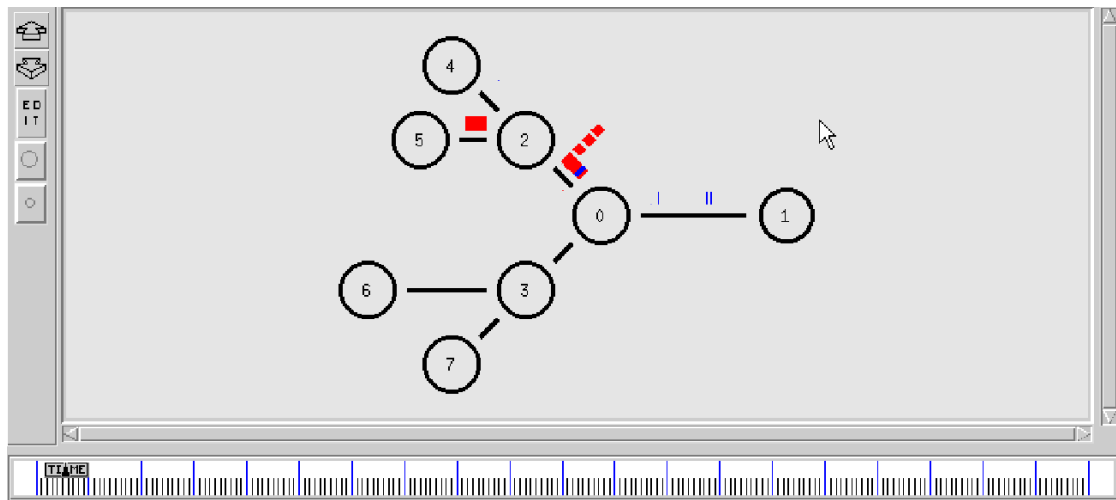


Figure 17 Animation for closed loop simulation

7. RESULT AND DISCUSSION

7.1 Open Loop Validation

In order to know how accurate the result approximated by the open loop model will be, the prediction will be compared with the results from both open loop simulations with heavy-tailed flows and Poisson process respectively.

And the result showed here is loss probability varying with the normalized intended load across different buffer size. The normalized intended load here is defined by $M\lambda / K[\lambda + \mu]$. All the results here are obtained by running the simulation with sufficient long time, which achieves the two-side 95% interval.

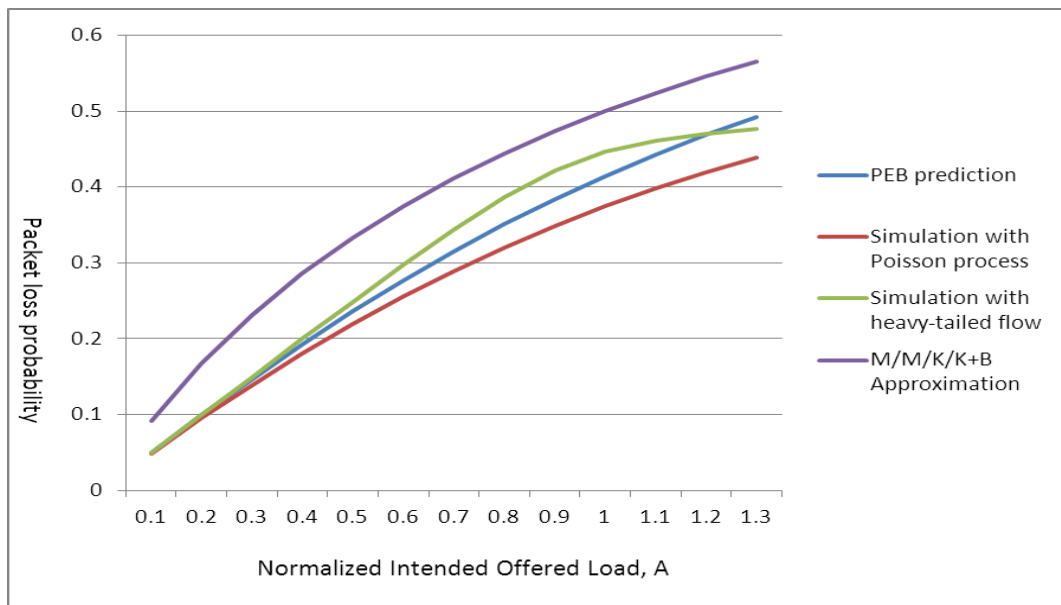


Figure 18 Open Loop simulation with 2 input ports , 1 out put port and buffer size equals to 0

In Figure 18, when the network has 2 incoming links, 1 output link and 0 buffer at output port, the result got from PEB model stays very close with the result I got from open loop simulation with heavy-tailed distribution. **However, in [23], the PEB model is considered to overestimate the packet discard probability over open loop simulation with Poisson process. The result in Figure 18 showed that, when we consider the LRD of heavy-tailed flows, PEB model actually gives a more accurate prediction.**

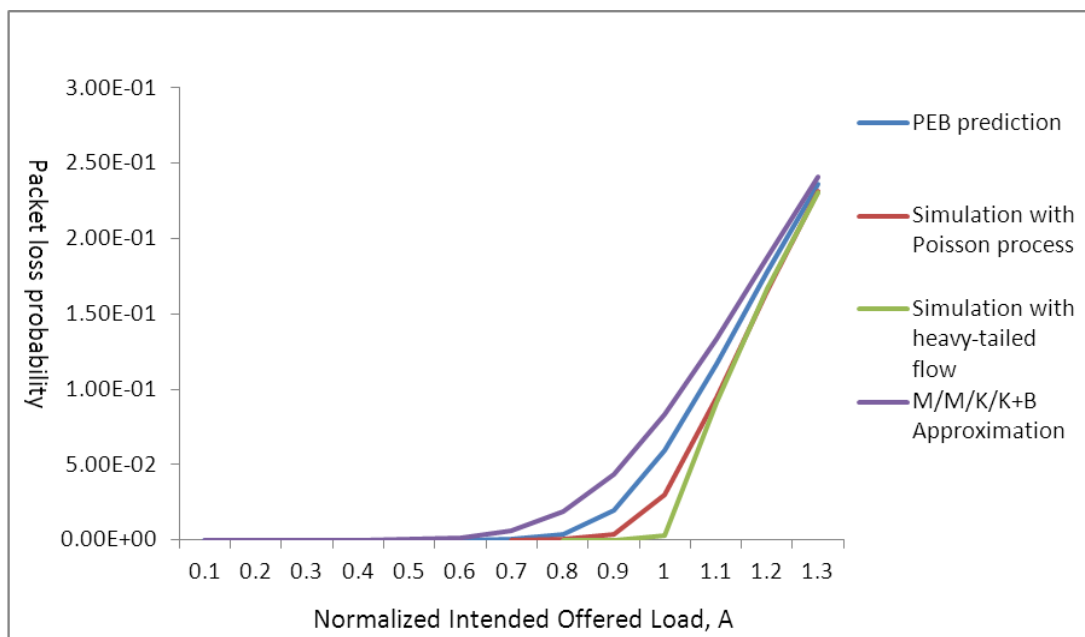


Figure 19 Open Loop simulation with 2 input ports , 1 out put port and buffer size equals to 10

In Figure 19, the same scenario is applied but with 10 packets of buffer. **Again, the PEB model gives a quite accurate result, the average difference between the loss probability given by PEB model and simulation with heavy-tailed flows is under 1%. Note that all the lines**

converge when the load is high. Moreover, the M/M/K/K+B still gives less accurate result, because it over-estimates the arrival rate at state K+B in which discards can occur^[23].

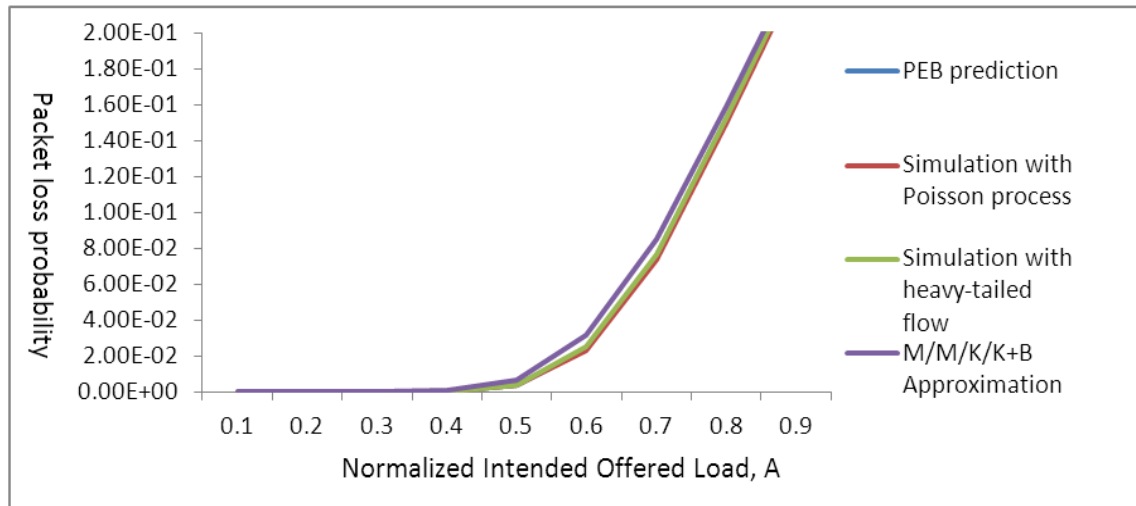


Figure 20 Open Loop simulation with 80 input links , 20 output links and buffer size equals to 0

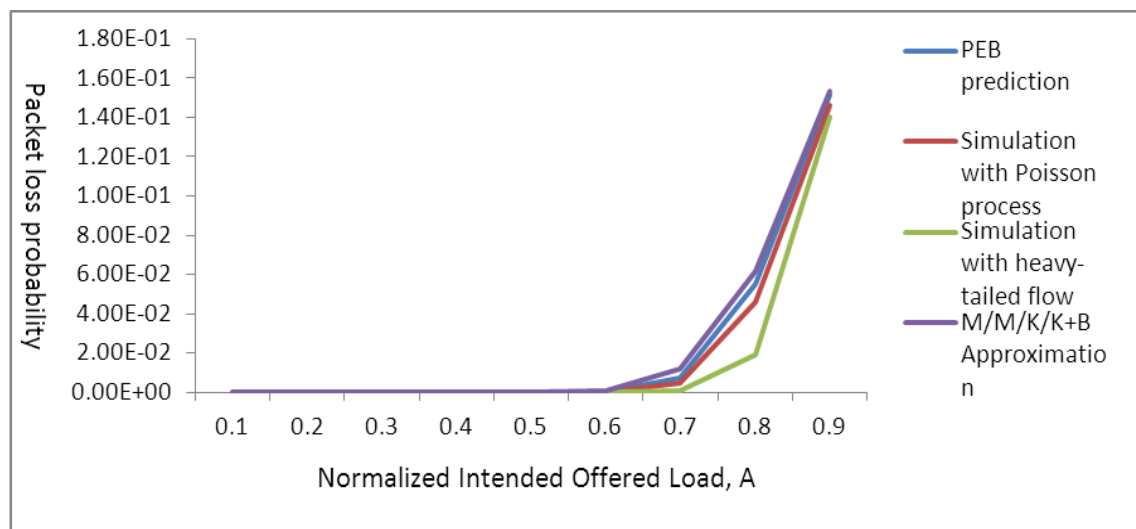


Figure 21 Open Loop simulation with 80 input links , 20 output links and buffer size equals to 10

For the case that we have large K and large M, in Figure 20 and Figure 21, it shows the loss probability of a network having 80 input links, 20 output links with zero buffer and 10 buffers respectively. **The result given by PEB model remains very accurate, and the average difference is about 0.1%. However, note that when there are ten buffers in the network and the load is relatively low, the simulation of Poisson process gives larger loss probability.** To investigate whether this finding also applies to the network with small buffer, say 1 buffer. I did one more round of simulation with network having 80 input links and 20 output links with buffer of one packet.

Table 1 Comparision between simulations with 80 input links, 20 output links, buffer size equals to 1

Normalized Intended Offered Load	Simulation with Poisson process	Simulation with heavy-tailed flow
0.1	5.90E-05	5.90E-05
0.2	5.90E-05	6.10E-05
0.3	5.90E-05	6.10E-05
0.4	0.000131	0.000164
0.5	0.001733	0.001863
0.6	0.014444	0.014772
0.7	0.055993	0.055703
0.8	0.125428	0.126623
0.9	0.209275	0.209272
1	0.290624	0.292137
1.1	0.365653	0.367039
1.2	0.433729	0.432339
1.3	0.491586	0.49023

Since the results got from two simulation stays every close, I use the table instead of graph. When there is only one buffer in the network, simulation with heavy-tailed flows gives slightly larger result than the simulation with

Poisson process. That means when there is relatively large buffer in the network and the input load is low, in the long run, the LRD traffic will give lower loss probability than Poisson. This is because when the load is low, the network will more likely to have large arrival time than to have large traffic flow.

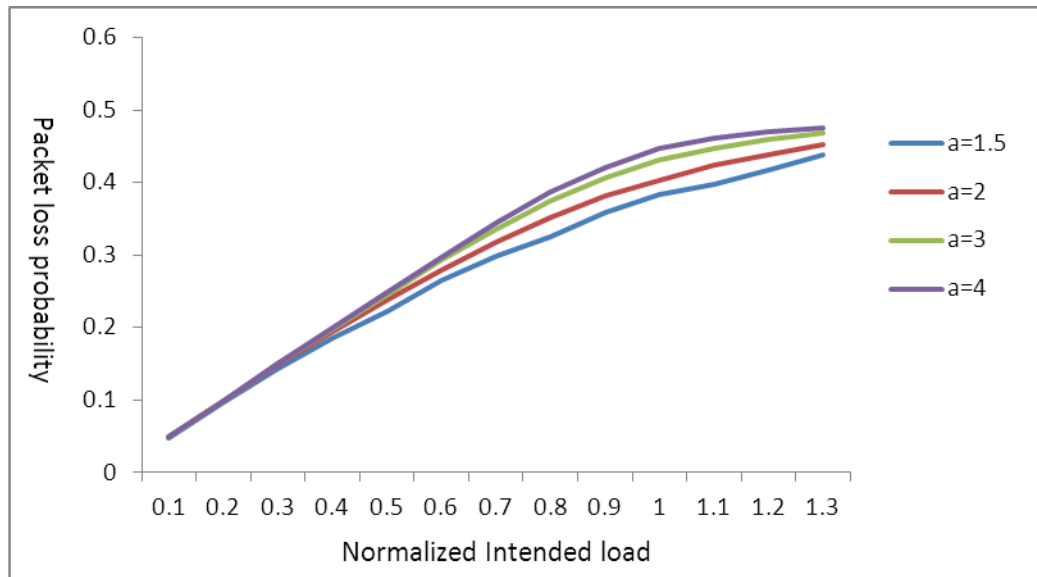


Figure 22 Open loop simulation across different value of shape parameter

In Figure 22, there is another scenario which shows my open loop simulation is correct. When there is no buffer in the network, I did a simulation by changing the shape parameter of Pareto distribution from 1.5 to 4. When there is no buffer in the network, it can be considered as an M/M/K/K model, which is insensitive to the service time distribution. The results from different values of shape parameter in Figure 22 stay very close which intuitively show my open loop simulation is correct.

7.2 Closed Loop Validation

In the closed loop validation, the result from both PEB approximation and NS2 simulation are compared. In NS2 simulation, I consider a network has 10 TCP sources, and each TCP source send heavy tailed flow at 100K bit/second. And the number of output ports is 2 and each output link has a capacity of 10Mb/second. Although it is small scale network, it still takes me about half a month to simulate 4 scenarios with different output buffer size.

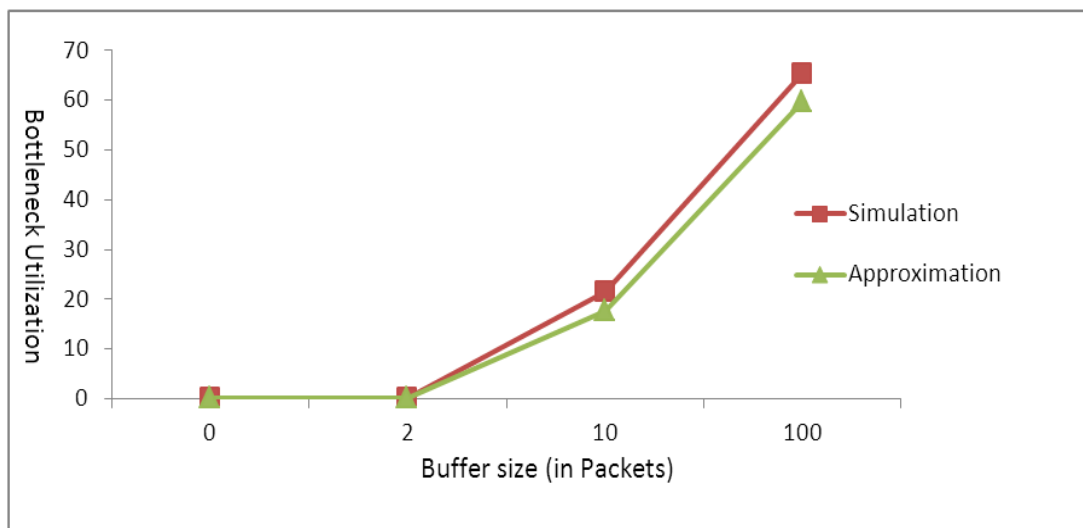


Figure 23 Closed loop validation

From Figure 23, we can see that when the buffer size is relatively small, say 0 and 2, the results from simulation and approximation stay quite close with each other. And when the buffer size becomes larger, say 10 and 100, the result from PEB approximation is slightly conservative compare with that in simulation, but it still can be considered to be accurate. The result in [23] reveals that when the buffer size becomes larger, the PEB approximation yields less accurate result, which agree the result in Figure 23. Moreover, I

found the PEB approximation achieved more accurate estimation for network with heavy tailed flows compare with network with Poisson process.

7.3 Approximation for Large Scale Network

The PEB approximation is proved to be accurate in previous parts, now it is time to use it for estimation of large scale network. However, since the time is tight, I did not do the extrapolation from results of small scale networks to that in large scale networks by using curve fitting. Based on the achievement above, it is confident to show the extrapolation can be done to get the prediction of large scale network. However, without doing Curve fitting, I can still approximate the result for a network with relatively lower bit-rate; say 1000 TCP sources and each send heavy-tailed flows at a rate of 10Mb/second. Then we can figure out the buffer requirement at the bottleneck switch of network with 10 Gb/second. By using the model in this project with extrapolation, larger scale network, such as one with 100 Gb/second input, will be studied later.

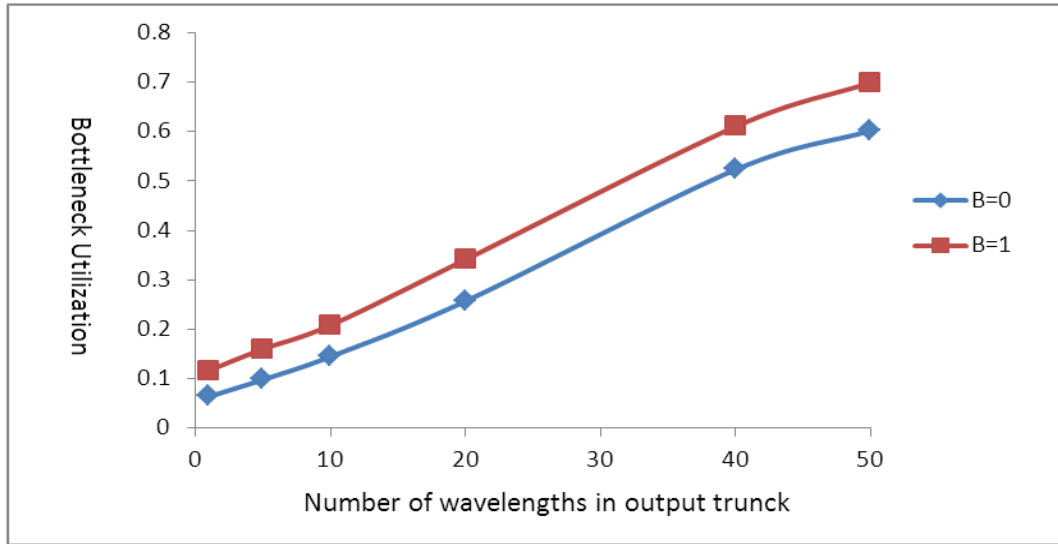


Figure 24 Bottleneck utilization approximation

As shown in Figure 24, the bottleneck utilization grows as the number of wavelengths in each output trunk increases. If there are 1000 TCP flows and 50 wavelengths in one output trunk, the bottleneck utilization reaches about 61% with no buffer. When there is a one buffer at per output link, the utilization achieves 69% utilization, which figures out that when the OBS network has sufficient number of wavelengths, a utilization about 70% can be achieved when there is a small buffer per output wavelength, and can even get above 60% of utilization where there is no buffer at all. Note that the utilization got here is lower than the result in [23], it is reasonable by considering the LRD of traffic and not enough output trunks were available. However, after the extrapolation part is done, a higher utilization above 80% will be reached when there are more TCP flows and larger number of output trunks in the network by estimation.

8. CONCLUSION

In this final year project, I learned a lot about network performance evaluation, especially for OBS network. I got an insight into the OBS network structure, such as OBS burst assembly algorithms, OBS reservation protocols and contention resolving scheme, then developed my own opinion on the benefits and drawbacks of OBS as well as its proposed techniques in the optical telecommunication field. Moreover, I also learned the basics about the performance of TCP and LRD of Internet traffic today, which built a solid foundation for the quantitative part of my project later.

In the quantitative part of my project, I used the same network topology and analytical model in [23] to analysis the performance of TCP over OBS with heavy-tailed flows. In order to validate the analytical model is also applicable to the network with LRD traffic, I implemented my own simulations for both open loop and closed loop model. Although the simulations towards LRD traffic is considered to be difficult and computational prohibitive, I still achieved accurate result when I simulated the small scale networks instead of large scale networks. However, by extrapolating the result from small networks, it is feasible to get the results for large scale network by curving fitting and approximation. And I will work on the extrapolation part in the further development of my project.

By analysing the results got from the quantitative part, my project demonstrated that **the PEB approximation can achieve higher accuracy of loss probability for the network with heavy-tailed flows compared with the one with Poisson model when this no TCP involved. Moreover, by introducing the influence from TCP, the PEB model still gave more accurate result. Then I showed that in a DWDM network with sufficient number of output wavelengths, TCP can achieve about 70% utilization with one buffer and more than 60% without any buffers when there are only 2 output fibres. And it is believed that if there are sufficient output fibres available in the same case, the utilization achieved will be more than 80% by estimation, which will be validated in the further development of my project.**

To summarize, my project investigate that the utilization in optical network is not only related to the buffer size, but also the number of wavelengths available in output fibre. Moreover, by considering the LRD traffic, this project also demonstrated that the PEB model is a good tool for investigating the buffer requirement at a DWDM switches in OBS.

9. APPENDIX

Appendix I: open loop simulation details

The simulation consists of three main parts:

a. CQueueServer:

This class is to simulate an output port of a queuing system, which is also considered as a server of a queuing system.

b. CQueue:

This class is to simulate the whole queuing system. A queuing system may have several output ports. Each port is a CQueueServer object. A queuing system may also include multiple buffer spaces. CQueue class uses a variable list named `m_pdExpectFinishTime` to store the timestamp when the packet in an output port finishes transmission. If at this moment an output port is idle, now is its expected finishing transmission time.

In the source code, there is no separate buffer class. CQueue class handles the buffering operation which maintains the following variables:

Buffer size: the number of buffer spaces in terms of packets;

Number of buffered packets: How many buffer spaces are now occupied by packets;

Followed Figure 25 is the logical flow of Enqueue function of CQueue.

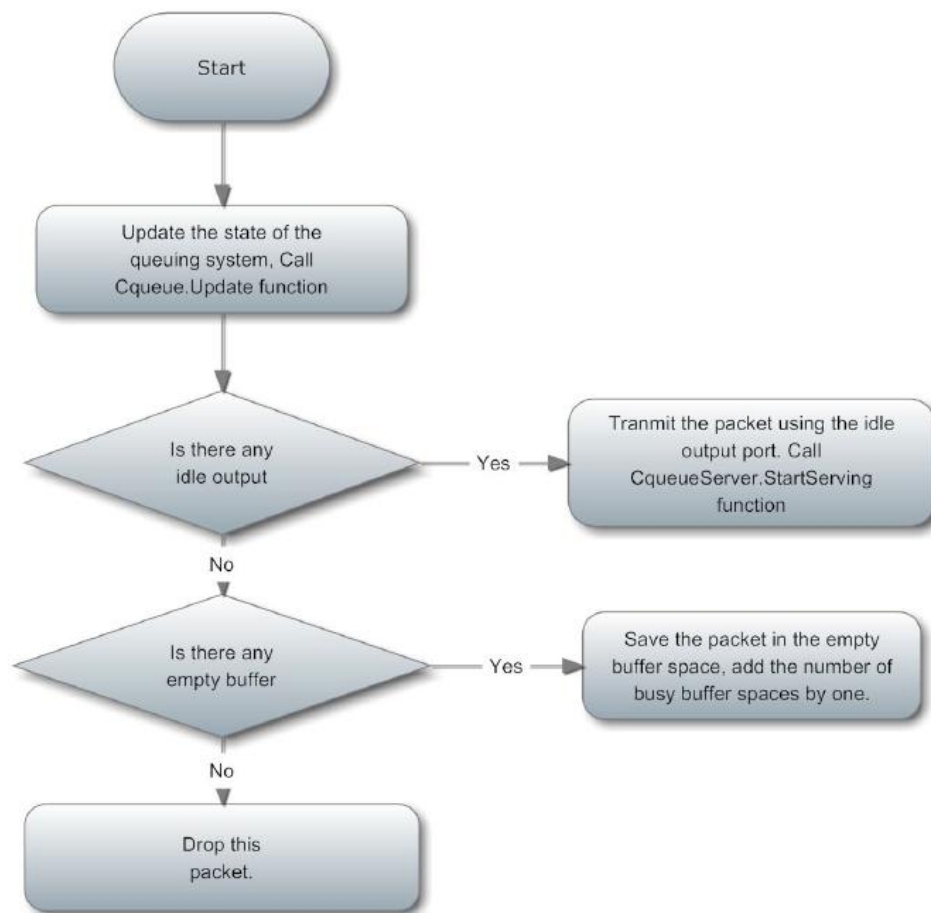


Figure 25 Open loop simulation logical flow - CQueue

c. The main body:

This part is to simulate the packet generation process. Each incoming port is a Pareto traffic source. Packets are then sent to CQueue object to process. If all the simulation packets have sent, the program quits. The logical flow shows below in Figure 26.

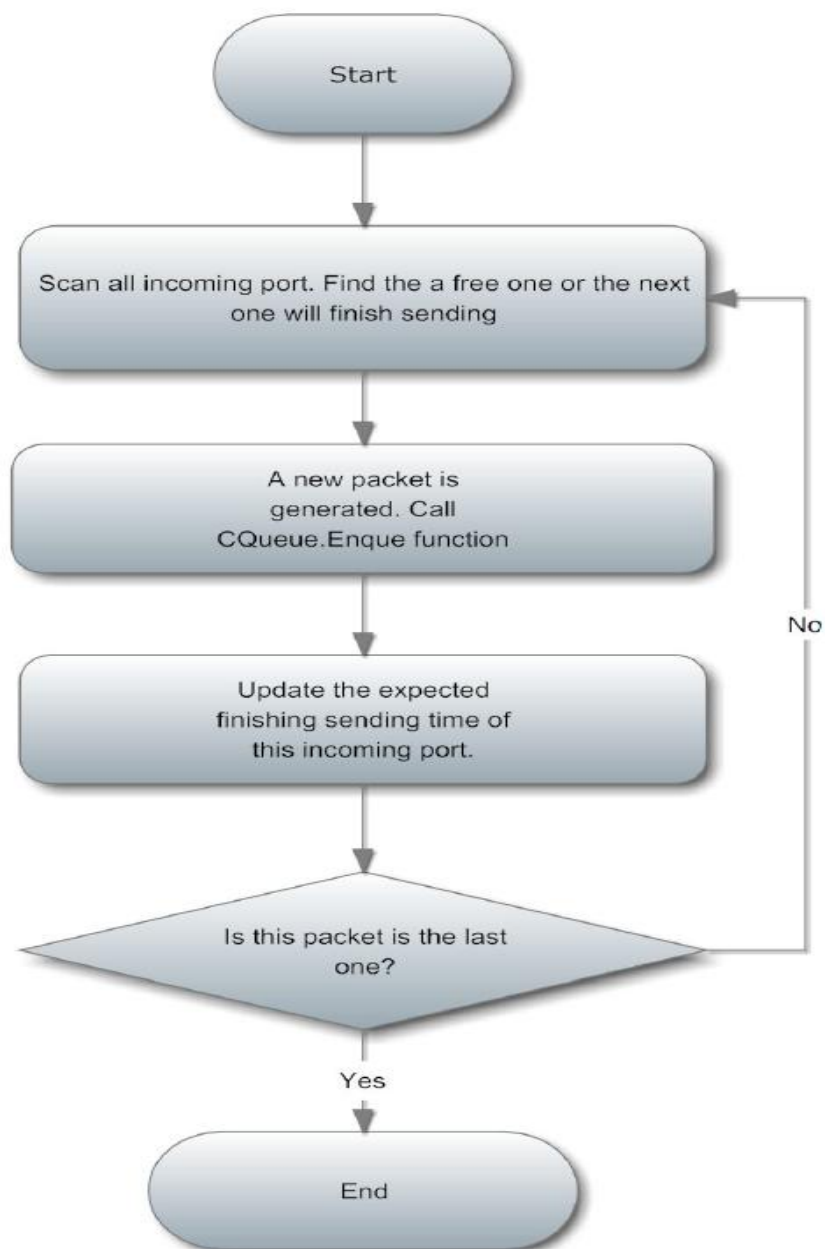


Figure 26 Open loop simulation logical flow - Main function

Appendix II: Pareto traffic generator

Pareto On/Off traffic generator can be created by following codes:

```
set tcp [new Application/Traffic/Pareto]
```

```
$tcp set burst_time_ 500ms           //set mean ON (burst)
time
$tcp set idle_time_ 500ms           //set mean OFF (idle)
time
$tcp set rate_ 100k                 //set send rate during
ON period
$tcp set packetSize_ 1000           //set Packet size
$tcp set shape_ 2                   //set shape parameter
                                     for Pareto
                                     //distribution
```

10. GLOSSARY

Dense Wavelength Division Multiplexing (DWDM)

DWDM is a technology that enables multiple signals with different wavelengths to transmit through one single optical fibre at the same time.

Optical Circuit Switching (OCS)

OCS is a packet switching scheme which reserves the end-to-end optical path for each connection.

Optical Packet Switching (OPS)

OPS is similar to the electronic packet switching scheme but in optical domain. All the traffic in OPS network is carried by packets, which are switched individually without any end-to-end path reservation.

Optical Burst Switching (OBS)

OBS is a combination of OPS and OCS. In OPS, all the traffic are carried by burst, which are aggregated by packets arriving at edge router with same destination. Bursts are transmitted by using one-way reservation between each hop.

Traffic Self-similarity

Traffic that exhibits the same property under different or all time scaling.

Long Range Dependency

LRD is a property of certain stationary stochastic processes, which have slow decay of correlations.

Burst Control Packet (BCP)

BCP is consider a separate header for each data burst, which includes the control information about its corresponding burst, such as source, destination and burst length.

Just-In-Time (JIT)

JIT is a wavelength reservation scheme in OBS, which reserves a free wavelength immediately when a burst control packet arrives.

Just-Enough-Time (JET)

JET is a wavelength reservation scheme in OBS, which the free wavelength is reserved for a certain period of time according to the burst length information in burst control packets.

Transmission Control Protocol (TCP)

TCP is the major transmission protocol in the Internet, which provide reliable communication by applying end-to-end connection.

NS2

NS2 is an advanced network simulator under Linux environment, which provides good support to simulation of wired and wireless networks.

11. REFERENCES

- [1] Ryan, J. P.: WDM: North American deployment trends. IEEE Communications Magazine 36 (Feb. 1998), 40–44.
- [2] Klaus Dolzer, Christoph Gauger, Jan Späth, and Stefan Bodamer, “Evaluation of Reservation Mechanisms for Optical Burst Switching”, AEÜ Int. J. Electron. Commun., 55 (2001) No. 1, 1–1.
- [3] Biao Chen, and Jianping Wang, “Hybrid Switching and P-Routing for Optical Burst Switching Networks”, IEEE Journal on Selected Area in Communications, Vol. 21, No. 7, September 2003
- [4] I. Chlamtac, A. Farago, and T. Zhang, “Lightpath (wavelength) routing in large WDM networks,” IEEE J. Select. Areas Commun., vol. 14, pp. 909–913, June 1996.
- [5] F. Callegati, G. Corazza, and C. Raffaelli, “Exploitation of DWDM for optical packet switching with quality of service guarantees,” IEEE J. Select. Areas Commun., vol. 20, pp. 190–201, Jan. 2002.
- [6] Anan, M.T. Chaudhry, G.M. Benhaddou, D., “Architecture and Performance of A Next-Generation Optical Burst Switch (OBS)”, 3rd International Conference on Broadband Communications, Networks and Systems, 2006. BROADNETS 2006.
- [7] Zvi Rosberg, Hai Le Vu, Moshe Zukerman, Jolyon White, “Performance Analyses of Optical Burst-Switching Networks”, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 21, NO. 7, SEPTEMBER 2003

- [8] Cisco Systems, "Approaching the Zettabyte Era," White Paper, June 16, 2008;
- [9] Chen, Yang; Qiao, Chunming and Yu, Xiang; "Optical Burst Switching (OBS): A New Area in Optical Networking Research", IEEE Network Magazine, Vol. 18 (3), pp. 16–23, May–June 2004.
- [10] Z. Rosberg, H. L. Vu, M. Zukerman and J. White, "Performance Analyses of Optical Burst Switching Networks", IEEE Journal of Selected Areas in Communications, vol. 21, no. 7, Sept. 2003, pp. 1187-1197.
- [11] X. Yu, Y. Chen, and C. Qiao, "Study of traffic statistics of assembled burst traffic in optical burst switched networks," in Proceeding of Opticomm, 2002, pp. 149–159.
- [12] A. Ge, F. Callegati, and L. Tamil, "On optical burst switching and Self-similar traffic," IEEE Communications Letters, vol. 4, pp. 98–100, March 2000.
- [13] M. Yoo and C. Qiao, "Just-Enough-Time (JET): A High Speed Protocol for Bursty Traffic in Optical Networks," IEEE/LEOS Technologies for a Global Information Infrastructure, pp. 26-27, Aug. 1997
- [14] Ilia Baldine, et al, "JumpStart: A Just-in-Time Signaling Architecture for WDM Burst-Switched Networks," IEEE Communications, vol. 40, no. 2, pp. 82-89, February 2002
- [15] A. Detti and M. Listanti, "Application of Tell and Go and Tell and Wait Reservation Strategies in a Optical Burst Switching Network: a Performance Comparison," Proceedings of the 8th IEEE, Bucharest, June 2001

- [16] J. Teng and G.N. Rouskas, A Detailed Analysis and Performance Comparison of Wavelength Reservation Schemes for Optical Burst Switched Networks, *Photonic Network Communications* 9(3) (2005) 75–81.
- [17] W. E. Leland, M. S. Taqqu, W. Willinger and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1-15, February 1994.
- [18] Shu-Gang Liu, Pei-Jin Wang, Lin-Jie Qu, Modeling and Simulation of Self-Similar Data Traffic Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on Volume 7, 18-21 Aug. 2005 Page(s):3921 - 3925
- [19] M. E. Crovella, M. S. Taqqu and A. Bestavros, "Heavy-tailed probability distributions in the World Wide Web," in R. J. Adler, R. E. Feldman and M. S. Taqqu, Eds. *A practical guide to heavy tails: statistical techniques and applications*, pp. 3-25, Birkhauser, 1998.
- [20] G. Kramer, "On generating self-similar traffic using pseudo-pareto distribution," technical brief, Department of Computer Science, University of California
- [21] A. Detti and M. Listanti, "Impact of Segments Aggregation on TCP Reno Flows in Optical Burst Switching Networks," in *Proceedings of INFOCOMM*, 2002, vol. 3, pp. 1803–1812.
- [22] E. W. M.Wong and M. Zukerman, "Bandwidth and buffer tradeoffs in optical packet switching," *J. Lightw. Technol.*, vol. 24, pp. 4790–4798, Dec. 2006.

- [23] Eric W. M. Wong, Lachlan Andrew, Tony Cui, Bill Moran, Andrew Zalesky, Rodney S. Tucker, and Moshe Zukerman, "Towards a Bufferless Optical Internet", IEEE/OSA Journal of Lightwave Technology, Volume 27, Issue 14, July 15, 2009, pp. 2817 - 2833.
- [24] M. Mathis, J. Semke, J. Madhavi, and T. Ott, "The macroscopic behaviour of the TCP congestion avoidance algorithm," ACM Comput. Commun. Rev., vol. 27, no. 3, pp. 67–82, Jul. 1997
- [25] E. W. M. Wong and M. Zukerman, "Bandwidth and buffer tradeoffs in optical packet switching," J. Lightw. Technol., vol. 24, pp. 4790–4798, Dec. 2006.