Analytical Modeling of Multi-Channel Optical Burst Switching with Multiple Traffic Classes

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULLFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

by

Volkan Dinç

August 2011

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

____

Assoc. Prof. Dr. Nail Akar (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Sinan Gezici

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Tuğrul Dayar

Approved for the Graduate School of Engineering and Sciences:

Prof. Dr. Levent Onural

Director of the Graduate School of Engineering and Sciences

ABSTRACT

ANALYTICAL MODELING OF MULTI-CHANNEL OPTICAL BURST SWITCHING WITH MULTIPLE TRAFFIC CLASSES

Volkan Dinç
M.S. in Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Nail Akar

August 2011

In this thesis, we study an Optical Burst Switching (OBS) node with links carrying multiple wavelength channels (called hereafter channels) with multiple traffic classes. We assume that offset-based service differentiation is used to differentiate among these traffic classes in terms of packet loss probabilities. We first propose a basic scheme, called bLAUC (Basic Latest Available Unused Channel) for channel scheduling. Although practicality of the bLAUC scheme is relatively limited when compared to other conventional schedulers such as LAUC, we study bLAUC in this thesis due to its tractability to analysis and moreover bLAUC possesses certain crucial properties of conventional schedulers. We then propose an iterative procedure to approximate per-class loss probabilities for the OBS link of interest when packet arrivals to the link are Poisson and packet lengths are exponentially distributed. In our iterative procedure, we model a multi-channel OBS link with Poisson arrivals by a single channel Markov fluid queue with occupancy-dependent packet arrival intensities. The proposed procedure provides acceptable approximations for a wide range of scenarios with relatively low complexity. Consequently, the proposed procedure can be used in optimization problems concerning multi-class OBS and in finding guidelines to effectively utilize OBS resources under loss probability constraints.

Keywords: OBS, QoS, Scheduling, Fluid queues

ÖZET

ÇOK SINIFLI TRAFİK ALTINDA ÇOK KANALLI OPTİK ÇOĞUŞMA ANAHTARLAMANIN ANALİTİK ANALİZİ

Volkan Dinç
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Doç. Dr. Nail Akar
Ağustos 2011

Çalışmamızda optik çoğuşma anahtarlama kullanan bir ağdaki birden çok dalgaboyu kanalına sahip bir düğümün çoklu trafik sınıflarına maruz kaldığı duruma yoğunlaşılmıştır. Trafik sınıflarının öteleme tabanlı servis kalitesi farklılığına gore oluştuğu varsayılmış ve servis kalitesi, çoğuşmaların değişik olasılıklarla kayboluşu olarak kendini göstermiştir. Çalışmamız içinde bLAUC (basic, temel LAUC) olarak adlandırdığımız bir kanal ayırma algoritmasını önerdik ve inceledik. bLAUC diğer konvansiyönel yöntemlere (LAUC ve LAUC-VF) göre uygulanabilirliği daha sınırlı da görünse, bir çok kaynak ayırma yönteminin genel özelliklerini taşımakta ve analizi daha mümkün hale gelmektedir. Çalışmamızda, bLAUC kullanan bir düğümdeki çoğuşma kaybetme olasılıklarını, paket gelişleri bir Poisson süreciyken ve paket uzunlukları üstel bir dağılıma sahipken yaklaşık olarak hesaplamak için bir yöntem önermekteyiz. Yöntemimizde, çoklu kanala sahip bir düğümün çoğuşma kaybetme karakteristiğini doluluğuna bağlı olarak paket geliş miktarı değişen bir Markov akışkan kuyruğu ile modellemekteyiz. Önerdiğimiz yöntem bir çok durumda, daha önce çözümü olmayan bu problem için, kabul edilebilir sonuçlar sunmaktadır ve bunu karmaşıklığı oldukça az olan bir algoritma ile sağlamaktadır. Yöntemimiz tasarım süreçlerindeki optimizasyonlarda veya genel tasarım fikirleri oluşturma gibi faaliyetlerde rahatlıkla kullanılabilinir.

Anahtar Kelimeler: Optik çoğuşma anahtarlama, servis kalitesi ayrımı, Markov akışkan kuyrukları

[This page is left blank intentionally]

Acknowledgements

I would like to thank my advisor (Assoc.) Prof. Dr. Nail AKAR for letting me experience the privilege of working with him and being more than an advisor when it is needed.

I appreciate financial support provided by TUBITAK BIDEB program 2228.

I would also like to my family for always being there for me, unconditionally.

[This page is left blank intentionally]

Table of Contents

List of Figures

List of Tables

To my family …

Chapter 0

Introduction

Today, optical networks are one of the strongest candidates for the future Internet. For a brief comparison between conventional electronic networks and optical networks, we refer the reader [1]. Optical networks can be categorized according to how datum is relayed throughout its path. Using this criterion, it is possible to define three types of optical networks: Opaque, translucent and transparent (See Figure 1). In an opaque network, data to be relayed on a point-to-point link (fiber) between two nodes is converted to an electronic signal at the node. After being processed an electronic signal, it is redirected to another link (fiber) as an optical signal. This method of changing domain of the data to be relayed is called optical-electronic-optical (O/E/O) conversion. Although a large amount of capacity is obtained using such point-to-point optical links, operating costs and electronic bottlenecks appear to be drawbacks of opaque networks. These drawbacks could be alleviated by using translucent networks [2] where O/E/O conversion ability is available only at a certain set of nodes[1]. However, these drawbacks could be eliminated by avoiding conversion between optical and electronic domains . Such networks, in which data is relayed as an optical signal throughout its path, are called transparent optical networks (or all-optical networks) [3]. The scope of our work lies within all-optical networks.

[1] This excludes edge nodes. Nodes interfacing a network with other networks which are to carry abilities needed to interface two networks are called edge nodes.

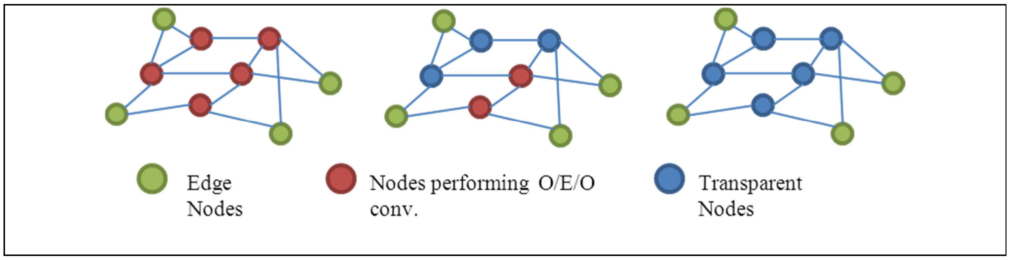


Figure 1. Sample illustrations of opaque, translucent, transparent optical networks (left to right)

There are three paradigms which attract researchers in switching in all-optical networks, namely: Optical circuit switching (OCS), optical burst switching (OBS) and optical packet switching (OPS). In OCS [4], as in other circuit switching techniques, an end to end circuit (lightpath in OCS context) is established and a part of the resources (usually wavelengths only) is dedicated to data to be relayed along that circuit (lightpath). Just like other circuit switching techniques, OCS cannot justify overhead in establishing a circuit if amount of data to be relayed is not large enough. Furthermore, due to the static nature of circuits (lightpaths), OCS cannot utilize network resources efficiently. On the other hand, in OPS [5], switching is carried out on a packet by packet basis. However, due to finer granularity[2] offered, OPS requires very fast (ns) switching techniques. Furthermore, fast header processing and buffering capabilities in the optical domain are also required in the OPS paradigm. The above-mentioned technical requirements cannot easily be met by current technologies and this renders the implementation of OPS very hard if not impossible. Limitations of OCS and OPS have eventually led to the paradigm of OBS. In OBS [6], coarser granularities compared to OPS are used and out of band signaling (separation of control plane from data plane) is preferred to overcome technological limitations faced by OPS. Bursts are formed by assembling multiple packets at the edge of the network. Even though burst assembly at the edge of the network causes additional delay, it relaxes the switching and burst processing requirements, making it possible to build OBS networks with current technologies. A brief comparison between the three switching paradigms is presented in Table 1.

[2] Granularity can be described as minimal entity which is being switched in the network

	OCS	OBS	OPS
Efficient granularity	In the order of GBs	~ KB-MB	~1KB
Control	Out of band	Out of band	In band
Control overhead	Connection setup	Burst Control Channel	Packet Header
Latency reasons	Setup time and propagation	Burst assembly and propagation	Propagation(Buffering in the future)
Advantages	QoS is established during setup. Reliability. Commercially available	High flexibility. Efficient network utilization. Slower switching constraints	Very high flexibility. Very efficient network utilization
Drawbacks	Low flexibility. Low network utilization. Long setup time	Traffic aggregation. Complexity of control plane. Components are partly available	High control complexity, problems in packet ordering. Components are not available
Predicted future	Short term deployment	Mid-term deployment	Long term deployment

Table 1. A simple comparison between main switching ideas in all-optical networks: Table 1 in page 846 of **[1]** gives a more detailed in comparison of these three paradigms.

For a more elaborate discussion of these three paradigms, we refer to [7] and Chapter 2 of [8]. However, this thesis concentrates only on OBS. We therefore provide a more detailed description of OBS in the next chapter.

Constructing analytical tools to model behavior of a network is an important task, since those may be helpful in designing or optimization processes. In this thesis, we aimed to provide an analytical method to model loss probabilities at OBS nodes under multi-channel, multi-class scenarios. The structure of this thesis can be summarized as: First, we will provide enough background knowledge and we will define our problem in the next chapter. In the following parts of this thesis we will focus on special cases of our problem. Then, we will present our proposed solution and numerical results. A more concise outline of this thesis can be found at the end of the next chapter.

Chapter 1

Background Material and the Problem Definition

Even though the motivation of this work is given in the introduction section, to define our problem in a formal way, some background knowledge is required. In this section, a brief overview of the OBS paradigm is given and concepts to be used in our problem definition are introduced. Respectively, overviews of OBS Networks and QoS in OBS paradigm are presented. Then, problem definition and the overview of the rest of the thesis will be presented.

1.1 A Brief Overview on OBS Networks

A simple illustration of an OBS network is given in Figure 2. Let us start with inspecting the entrance point to the OBS network, namely the ingress node. At an OBS ingress node, client data (packets) are aggregated into bursts according to a burst assembly algorithm[3]. Burst assembly algorithms can be categorized into three types: timer-based, threshold-based, and mixture of both. In timer-based assembly, a burst is simply formed by waiting for client packet arrivals and combining all such packets after a fixed time period. In threshold-based assembly, burst size becomes a constraint in which client packets are held in an assembly buffer until the cumulative size of the packets in the buffer exceeds the

[3] Bursts are disassembled at an egress node to obtain packets.

so-called buffer size threshold. Finally, mixture-based assembly methods depend on hybrid use of the previous two mechanisms.

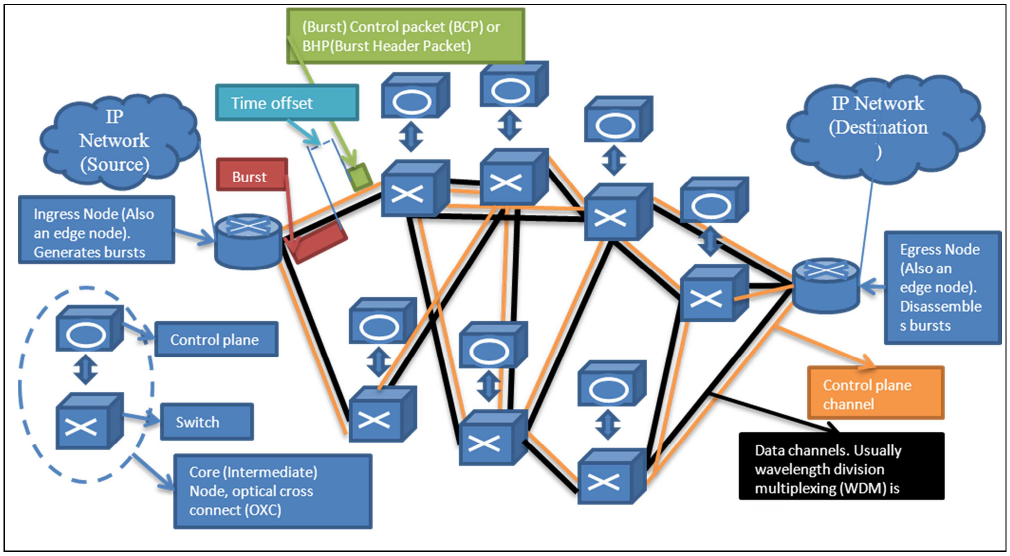


Figure 2. A simple illustration of an OBS network

Burst assembly is an important problem in itself; however, it is not within the scope of this thesis. Burst assembly algorithms play out the trade-off between burstification delays and large burst sizes which a better fit to slower[4] switching components. After a burst is formed, it cannot be sent right away, as shown in Figure 2. Considering that there is no actual buffering capability in OBS nodes, it should be clear that when the burst arrives, switch must be already configured for this burst. That is why there has to be a time offset between the (burst) control (or header) packet (BCP or BHP)[5]. However, how to determine the amount of this offset and how to utilize it remains as an important problem. Several signaling methods and resource scheduling algorithms are proposed to

[4] Compared to requirements of OPS. With larger granularity, OBS breaks the requirements of switching in OPS.
[5] Since control packets and actual data are separated in time, it is only natural to separate data and control planes. Having a separated control plane it is also possible to process BCP's in a different way. Considering there are no effective optical logic elements, this separation in processing is also helpful. It is simply possible to process BCP's electronically and configure switching/resources. This can be justified as, in a packet switching scenario, data and header competes with other pairs, however, if a burst and header share same channel a burst and a header competes and result may end up as burst without headers or headers without an actual burst.

address this problem. Mainly, there are two signaling modes: One-way or two-way. In one-way reservation, a BCP is sent towards the network over the electronically-switched control channel for reserving network resources for the corresponding burst without triggering any positive acknowledgements. Without having to wait for an acknowledgement from the network, the burst itself is sent towards the network over the optical-switched data channel after an offset time. The main problem with one-way signaling is that, a burst may be sent even in case network resources are not available leading to burst losses. In two-way reservation, a BCP is sent towards the network reserving resources. When resources are available at each hop towards the destination, a positive acknowledgement is sent to the ingress node after which the burst would be sent. (See Figure 3). One-way signaling provides a shorter burstification delay between control data and actual transmission; however, two-way methods provide reliability (bursts won't get dropped if they are accepted and if a burst cannot find enough resources it may wait at the ingress node instead of getting dropped). As most of the research focuses on, we also focus on one-way signaling/reservation protocols. For more detailed examples/algorithms for signaling modalities, we refer to [9].

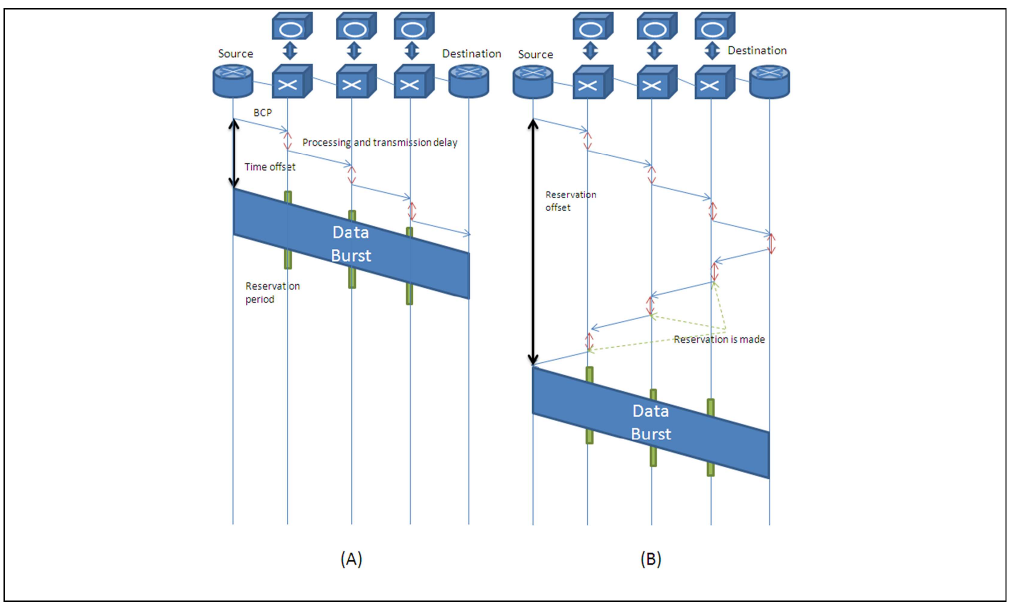


Figure 3. One-way (A) and two-way (B) signaling schemes. Reservation period also depends on the scheduling algorithm of choice.

We will focus on three popular reservation protocols in the literature: Just-in-time (JIT) [10], horizon [11] and Just-enough-time (JET) [9] (See Figure 4). In JIT, a burst is sent after a time period of its BCP, however, time offset is only enough to complete the required configurations in each OXC along its path. At an OXC, resources are allocated for a period which is equal to (or larger than in case of using guard time periods) the burst's length (in time). The last OXC before the egress node will be configured just in time (just before); hence name JIT. Horizon and JET are known as delayed reservation (DR) schemes due to the fact that an additional delay used in JIT is in effect in both of them. Usefulness of DR schemes is shown in [6]. In Horizon, whenever BCP of a burst arrives to an OXC, its resources are reserved for that burst for a period until the burst is completely switched. In systems using Horizon, there are voids which are reserved for a burst but not used for actual transmission. In JET, unlike horizon, only the time between burst's arrival to an OXC and burst's departure from same OXC is reserved for that burst at that OXC. Compared to JIT, DR schemes provide less dropping rate at OXCs, however, scheduling algorithms must take scheme-specific details to utilize resources more efficiently. For instance, if JET is being used, then scheduling algorithms must take void filling techniques into consideration. For detailed analysis and further reservation protocols see [12].

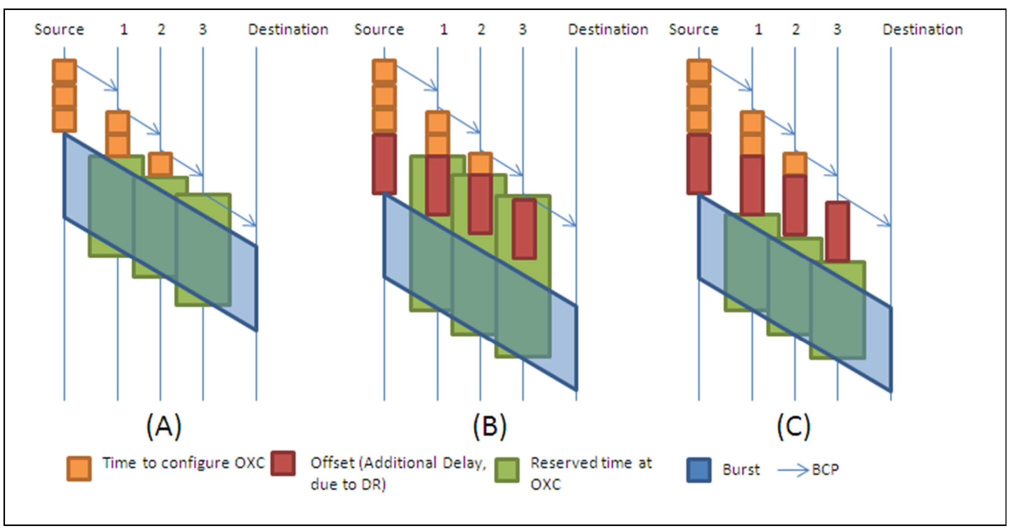


Figure 4. Illustration of one-way reservation schemes: JIT (A), Horizon (B), JET (C)

An OBS node is depicted in further detail in Figure 5 which comprises: an input interface, an output interface, a switch control unit, fiber delay lines (FDLs), wavelength converters, and most importantly, a switching fabric. The switch control unit is responsible in configuring the switching fabric according to the node's resources and information (BCPs) coming from the control plane. Switching fabric is the actual mechanism in switching data from one link to another link. In an OBS node, sometimes, we can also switch a burst to not only a different link but also to a different channel (frequency, wavelength). To achieve this conversion, OBS nodes may be equipped with wavelength converters. Design and type of the converters or classification of nodes according to their wavelength structure are important problems as well, yet they are outside our work's scope since we will assume there is always a sufficient number of wavelength converters in a node. FDLs are fiber coils which are used to delay bursts for a certain amount of time in order to resolve contention when it arises. However, throughout this thesis, OBS core nodes are assumed to have no FDLs.

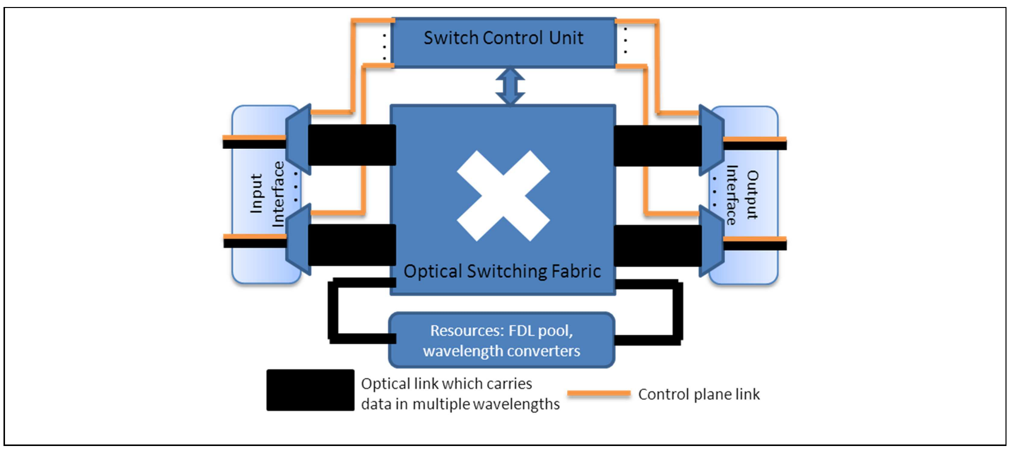


Figure 5. An OXC node.

It is possible to have contentions between bursts even though a scheduling algorithm takes resources/capabilities of a node into consideration. Due to this possibility, several contention resolution techniques are proposed. Most common contention resolution techniques are based on the use of FDLs, wavelength conversion, deflection routing, dropping, and segmentation. Detailed information regarding building blocks in an OBS node and comparison of contention resolution techniques can be found in Chapters 2 and 18 of [1].

Next, we discuss OBS scheduling algorithms. Though there are many, we will mention only three popular algorithms which attracted the attention of many researchers and they gave rise to more sophisticated schedulers proposed in the literature[6]. The first algorithm is called latest available unused channel algorithm (LAUC) [6] [9] [13] which is also called as the horizon algorithm. In this method, for each channel (or wavelength) a so-called horizon is maintained. Horizon value for a channel indicates a future time after which the channel is entirely available for newcoming bursts. When a new burst header packet arrives, a search to find available unused channels is performed. If there are none, then the packet is dropped, however, if available unused channels do not constitute an empty set, within available channels the one with the minimum time gap between the channel horizon value and the starting time of the actual burst is chosen for the newcoming burst. Consequently, the horizon parameter of the chosen channel gets updated and the new horizon value becomes the ending time of the transmission of the accepted burst (See Figure 6 and Table 2).

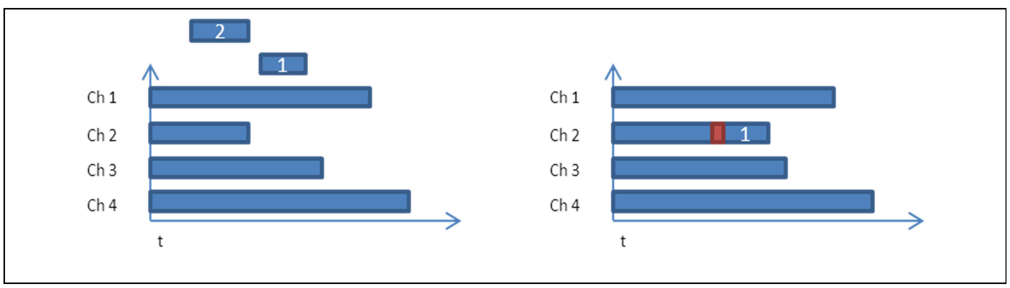


Figure 6. LAUC procedure: Arrival order of the bursts are given on them. 2nd burst gets dropped.

[6] Algorithms explained in thesis have generalizations for nodes having FDLs, however, since our work will focus on nodes with no FDLs, we will provide no FDL versions of the algorithms.

For each channel $h_n \in \mathbb{R}, n \in \mathcal{N} = \{1,2,\dots,N\}$ *denotes the horizon of the channel (which are initially 0). For an arriving burst whose BHP enters system at time t, has an offset value* δ *and length* L

1.) Find set of indices of available channels $\mathcal{I}$, $\mathcal{I} = \{\, i \mid h_i \le t + \delta \}$

2.) If $\mathcal{I} \neq \emptyset$ *then continue with 3.*

If $\mathcal{I} = \emptyset$ *then burst is dropped.*

3) Find n, *index of the channel to use for scheduling,*

$$n = \underset{l \in I_m}{\operatorname{argmin}} \; t + \delta - h_l, \text{ then update } h_n = t + \delta + L$$

Table 2. Pseudo-code of LAUC for an ***N*** channel system at time instant ***t***

The second algorithm called LAUC-VF (LAUC with void filling) [13] aims at eliminating unused voids between time offsets between burst header-burst and to minimize gaps where channel is reserved but there is no burst being relayed. To achieve this, starting and ending epochs of each burst accepted to a channel is kept track of. When a new burst header packet arrives at the node, first, available channels are found. An available channel is defined as a channel which does not have any coinciding bursts during the switching period of the burst (See Figure 7 and Table 3). If there is no such channel, the burst is dropped. In comparison, LAUC algorithm has a tendency to drop more bursts, however, LAUC-VF algorithm has a higher complexity both in search space and memory management.

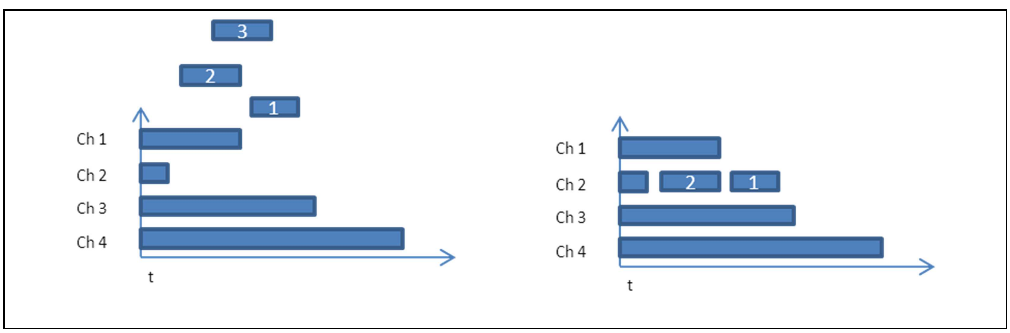


Figure 7. LAUC-VF procedure. Arrival order of the bursts is given on them. 3rd burst gets dropped

For each channel $\mathcal{H}_n \in \mathbb{R}$, $n \in \mathcal{N} = \{1,2,\dots,N\}$ *denotes a set which contains occupied time ranges* $r_{n,j} = \left(s_{n,j}, e_{n,j}\right)$, $j \in \mathbb{Z}^+ \cup \{0\}$ *(* $\mathcal{H}_n = \bigcup_j r_{n,j}$ *and initially all* $\mathcal{H}_n = \emptyset$*). For an arriving burst whose BHP enters system at time t, has an offset value* δ *and length* L *1.)Find set of indices of available channels, namely* $\mathcal{I}$, $\mathcal{I} = \{i \mid \mathcal{H}_i \cap (t+\delta, t+\delta+L) = \emptyset\}$ *2) If* $\mathcal{I} \neq \emptyset$ *then continue with 3.* *If* $\mathcal{I} = \emptyset$ *then burst is dropped* *3) For every available channel find the previous burst before the burst to be scheduled. Then schedule the burst in the channel that minimizes gap between two mentioned bursts.* *For every element of* $\mathcal{I}$ *find a* $j_l = Max\left(\{j \mid e_{l,j} \leq t+\delta\}\right)$, $(l \in \mathcal{I})$ *Then find* $n = \underset{l \in \mathcal{I}}{argmin}\ t+\delta-e_{l,j_l}$, *and update* $\mathcal{H}_n = \mathcal{H}_n \cup (t+\delta,\ t+\delta+L)$

Table 3. LAUC-VF algorithm run at the instance of a new burst arrival for an OBS system with N channels

The final algorithm we discuss is the so-called group LAUC [13] scheduling is obtained out of a trade-off between LAUC and LAUC-VF. In this method, arrivals of multiple BHPs are queued and then the LAUC algorithm is applied for accumulated bursts consecutively starting with the earliest burst to be switched (See Table 4 and Figure 8).

After predetermined period is over LAUC is used with the burst which has the earliest beginning among all. Then with the updated system LAUC is used repeatedly until all of the grouped bursts are used in this procedure

Table 4. group LAUC algorithm

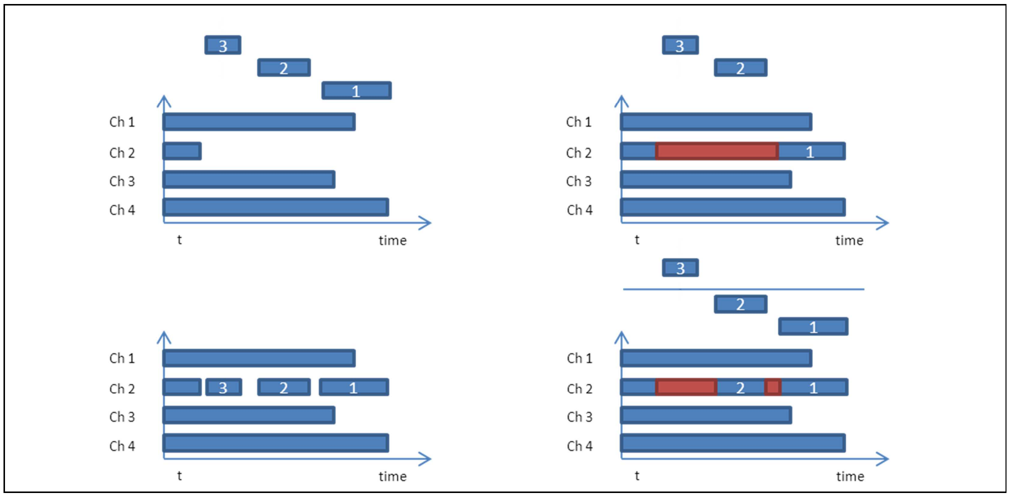


Figure 8. Comparison of LAUC(top right), LAUC-VF(bottom left) and group LAUC (bottom right) for a given scenario(top left) in a no FDL setting. Arrival order of the bursts are given on them. For group LAUC scenario, 3rd burst is in another group than first two.

Detailed comparison of the three above-mentioned scheduling algorithms can be found in [14].

1.2 A Brief Overview of QoS in OBS Networks

It is obvious that clients using communication networks via different applications for different purposes have difference performance expectations from the network. Bandwidth, error rate, loss rate, delay, jitter and packet ordering can be listed as a subset of performance criteria. QoS (quality of Service) provisioning is simply a set of rules and methods to meet required performance criteria. It is possible to group QoS provisioning techniques into two categories: Absolute QoS and QoS differentiation (or relative QoS) methods [15] . Absolute QoS methods provide end-to-end strict guarantees if the client does not violate a contract established between the client and the provider. For instance, an end-to-end delay bound or a guarantee regarding minimum value of provided bandwidth. In differentiation methods, traffic class-based preferential treatment is provided by the network. In this context, a traffic class is defined as set of client packets which would receive the same treatment from the network.

Simply, performance of a traffic class is determined not solely by its own but also the performance encountered by others. For example, assume there are two QoS classes, namely, high priority (HP) and low priority (LP). HP class may experience a lower rate of packet loss compared to LP class. Usually, providing absolute QoS requires more complex systems. For today's conventional electronic networks, both techniques are well-researched and differentiation usually depends on queuing and scheduling techniques using electronic buffers (RAM); hence these methods cannot be expanded to the OBS paradigm directly because of not having optical buffers. In OBS, both absolute and relative QoS mechanisms exist. In [15], a classification of relative QoS mechanisms in OBS is given and [16] is pointed out as a source for absolute QoS mechanisms. Table 1 from [15] is actually quite informative and self-explanatory regarding classifications. In the first group of algorithms, called packet dropping mechanisms, lower class packets are dropped in favor of higher class packets. In scheduling-based differentiation, after waiting a time period for requests, requests are ordered in such a way that higher classes packets get preferential treatment when they grasp system resources. In hybrid signaling methods, both one-way and two-way reservation protocols are used for classes for different needs. In bit-rate differentiation, bit rates are provided by observing accepted bursts and enforcing rates in favor of higher classes (or according to other fairness criteria). The simplest, yet the most popular two are now summarized. In the burst length-based differentiation (BLD) [17] , the main idea is that a shorter burst can find available resources more easily[7] (see Figure 9). At the ingress node, it is possible to use different parameters for different QoS classes in burst assembly algorithm. Using shorter bursts for the HP class also means shorter end-to-end delay due to the fact that burst assembly takes shorter, however, using shorter bursts also mean more overhead due to control packets.

[7] When burst separation is not in use, in which bursts can be divided in parts and carried in different channels or some of the partitions are even dropped.

Moreover, BLD requires more complex algorithms at the edge and core nodes for burst assembly and scheduling[8], respectively.

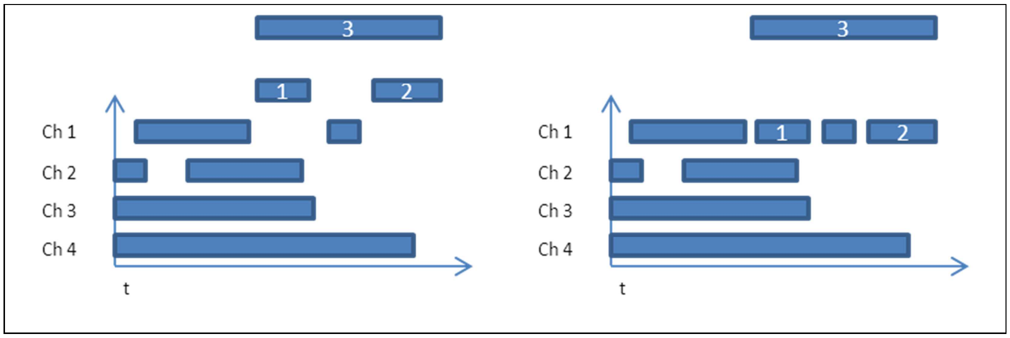


Figure 9. Idea behind BLD. 1,2,3 are bursts waiting for reservation

In offset time-based differentiation (OTD) [18], an additional delay between sending a BCP and its burst is introduced. By introducing this additional delay, the BCP arrives at the node earlier and reserves resources; thus experiences a lower blocking rate (Figure 10). It is also possible to explain this phenomenon from the perspective of the system. At a fixed time instant, available channels in a node is a non-decreasing function of offset which can be interpreted as it is more likely to reserve system resources using larger offset times (Figure 11). Longer additional delays are used by higher priority classes. Advantage of this system is its simplicity; there is need to change core node mechanisms. Drawbacks of this approach can be listed as: increased end-to-end delay for the HP class, lower throughput due to increased delay and sensitivity to burst lengths. In our work, our modeling efforts will be concentrated towards the OTD mechanism.

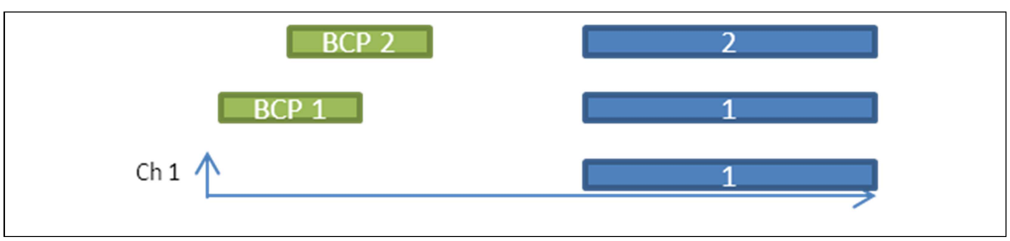


Figure 10. OTD: Earlier arriving BCP reserves the channel for its burst

[8] BLD to be effective, a void filling scheduling method is required.

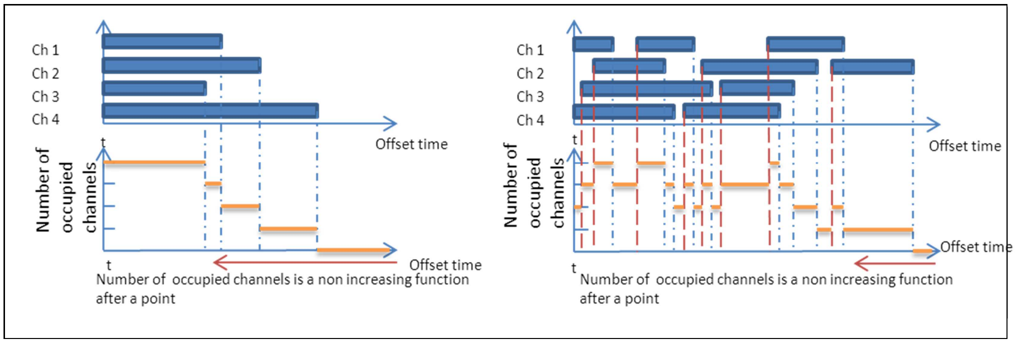


Figure 11. Regarding availability of resources wrt. offset time. Left: A system using horizon. Right: A system using void filling.

1.3 Related Work and Overview of the Thesis

Our work in this area as an effort to analytically model blocking probabilities of different QoS classes that employ OTD (offset time difference) mechanism at an output port of a core node using LAUC. In this case, LAUC algorithm is preferred since it has received much attention due to its simplicity compared to other algorithms. Although in thesis, we only propose an analysis using a relaxation of LAUC, we believe it is still helpful to review work done on modeling QoS characteristics of a node using OTD and LAUC.

The earliest work [18] assumes complete class isolation and uses an M/M/k/k model which is not sensitive to offset time difference between classes and model becomes accurate only in high load scenarios[9]. A more recent work in [19] assumes Poisson arrivals and discretized system to work on boundaries of small time steps and assumes discrete-finite pmfs for offset and service times. Although it may give relatively accurate results, the method's accuracy depends on how small the time step size is in discretization. The drawback of this model is its relatively higher computational complexity. The proposed model cannot be used for large number of QoS classes or channels using time steps small enough to get accurate results. Its complexity becomes comparable to that of discrete event simulation. In [20] an analytical approach is developed to find an exact

[9] It is sometimes referred as conservation law in literature

expression for the distribution of the number of bursts that contend with an arriving burst. The model is applicable to systems in which each class has an arbitrary burst-length distribution and an arbitrary offset size. However, the method approximates blocking probability of only the premium QoS class. The references [21] and [22] provide an exact solution of the single channel problem with Poisson arrivals, deterministic offset values and phase type burst length distributions by using (Markovian modulated) first order multi-regime feedback fluid queues (FMFFQs). Unfortunately, this method cannot be generalized to multiple channels.

In this thesis, which can be seen as a first step to analyze QoS performance of an OBS node using OTD and LAUC, we propose an iterative method to approximate blocking probabilities of a node which uses OTD and a slightly relaxed version of LAUC (which is called basic LAUC or bLAUC). Moreover, we assume the OBS node has sufficiently many wavelength converters but is not equipped with any FDLs. Furthermore we also assume that BHP arrival process is Poisson, offset times are deterministic and burst lengths are exponentially distributed with mean 1 for each QoS class. Our iterative method aims at providing approximations with relatively low complexity compared to discrete event simulations and insensitivity of the complexity of the model to increasing number of channels (wavelengths) and linear dependence of the complexity to the number of overall traffic classes.

The proposed method models multiple dependent channels with a different number of independent channels and solves marginal steady-state distribution of an independent single channel instead of solving the complete system[10] and makes use of [22] and a method, which will be described in Chapter 3, to model multiplexing effects due to having multiple channels. Our contribution can be summarized as presenting an analytical model to approximate per-class loss probabilities for a system described by multiple FMFFQs with setting described above. This is important in the sense that there are no proposed analytical

[10] As it can be guessed non-increasing in number of channels is due to this step (Modeling identical but dependent random variables by using independent identical variables).

approaches for similar problems which can also be described by coupled multiple FMFFQs.

We now describe bLAUC before continuing on. When a BHP arrives at the node, channels whose horizon is between offset value of class of arriving burst and offset value of the next lower priority class are listed. If this list is nonempty, search is terminated, else channels whose horizon is within the offset value of next lower priority class and the next one are listed, until getting a nonempty list in the end. Then from available channel candidates, one is selected randomly and its horizon is updated to the end of the burst. If there are no available channels after this search, the burst is dropped. How bLAUC operates in a no-FDL scheme is depicted in Figure 12. Dashed vertical lines indicate offset values of different classes. The rectangle on the top right is a class 5 burst which tries to enter the system. Channels 1 through 3 are available. If it was LAUC, Ch. 1 would be the one to accommodate the burst since it is the latest available unused channel. bLAUC's algorithm checks whether there is any available channel which is reserved until a point between δ_4 and δ_5 , however, there is none. So, next, whether there is any available channel within δ_3 and δ_4 is to be checked. There are two available channels, namely, Ch. 1 and Ch. 3. One of these two will be chosen randomly (with probability 1/2). In this example Ch. 3 gets chosen. However, it causes a larger gap (in red) compared to choosing Ch. 1; hence, it is possible to conclude that bLAUC is less efficient in using system resources than LAUC; also see Table 5.

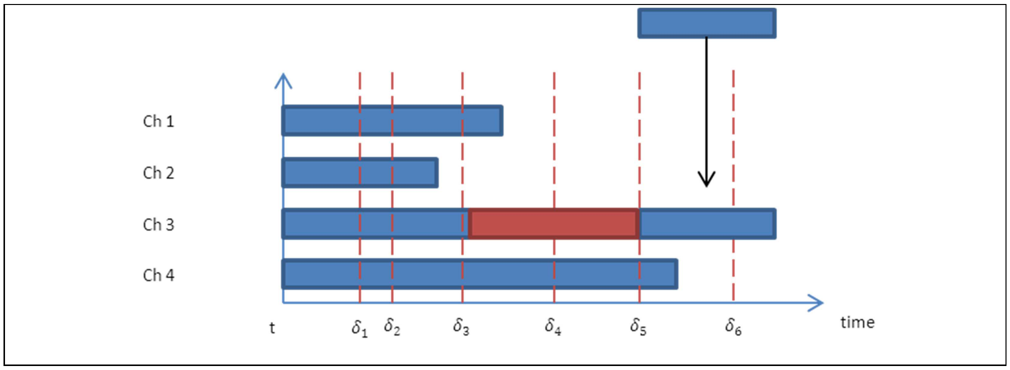


Figure 12. Illustration of inner-working of bLAUC

For each channel $h_n \in \mathbb{R}, n \in \mathcal{N} = \{1,2,\ldots,N\}$ *denotes horizon of the channel (which are initially 0). There are P OTD QoS classes with offsets* $\delta_1, \delta_2, \ldots, \delta_P$ $(\delta_0 = 0)$. *An arriving burst whose BHP enters system at time t, has an offset value* δ_p $(p \in \mathcal{P} = \{1,2,\ldots,P\})$ *and length* L

1.) Find set of indices of available channels

$\mathcal{I} = \{i \mid h_i \leq t + \delta_p\}$

2.) If $\mathcal{I} \neq \emptyset$ *then continue with 3.*

If $\mathcal{I} = \emptyset$ *then burst is dropped*

3.) Start with $\hat{p} = p$

4.) Find set of indices of available channels within following offset times

$\mathcal{I}_{\hat{p}} = \{i \mid h_i \in [t + \delta_{\hat{p}-1}, t + \delta_{\hat{p}}]\}$

If $\mathcal{I}_{\hat{p}} = \emptyset$ *then update* $\hat{p} = \hat{p} - 1$ *and go to step 4*

Else choose one of the elements of $\mathcal{I}_{\hat{p}}$ *in a uniformly random way. Call selected one* n *and then Update* $h_n = t + \delta + L$

Table 5. Pseudo-code of bLAUC for a $\boldsymbol{N}$ channel system which sees $\boldsymbol{P}$ traffic classes which have offsets $\boldsymbol{\delta_1}, \boldsymbol{\delta_2}, \ldots, \boldsymbol{\delta_P}$ and a class p burst which has length $\boldsymbol{L}$ arrives at time instant $\boldsymbol{t}$

As it can be seen in Figure 13 (system load is 0.5 and it is divided evenly between classes), bLAUC causes more drops compared to LAUC. Furthermore, its computational complexity is no less than LAUC. However, its analysis is more tractable which will be made clear in the next two chapters of the thesis.

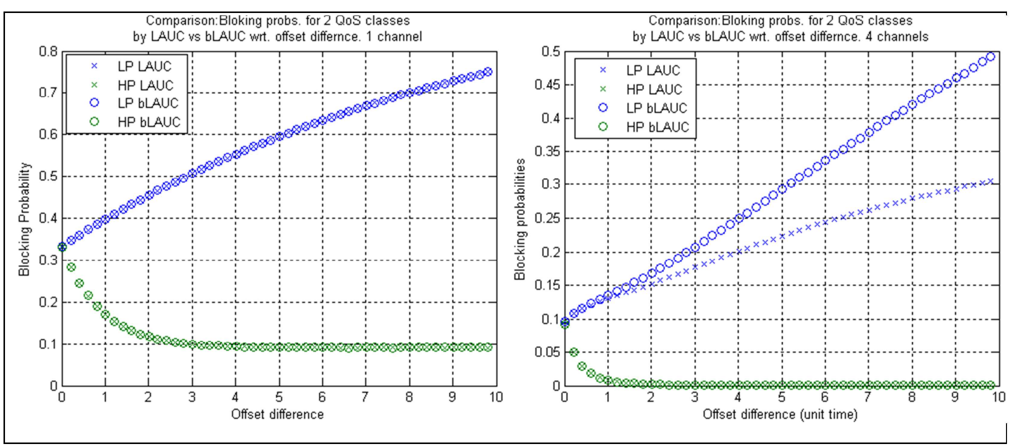


Figure 13. Comparison of bLAUC and LAUC in an OTD scenario. Left: 1 Ch, Right: 4 Ch. case

The outline of the thesis is as follows. In Chapter 2, an overview of existing methods for the analysis of blocking probabilities for a node using LAUC[11] and OTD in single channel-case by using FMFFQs will be provided. The traffic assumptions are Poisson arrivals (also allowed to be regime-dependent arrival rates) and exponentially distributed burst lengths. In Chapter 3, after giving some motivation, a novel method to model the multiplexing effect in multi-channel systems is introduced so as to extend the analysis of single-channel OBS to multi-channel OBS. This analysis is based on an entirely novel iterative process[12]. In Chapter 4, numerical results will be given to validate the approximate stochastic model. The final chapter is devoted to conclusions and future work.

[11] For single channel case bLAUC and LAUC are identical methods since if the single channel is available it is the latest available one and it's the only one which can be chosen "randomly". See Figure 13

[12] To model blocking probabilities of OTD QoS classes on a tagged output port of a node using bLAUC

Chapter 2

Single-channel Multi-class OBS

In this chapter, FMFFQs will first be presented. The references [23] and [21] are the two recommended resources for detailed description of FMFFQs regarding derivation of structure of the solution and boundary conditions to be used in obtaining the exact solution, respectively. After giving notation and introducing FMFFQs, analysis of the single channel LAUC OTD QoS problem will be given for exponentially distributed service times (Remember for single channel case, LAUC and bLAUC are exactly the same, see Figure 13). Finally, same problem will be considered with regime specific arrival rates.

2.1 Brief Overview of FMFFQs

FMFFQs are described by a joint Markovian process $\{\mathrm{C(t)}, \mathrm{M(t)};\ \mathrm{t}\ \geq\ 0\}$ where $\{\mathrm{C(t)};\ \mathrm{t}\ \geq\ 0\}$ indicates the fluid level at time t and $\{M(t);\ t\ \geq\ 0\}$ is an underlying continuous-time Markov chain that determines the net rate at which the buffer content $C(t)$ changes. The latter process $\{M(t);\ t\ \geq\ 0\}$ is often called the background or the modulating process of the Markov fluid queue. Thresholds which determine regimes are identified as $0 = T^{(0)} < T^{(1)} < \cdots < T^{(K)} = \infty$. System is said to be in regime k ($k \in \mathcal{K} = \{1, \dots, K\}$) if $T^{(k-1)} < C(t) < T^{(k)}$ and at threshold $T^{(k)}$ if $C(t) = T^{(k)}$. When the system is in regime k modulating process $M(t)$ which has a finite state space ($\mathcal{M} = \{1, .., M\}$) is characterized by infinitesimal generator $Q^{(k)}$ (when at threshold T_k then by

$\tilde{Q}^{(k)}$). The drift of the system ($dC(t)/dt$) is given by $r_m^{(k)}$ when system is in regime k ($\tilde{r}_m^{(k)}$ when system is at threshold $T^{(k)}$) and modulating process is at state m ($m \in \mathcal{M}$). Rate matrices are defined as $R^{(k)} = diag(r_1^{(k)}, r_2^{(k)}, \dots, r_M^{(k)})$ and $\tilde{R}^{(k)} = diag(\tilde{r}_1^{(k)}, \tilde{r}_2^{(k)}, \dots, \tilde{r}_M^{(k)})$.

Considering $0 \le C(t) < \infty$ it is possible to conclude:

$$\frac{dC(t)}{dt} = \begin{cases} \max\left(0, \tilde{r}_{M(t)}^{(0)}\right), & if\ C(t) = T^{(0)} = 0 \\ r_{M(t)}^{(k)}, & if\ T^{(k-1)} < C(t) < T^{(k)} \\ \tilde{r}_{M(t)}^{(k)}, & if\ C(t) = T^{(k)}, k \neq K \\ \min\left(0, \tilde{r}_{M(t)}^{(K)}\right), & if\ C(t) = T^{(K)} \end{cases}$$

Furthermore, instantaneous distribution functions are given as:

$F_m^{(k)}(y,t) = P(C(t) \le y, M(t) = m), y \in \left(T^{(k-1)}, T^{(k)}\right) k \in \mathcal{K}\ and\ m \in \mathcal{M}.$

And $f_m^{(k)}(y,t) = \frac{\partial F_m^{(k)}(y,t)}{\partial y}, \quad y \in \left(T^{(k-1)}, T^{(k)}\right) k \in \mathcal{K}\ and\ m \in \mathcal{M}.$

Hence row vector of transient joint probability density functions at time t in regime k is:

$f^{(k)}(y,t) = [\ f_1^{(k)}(y,t)\ f_2^{(k)}(y,t)\ f_3^{(k)}(y,t) \dots f_M^{(k)}(y,t)\ ]$

Then, steady-state distribution functions are:

$f_m^{(k)}(y) = \lim_{t\to\infty} f_m^{(k)}(y,t)$ and $F_m^{(k)}(y) = \lim_{t\to\infty} F_m^{(k)}(y,t)$

$f^{(k)}(y) = \left[\ f_1^{(k)}(y)\ f_2^{(k)}(y)\ f_3^{(k)}(y,t) \dots f_M^{(k)}(y)\ \right],$

$F^{(k)}(y) = [\ F_1^{(k)}(y)\ F_2^{(k)}(y)\ F_3^{(k)}(y,t) \dots F_M^{(k)}(y)\ ]$

And finally $f_m(y) = f_m^{(k)}(y)\ and\ F_m(y) = F_m^{(k)}(y)\ \ if\ \ y \in \left(T^{(k-1)}, T^{(k)}\right)$

In a similar way we can define mass accumulations at thresholds.

$c_m^{(k)}(t) = P\left(C(t) = T^{(k)}, M(t) = m\right),\ k \in \{0,1,2,\dots,K\}\ \ m \in \mathcal{M}$

$c^{(k)}(t) = [c_1^{(k)}(t)\ c_2^{(k)}(t) \dots\ c_M^{(k)}(t)],\ k \in \{0,1,2,\dots,K\}$

$c_m^{(k)} = \lim_{t\to\infty} c_m^{(k)}(t)\ \ and\ c^{(k)} = \left[c_1^{(k)}\ c_2^{(k)} \dots\ c_M^{(k)}\right],\ k \in \{0,1,2,\dots,K\}$

$, m \in \mathcal{M}$ (See

Figure 14 for a visualization of defined variables.).

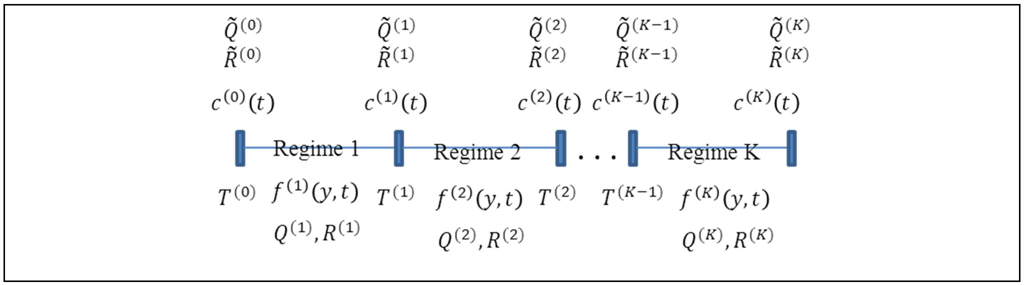


Figure 14. FMFFQ's structure

Solution to this system (finding $c^{(k)}$ and $f_m(y)$ or $F_m(y)\ for\ \forall m \in \mathcal{M}$) is given in [23] and [24]. Moreover, boundary conditions and a numerically stable computation method are given in [21].

2.2 Single Channel OBS LAUC Scheduling Problem

As it is stated before, problem under spotlight is determining per-class blocking probability on a tagged single channel output port of an OBS node using LAUC. In this system, bursts belong to one of the classes in the class-set $\mathcal{I} = \{1,2,\ldots,I\}$. Furthermore, arrivals of BHPs of class i are experienced as a Poisson process with rate λ_i by the tagged output port. BHPs of class i arrive to the system δ_i time units before their corresponding bursts. Length of a burst of class i is characterized by an exponential random variable with rate μ_i. If the channel horizon is less than or equal to δ_i at the arrival of the class i BHP, then burst gets scheduled, otherwise it is not accepted to the system and it is dropped. The channel consumes jobs with unity rate. Throughout this thesis, if the number identifying a class is larger than another one, then it is assumed the class with larger identifier has larger priority compared to the other. This implies $if\ i < j\ then\ \delta_i < \delta_j$. Figure 15-a illustrates problem mentioned above, for a three classes scenario, $H(t)$ denotes channel horizon.

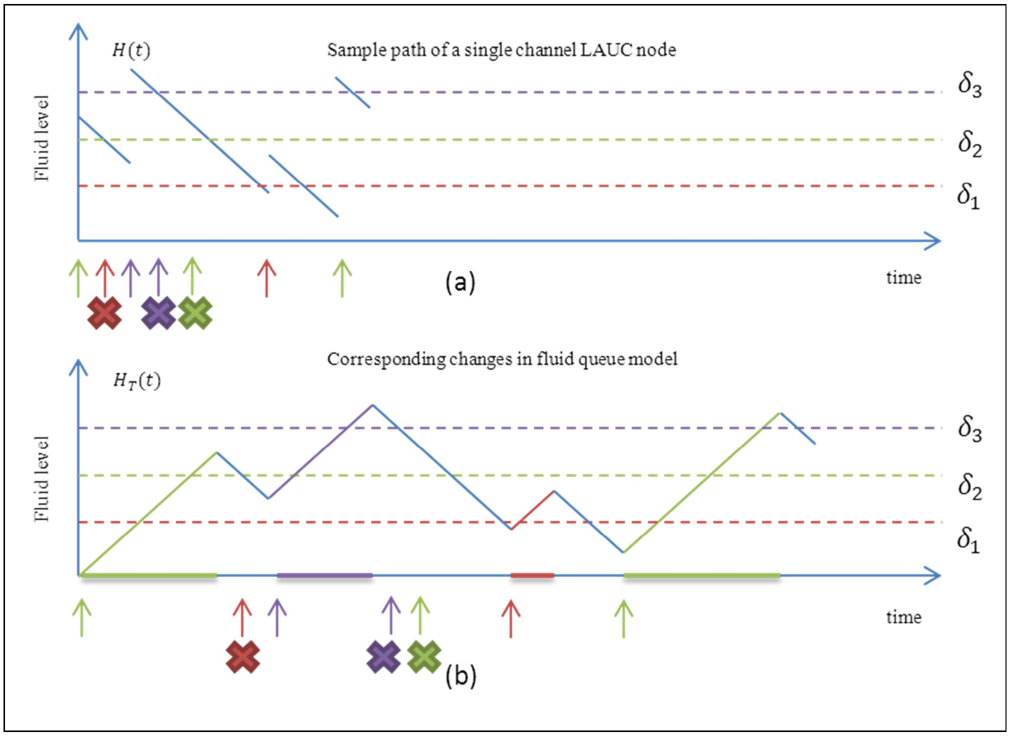


Figure 15. (a) A sample path of a single channel system with 3 traffic classes which uses LAUC and (b) How fluid queue models its behavior using a transformation.

As is mentioned in [22], FMFFQs cannot model behavior of $H(t)$ in Figure 15-a due to delta function like jumps at arrival epochs. Instead, FMFFQs can solve steady-state behavior of transformed process, in which jumps are replaced by linear increases (See $H_T(t)$ in Figure 15-b), then solution of original process can be found. Problem parameters can be defined by inheriting notation from previous parts. Class-set is $\mathcal{I} = \{1,2,3,\dots,I\}$, then $I+1$ regimes can be identified by $I+2$ thresholds in following setup:

$$T^{(0)} = 0 < T^{(1)} = \delta_1 < T^{(2)} = \delta_2 < \cdots < T^{(I)} = \delta_I < T^{(I+1)} = \infty$$

State space of $M(t)$ consists of elements of $\mathcal{M} = \{1,2,\dots,I,P = I+1\}$. First I elements denote states in which $H_T(t) = C(t)$ is increasing due to an arrival of that class. pth state $P = I+1$ denotes the state in which $H_T(t)$ decreases with boundary 0. In this setup, if $M(t) = m \neq P$ then $M(t)$ stays in state m until $H_T(t)$ reaches δ_m. When $H_T(t) > \delta_m$ $M(t)$ tries to have a transition from m to P with rate μ_m. Moreover, when $M(t) = P$ and $H_T(t) \leq \delta_m$, $M(t)$ transitions from P to m with rate λ_m. So it is possible to define drifts and infinitesimal generators as following:

$r_p^{(k)} = -1, k \in \{1,2,\dots,I,I+1\}$

$\tilde{r}_p^{(k)} = -1, k \in \{1,2,\dots,I,I+1\}$

And $\tilde{r}_p^{(0)} = 0$

$r_m^{(k)} = 1, m \in \{1,2,\dots,I\}\ and\ k \in \{1,2,\dots,I+1\}$

$\tilde{r}_m^{(k)} = 1, m \in \{1,2,\dots,I\}\ and\ k \in \{0,1,2,\dots,I+1\}$

Let $q_{i,j}^{(k)}$ denote (i,j)th element of $Q^{(k)}$ and $\tilde{q}_{i,j}^{(k)}$ denote (i,j)th element of $\tilde{Q}^{(k)}$

$$\tilde{q}_{i,j}^{(k)} = q_{i,j}^{(k)} = \begin{cases} -\mu_i, & if\ 1 \le i \le k-1\ and\ i = j \\ \mu_i, & if\ 1 \le i \le k-1\ and\ j = p = I = 1 \\ \lambda_j, & if\ i = p\ and\ k \le j \le I = p-1 \\ -\sum_{j=k}^{p-1} \lambda_j, & if\ i = p\ and\ j = p \\ 0, & otherwise \end{cases}$$

And $\tilde{q}_{i,j}^{(0)} = q_{i,j}^{(1)}\ \forall (i,j)$. Where $k \in \{1,2,\dots,p\}$. Figure 16 may be helpful to understand structure of infinitesimal generators. Defining parameters is completed with this step.

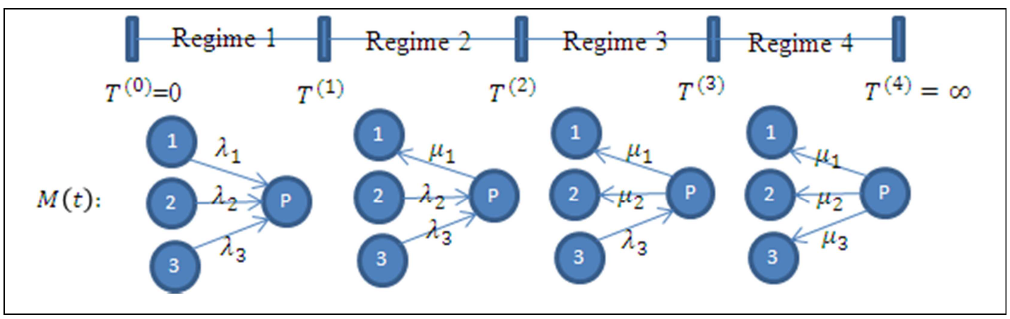


Figure 16. Structure of M(t) wrt. regimes in a example with 3 QoS classes

As mentioned in [24] solution is of following form:

$$\frac{df^{(k)}(y)}{dy} R^{(k)} = f^{(k)}(y) Q^{(k)}, for\ k \in \{1,2,\dots,I+1\}$$

Additionally, required boundary conditions can be derived using given boundary conditions in [21] which are derived in a more general conjuncture.

$c_m^{(k)} = 0, for\ all\ m\ and\ k\ except\ for\ k = 0\ and\ m = p, (c_p^{(0)} \neq 0)$

$f^{(k+1)}\left(T^{(k)}\right)R^{(k+1)} - f^{(k)}\left(T^{(k)}\right)R^{(k)} = c^{(k)}\tilde{Q}^{(k)}, k \in \{1,2, \dots, I\}$

$f^{(0)}(0)R^{(1)} = c^{(0)}\tilde{Q}^{(0)}$ and $f^{(I+1)}\left(T^{(I+1)} = \infty\right)R^{(I+1)} = 0$

$$\sum_{m=1}^{M}\left(\sum_{k=1}^{I+1}\int_{T^{(k-1)}}^{T^{(k)}} f_m^{(k)}(y)dy + \sum_{k=0}^{K} c_m^{(k)}\right) = 1$$

This concludes solution system to characterize steady state behavior of $H_T(t)$, however, we are to find behavior of $H(t)$. If we condition fluid queue to only P(decreasing state) then steady state behavior of $H(t)$ can be found since the process after removing linear increases in $H_T(t)$ is exactly same as $H(t)$. Let $F(y)$ denote steady state behavior of the process $H(t)$, then $F(y) = \lim_{t\to\infty} P(H(t) \leq y) = F_p(y)/F_p(\infty)$. Due to having Poisson arrivals it is possible to use PASTA property and compute blocking probabilities. Scheduling requests of class i is rejected with probability $PB_i = 1 - F(\delta_i)$.

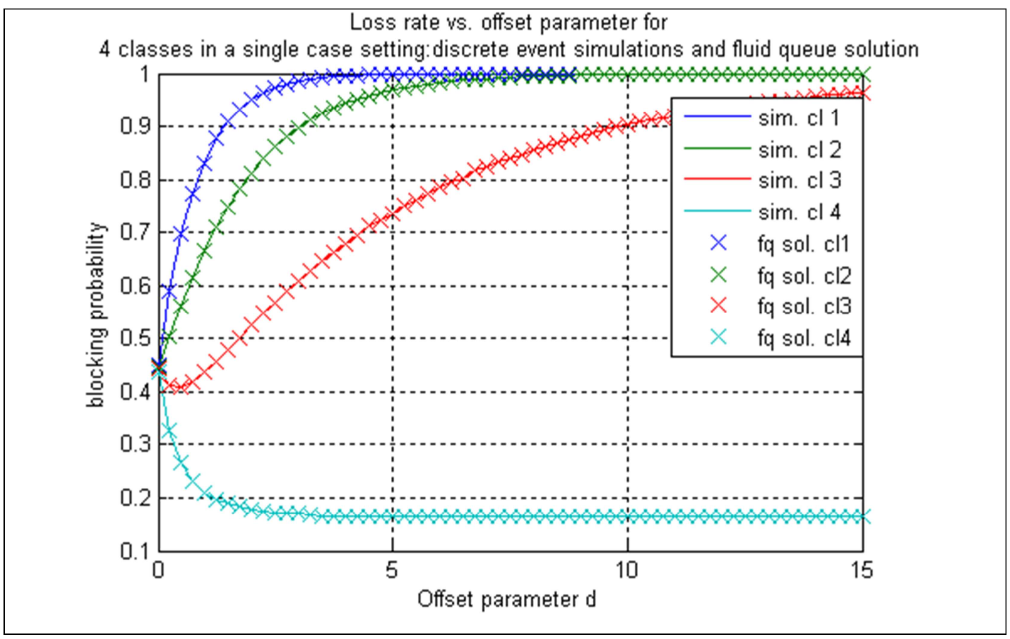


Figure 17. Comparison of fluid queue solution and discrete event simulations

Simulation results and fluid queue based solution of a single channel output port experiencing arrivals from 4 QoS classes is given in Figure 17. Total load experienced by the output port is 0.8 and distributed evenly between classes.

Furthermore, offsets are characterized by delay parameter d in such a scheme: $\delta_1 = 0, \delta_2 = d, \delta_3 = 2d, \delta_4 = 3d.$

2.3 Single Channel OBS LAUC Scheduling Problem with Regime Specific Arrival Rates

In our work, we will need to find solution to same problem but with regime specific arrival rates. Let $\lambda_i^{(k)}$ denote arrival rate of class i offered to the system when fluid level is in regime k or at $T^{(k)}$ ($i \in \{1, \dots, I\}$, $k \in \{0,1,2, \dots, I+1\}$, $k = 0$ case corresponds to only being at $T^{(0)}$). Solution system, boundary conditions and parameters defining the problem remains same except for definitions of infinitesimal generators. For $k \in \{1,2, \dots, I, I+1\}$ infinitesimal generators with regime specific rates can be summarized as:

$$\tilde{q}_{i,j}^{(k)} = q_{i,j}^{(k)} = \begin{cases} -\mu_i , & if\ 1 \le i \le k-1\ and\ i = j \\ \mu_i , & if\ 1 \le i \le k-1\ and\ j = p = I+1 \\ \lambda_j^{(k)}, & if\ i = p\ and\ k \le j \le I = p-1 \\ -\sum_{j=k}^{p-1} \lambda_j^{(k)} , & if\ i = p\ and\ j = p \\ 0 , & otherwise \end{cases}$$

Furthermore,

$$\tilde{q}_{i,j}^{(0)} = \begin{cases} \lambda_j^{(0)} , & if\ i = p\ and\ 1 \le j \le I = p-1 \\ -\sum_{j=k}^{p-1} \lambda_j^{(0)} , & if\ i = p\ and\ j = p \\ 0 , & otherwise \end{cases}$$

Solution structure and boundary conditions remains same as in previous subsection. Blocking probabilities are also computed exactly in the same manner: Using notation in previous subsection and PASTA property $PB_i = 1 - F(\delta_i)$

As it will be made clear in the following chapter, solution of this problem will have a crucial to our approximation of multichannel bLAUC problem.

Chapter 3

Multi-channel Multi-class OBS

In this chapter, an iterative procedure to approximate per class blocking probabilities on a multiple channels output port of an OBS node using bLAUC will be given. To do so, first we will visit M/M/c/c (c=N for our problem) systems due to the fact that blocking probabilities experienced in our problem can be explained by Erlang loss systems in some extreme cases. We will provide an iterative solution to blocking rates experienced by an M/M/c/c system, whose solution is already well-known, in this subsection. While developing this iterative solution, we will introduce a concept that we call "effective number of independent servers". Inheriting structure of the iterative procedure and idea of the "effective number of independent servers", we will give the iterative procedure for bLAUC scenario in the next subsection.

3.1 An Iterative Procedure to Model M/M/c/c Systems and Idea of Effective Number of Servers

Unfortunately, generalizing the methods given in Chapter 2 to multiple dependent fluid queues is not trivial at all. In fact, there is no work done which has achieved providing an exact solution or even a framework to approximate solution, till this day. Only work can be addressed as similar is [25], in which tandem systems are investigated. Furthermore, it is also impossible to evaluate performance of multichannel systems using advance reservation schemes as ordinary Erlang loss systems since using time offsets violates stochastic nature of the problem. However, for the limiting cases in which system observes traffic as it comes from a single class, it is possible to model system's behavior by using Erlang loss systems. A subset of examples of these limiting cases can be summarized as: When total load to the system is dominated by a class, when offsets between classes are negligible, when a class is known to be isolated from other classes (experiencing very low blocking probabilities due to other classes, i.e. having time offset comparable to infinity). In these mentioned limiting cases, blocking characteristics of bLAUC in a multiple class environment can be modeled by an M/M/c/c system as it is represented in Figure 18. Parameters in this example are simply: Load is 0.5 and distributed evenly between classes, number of channels is $N = 2$ and there are two QoS classes. Blocking probability when offset difference is negligible which is denoted as "x" in the figure corresponds to $P_L = ErlangB\left(N, \frac{\lambda}{N}\right)$ (λ denoting sum of all arrival rates) and square symbol corresponds to loss probability of HP class when offset difference between classes is very large and it is equal to $P_{Lp} = ErlangB\left(N, \frac{\lambda_p}{N}\right)$ (λ_p denotes arrival rate of the premium class).

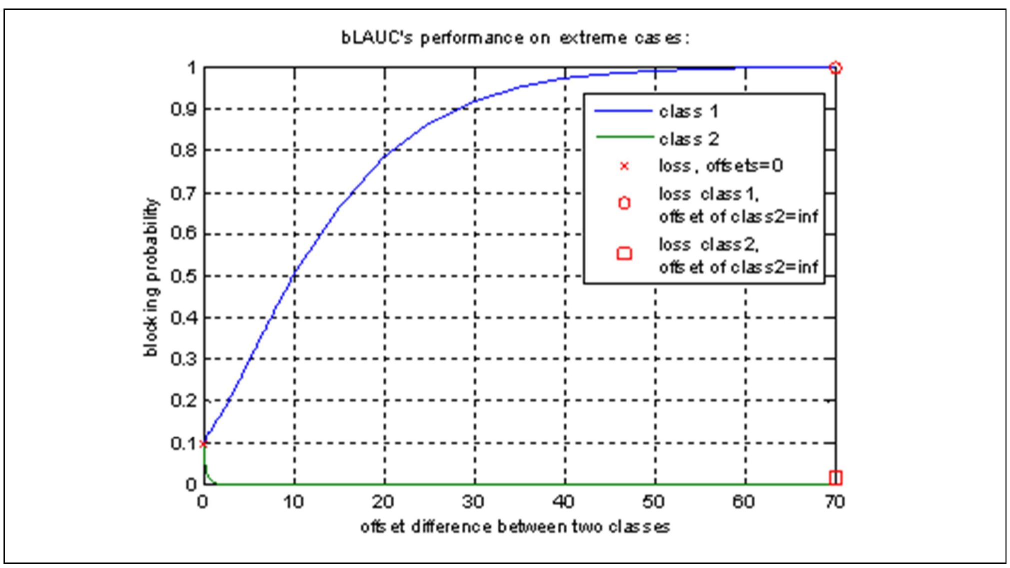


Figure 18. bLAUC's blocking behavior in a 2 channel 2 classes setting wrt. offset difference and its behavior on limiting cases

Due to these limiting case behaviors, modeling multiplexing effect of an M/M/c/c system by a number of independent M/M/1/1 systems and obtaining same loss probability would be helpful, even though M/M/c/c system has a well-known solution.

Notice a single server in M/M/c/c system is actually an M/M/1/1 server and experiences same loss rate as overall system. This implies that loss probability of an M/M/c/c system can be obtained by using an M/M/1/1 model. However, this approach requires finding arrival rates experienced by an M/M/1/1 server both when it is idle and when it is busy (See Figure 19). Finding the mentioned arrival rates is a non-trivial task, since somehow it is needed to reflect effects of having multiple servers and multiplexing of jobs between them[13]. Even though, all servers in an M/M/c/c system are identical, they are also dependent. Reflecting effect of multiplexing is non-trivial because of this dependency. Next, we will try to model multiplexing effects by using a certain number of independent servers.

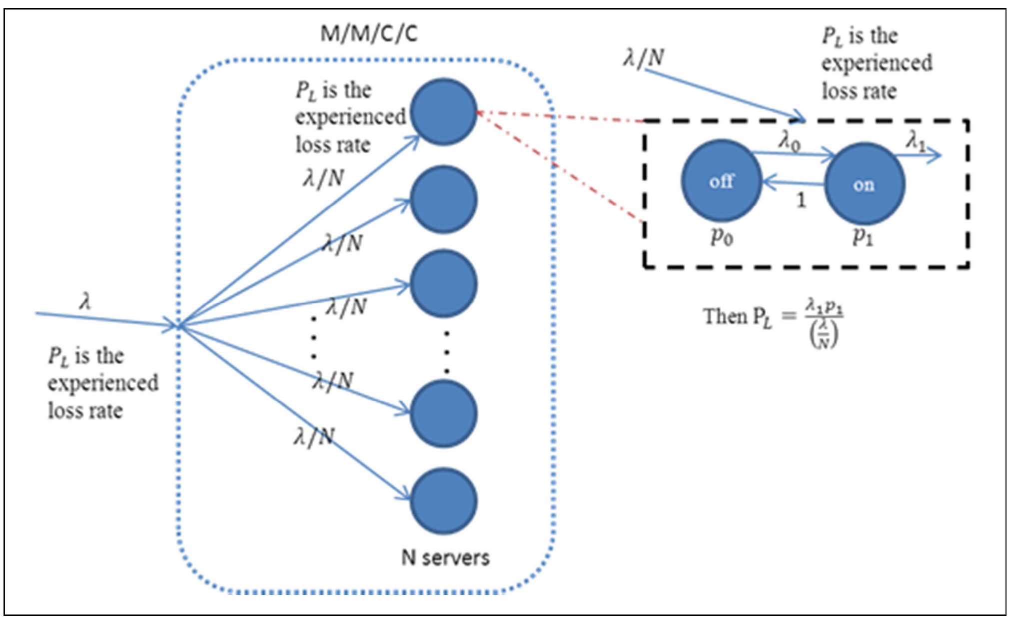


Figure 19. An M/M/1/1 server in an M/M/c/c system (Using Kendall notation, Markovian arrivals and job lengths, c servers and no buffers).

[13] In fact this is why an M/M/c/c model is being used.

Consider an M/M/c/c system which has a loss probability $P_L = ErlangB(N, \lambda/N)$, with offered load λ, exponentially distributed service times with mean 1 and N servers. Now consider an M/M/1/1 server which is called tagged server (Figure 20) and will be used to obtain the same loss probability[14][15]. Server has two states: On (state 1, busy) and off (state 0, idle). In the off state server is idle and having arrivals with rate λ_0 . In the on state server is occupied and arrivals -which cannot be accommodated by the tagged server have rate λ_1. Let steady state probabilities of the tagged server be p_0 and p_1.

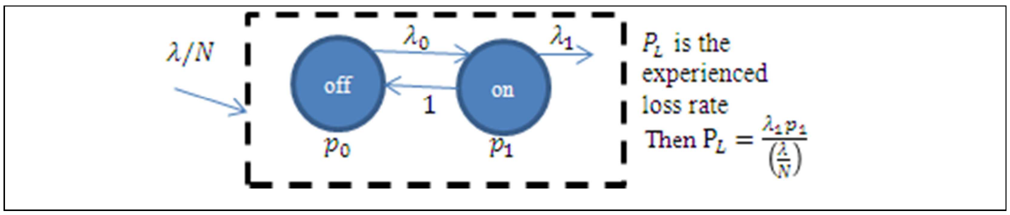


Figure 20. Tagged server

Then $p_0 = \frac{1}{1+\lambda_0}$ and $p_1 = \frac{\lambda_0}{1+\lambda_0}$ (Due to balance equations)

$\lambda_0 p_0 + \lambda_1 p_1 = \frac{\lambda}{N}$ (Preservation of jobs. A single server in M/M/c/c system with N servers and load λ experiences its load as $\frac{\lambda}{N}$).

$\frac{\lambda}{N} P_L = p_1 \lambda_1, \quad P_L = \frac{p_1 \lambda_1}{\frac{\lambda}{N}}$ (Loss rates experienced by both M/M/c/c system and tagged M/M/1/1 server must be equal). Furthermore, it is possible to derive:

$\Delta = \frac{\lambda}{N}(1 - P_L), p_1 = \Delta, p_0 = 1 - \Delta$ and $\lambda_0 = \frac{\Delta}{1-\Delta}, \lambda_1 = \frac{P_L}{1-P_L}$

Somehow, it is needed to relate λ, λ_1 and p_1. If all N of the M/M/c/c servers were independent then this would be true:

$\lambda_1 p_1 = \frac{\lambda}{N}(p_1)^N$

However, it is quite optimistic since it implies that even though there are some busy servers it is possible to find N idle servers. Furthermore, it indicates

[14] Notice the emphasis is only on loss rate

[15] Considering each server in an M/M/c/c system has an On-off behavior and its steady state probabilities depends on other servers in the M/M/c/c system, using an M/M/1/1 server with state dependent arrival rates makes sense.

independence, however, in the case that is being inspected servers are definitely correlated.

If all N of the M/M/c/c show completely same behavior then this would be true:

$\lambda_1 p_1 = \frac{\lambda}{N}(p_1)^1$

This one is also cannot be true since being dependent does not necessarily mean being completely identical (There is no point of same job to get done multiple times by multiple servers).

$\lambda_1 p_1 = \frac{\lambda}{N}(p_1)^K$ definitely holds for some K. Knowing λ, N (since λ_1, p_1 can be computed from these two), K (named as effective number of servers) can be calculated simply by:

$K = 1 + \log_{p1}(\lambda_1/(\lambda/N)\ )$

Using this system it is possible to form an iterative procedure (See Table 6) to find P_L in both ways to confirm that K is a meaningful quantity. $\bar{P}_L$ denotes the blocking probability approximated by our proposed iterative procedure.

1)$P_L = ErlangB(N, \lambda/N)$
2)$p_1 = \Delta = (\lambda/N)(1 - P_L)$
3)$\lambda_0 = \Delta/(1 - \Delta),\ \lambda_1 = P_L/(1 - P_L)$
4)$K = 1 + \log_{p1}(\lambda_1/(\lambda/N)\ )$
5)$p_0 = \frac{1}{1+\lambda_0}\ p_1 = \frac{\lambda_0}{1+\lambda_0}$
6)$\lambda_1 = \frac{\lambda}{N} p_1^{K-1}$
7)$\lambda_0 = \frac{\frac{\lambda}{N} - \lambda_1 p_1}{p_0}$
8) If a convergence criterion is not met continue with step 5, otherwise $\bar{P}_{\mathrm{L}} = \lambda_1 p_1/(\lambda/N)$

Table 6. Suggested iterative procedure

As it can be observed in Figure 21, suggested iterative procedure models M/M/c/c system very accurately by using K effective independent M/M/1/1 servers. For instance, it is possible to interpret $\lambda_1 p_1 = \frac{\lambda}{N}(p_1)^K$ as; for a job to be lost, all K of the M/M/1/1 servers must be busy. K is valuable in reflecting the multiplexing effect which can be observed in an M/M/c/c system.

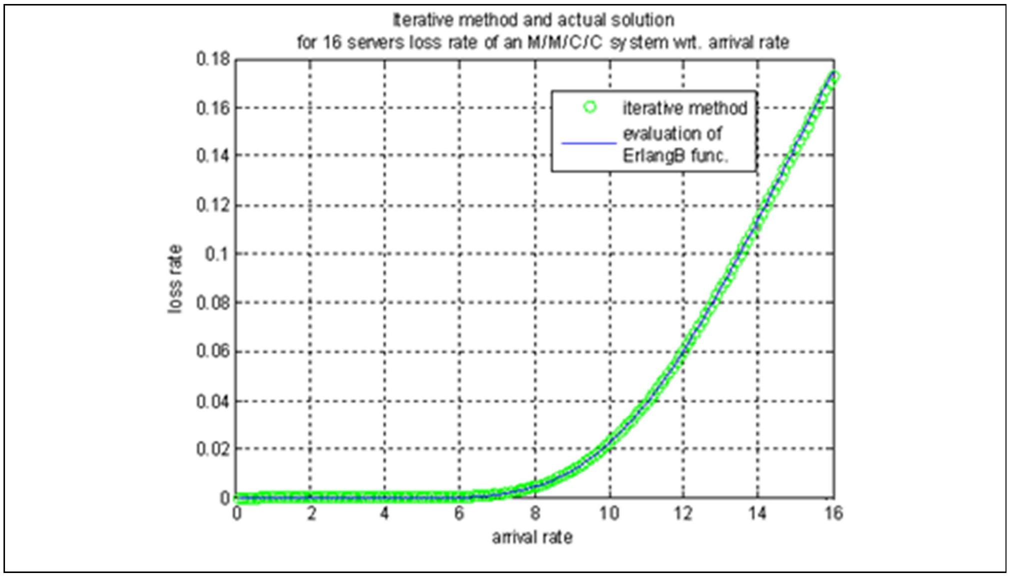


Figure 21. Comparison of ErlangB function and suggested procedure to model loss probability experienced by an M/M/c/c system.

In the next subsection we will try to extend this procedure to the problem stated as this thesis' focus.

3.2 Extension to the Multi-channel Multi-class OBS Problem

In this subsection, we will attempt to solve a multi-channel multi-class OBS problem by using an extended iterative procedure, the idea of effective number of independent servers and solution of single channel multi-class OBS node using bLAUC with regime dependent arrival rates. Before going into further details let us start with focusing on why using bLAUC instead of LAUC makes analysis simpler and why we need regime specific solution introduced in Section 2.3. In the single channel problem, observed rates in regimes are equal to constant offered rate if arrivals of that class are accepted in the specified regime.

However, in a multi-class multi-channel scenario, if LAUC is used then it is expected that arrivals have greater concentration nearby end of a regime, since in equal conditions a channel which has a further horizon is chosen over the one with closer horizon (See Figure 22-b and Figure 23-b). Even though, solution of a fluid queue system with level specific arrival rates is given in [23], there is no known structure for marginal arrival rates experienced by a channel in a multi-channel environment. This is the reason why bLAUC is defined as in Table 5, this way modeling level specific arrival rates observed by a single channel in multi-channel environment with regime specific rates would be possible (See Figure 22-a and Figure 23-a). Total load in the systems is 0.5 and distributed evenly between 2 QoS classes ($\delta_1 = 1$ and $\delta_2 = 6$) in Figure 22 and Figure 23.

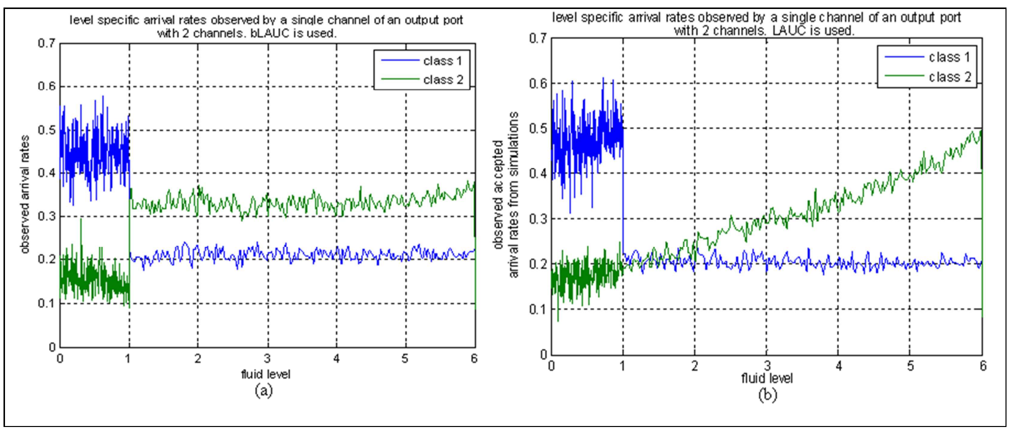


Figure 22 Arrival rates observed by a single channel in a multi-channel environment wrt. fluid level. 2 channels. (a) bLAUC and (b) LAUC

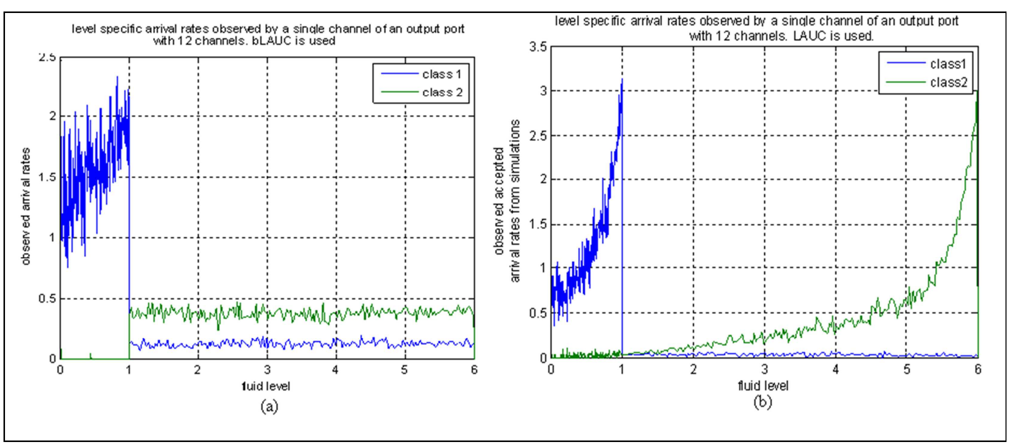


Figure 23. Arrival rates observed by a single channel in a multi-channel environment wrt. fluid level. 12 channels. (a) bLAUC and (b) LAUC

This property of bLAUC makes it easier to analysis and this is the reason why we introduced solution form in Section 2.3. M/M/1/1's in previous subsection will be replaced by a single channel bLAUC system with regime specific arrivals in extension of the iteration given in Table 6.

Now we can focus on extending iterative process to solve our main problem. To achieve this, going over iterative process in Table 6 and characterizing main parts of the procedure may be helpful. There are four main parts described as following: Steps 1-4, which form part 1, are initialization steps and can be identified as determining a valid effective number of independent servers, K. Step 5 can be categorized as solving single channel model and updating steady state probabilities. Step 6-7 can be named as updating state specific arrival rates using steady state probabilities. Final part, step 8, is finding blocking probabilities after system is converged[16].

For first part, we'll use effective number of independent servers computed for the case all offsets of QoS classes are 0 (where M/M/c/c behavior is observed), though better heuristics -possibly depending on offset value- can be proposed. Second part -in which it is needed to solve steady state behavior of a single channel bLAUC problem with current regime specific arrival rates- is given in Section 2.3. Last two parts will be given in this subsection. After giving a more formal definition of the problem and notation, we will again go over all of the mentioned parts.

We will inherit problem definition from introductory part of Section 2.2 with the addition of N denoting number of channels. We can rephrase our main problem as following: We will focus on an N channel output port of an OBS node using OTD method for QoS and bLAUC for scheduling. Classes observed on output port of the node are defined by the class-set $\mathcal{I} = \{1,2,\dots,I\}$. Arrivals of BHPs of class i are experienced as a Poisson process with rate λ_i by the tagged output port. Total load experienced by the system is $\lambda = \sum_{i\in\{1,2,\dots,I\}} \lambda_i$. BHPs of class i arrive to the system δ_i time units before their corresponding bursts. Burst lengths have same distribution and distributed according to an

[16] Steps 5-7 are in a loop.

exponential random variable with mean equal to 1 ($\mu = 1$). If the number identifying a class is larger than another one, then it is assumed the class with larger identifier has greater priority compared to the other. This implies:

$if\ i < j\ then\ \delta_i < \delta_j$.

We will also inherit the regime specific arrival rates notation from Section 2.3. Let $\lambda_i^{(k)}$ denote arrival rate of class i offered to the tagged single channel system (with regime specific arrival rates) when its fluid level is in regime k or at $T^{(k)}$ ($i \in \{1, \dots, I\}$, $k \in \{0,1,2, \dots, I+1\}$, $k = 0$ case corresponds to only being at $T^{(0)}$). Furthermore, we will use $p^{(k)}, k \in \{0,1,2, \dots, I+1\}$ to denote probability of tagged single channel system's being in regime k. In other words, if steady state CDF of the single channel system with regime specific arrival rates is given by $F(y)$ function, then $p^{(0)} = F(\delta_0 = 0)$

$and\ p^{(k)} = F(\delta_k - \delta_{k-1})\ for\ k \geq 1$.

(At this point we can say $\frac{\lambda_i}{N} = \sum_{k=0}^{I+1} p^{(k)} \lambda_i^{(k)}$)

Moreover, blocking probability experienced by class i is denoted as PB_i. Now that notation is given we may go over the previously mentioned parts of the iterative procedure (Seeing Figure 24 may be helpful).

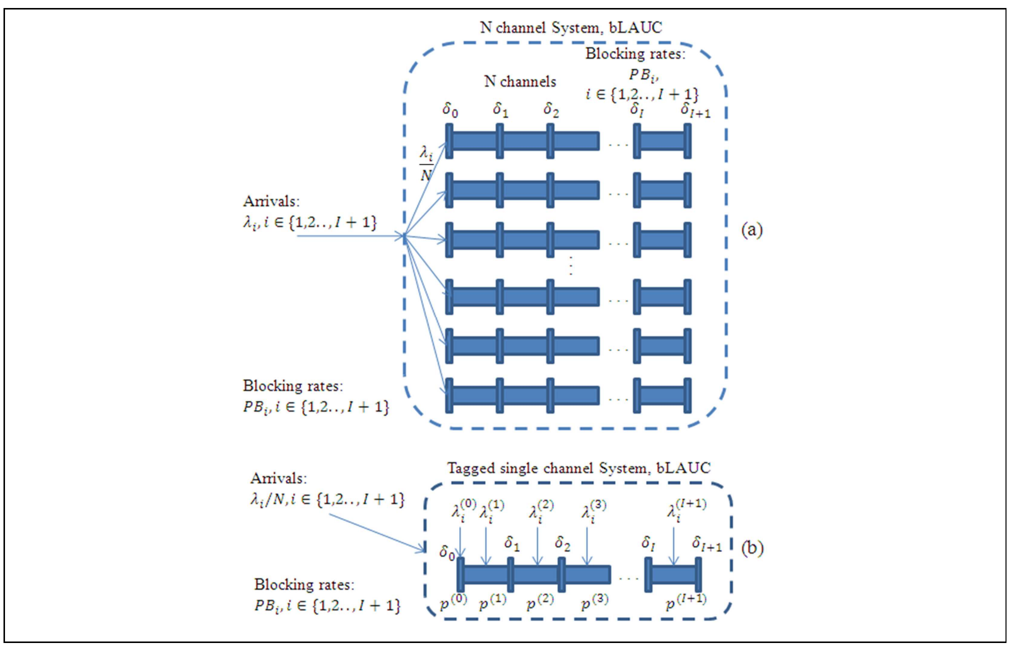


Figure 24. System to be modeled(a) and the model(b) with the notation used.

Now that reviewing notation and problem definition are over, we may focus on the mentioned parts of the iterative process in more detail.

In first part, K is calculated as if given system is an M/M/c/c system with load λ and N servers. $P_L = ErlangB\left(N, \frac{\lambda}{N}\right)$, $\Delta = \left(\frac{\lambda}{N}\right)(1 - P_L)$ and $\bar{\lambda} = \frac{P_L}{1-P_L}$. Therefore, $K = 1 + \log_{\Delta}\left(\frac{\bar{\lambda}}{\left(\frac{\lambda}{N}\right)}\right)$

Second part, in which $p^{(k)}$'s are calculated, is given in Section 2.3.

Third part, which reflects multiplexing effects, state specific arrival rates are updated by using $p^{(k)}$'s and bLAUC algorithm. Now consider amount of class i work comes to the system when a tagged single channel system is in regime k, obviously[17] it is $\lambda_i^{(k)} p^{(k)}$. Same amount of work can be expressed as multiplication of λ_i/N (total work offered to the tagged system) and probability of K effective independent servers being in such a state that an arrival of class i is redirected to a server which is in regime k. Denote this probability with $P_i^{(k)}$. Thus,

$$\frac{\lambda_i}{N} P_i^{(k)} = \lambda_i^{(k)} p^{(k)} \Rightarrow \lambda_i^{(k)} = \frac{\lambda_i}{N}\left(\frac{P_i^{(k)}}{p^{(k)}}\right)$$

Knowing K, $p^{(k)}$'s and using bLAUC algorithm it is possible to calculate the mentioned $P_i^{(k)}$ probabilities:

$$for\ k \in \{0, \dots, i\}, P_i^{(k)} = \left(\sum_{j=0}^{k} p^{(j)} + \sum_{j=i+1}^{I+1} p^{(j)}\right)^K - \left(\sum_{j=0}^{k-1} p^{(j)} + \sum_{j=i+1}^{I+1} p^{(j)}\right)^K \quad [18]$$

And

$$for\ k \in \{i+1, \dots, I+1\}, P_i^{(k)} = \left(\sum_{j=k}^{I+1} p^{(j)}\right)^K - \left(\sum_{j=k+1}^{I+1} p^{(j)}\right)^K \quad [19]$$

[17] Since, arrivals experienced in regime k is $\lambda_i^{(k)}$and probability of tagged single channel system's being in regime k is $p^{(k)}$

[18] Verbally, it can be interpreted as at least one of the K servers (single channel system) must be in regime k, none of the K servers in regime $k+1$ to i (According to bLAUC algorithm, otherwise job redirected to one of the servers which are in the highest numbered regime among others) and remaining servers can be in regimes 0 to $k-1$ or $i+1$ to $I+1$

Perhaps, an example may be helpful in understanding the pattern of $P_i^{(k)}$'s. Consider $I = 2$, which means there are 3 regimes. For a class 2 burst to get rejected all of the K channels must be in regime 3. This implies:

$$P_2^{(3)} = \left(p^{(3)}\right)^K$$

For a class2 burst to get accepted in a channel which is at threshold $T^{(0)} = 0$, there must be at least one server at threshold $T^{(0)}$, none must be in regime 1-2:

$$P_2^{(0)} = \left(p^{(0)} + p^{(3)}\right)^K - \left(p^{(3)}\right)^K$$

For a class2 burst to get accepted in a channel which is in regime 1, there must be at least one server in regime 1, none must be in regime 2:

$$P_2^{(1)} = \left(p^{(0)} + p^{(1)} + p^{(3)}\right)^K - \left(p^{(0)} + p^{(3)}\right)^K$$

For a class2 burst to get accepted in a channel which is in regime 1, there must be at least one server in regime 2

$$P_2^{(2)} = \left(p^{(0)} + p^{(1)} + p^{(2)} + p^{(3)}\right)^K - \left(p^{(0)} + p^{(1)} + p^{(3)}\right)^K$$

And pattern goes so on…

In Table 7 validity of this pattern is verified. Table 7 is obtained via simulations for the case $N = 12, \lambda_1 = \lambda_2 = 3, K = 6.3545, \delta_1 = 1, \delta_2 = 6$. 50M packet arrivals are experienced in discrete event simulations and both regime 1 and regime 2 divided into 100 bins. $p^{(0)} = 0.0236, p^{(1)} = 0.0656, p^{(2)} = 0.6598$, $p^{(3)} = 0.2510$ are obtained via simulations. Comparisons of experienced rates vs. computed rates using mentioned pattern above and $p^{(k)}$'s is given in Table 7.

	BP0	Regime 1	Regime 2	Regime 3
Class 1 sim.	1.1912	1.5937	N/A	N/A
Class 1 comp.	1.0326	1.3348	N/A	N/A
Class 2 sim.	0.0010	0.0030	0.3766	N/A
Class 2 comp.	0.0012	0.0030	0.3787	N/A

Table 7. Comparison of rates from simulation and by using part 3 of the iterations

[19] It is not needed to compute this set of probabilities since these regime specific rates (of the regimes where arrivals cannot be accepted) do not appear in calculations of previous part

Using Table 7 it is possible to conclude pattern in updating regime specific arrival rates is valid[20]. In the last part, after the iterative procedure converges, blocking probability experienced by class i can be found as:

$PB_i = 1 - (\sum_{k=0}^{i} p^{(k)} \lambda_i^{(k)}) / (\lambda_i / N)$. Note that an arrival of class i cannot be accepted if the system is in one of the regimes $i+1$ to $I+1$. Of the all traffic of class i directed to a tagged single channel system (λ_i/N), a fraction sees server is unable to accommodate a class i burst. This fraction indicates loss probability for that class, and PB_i can be justified like this.

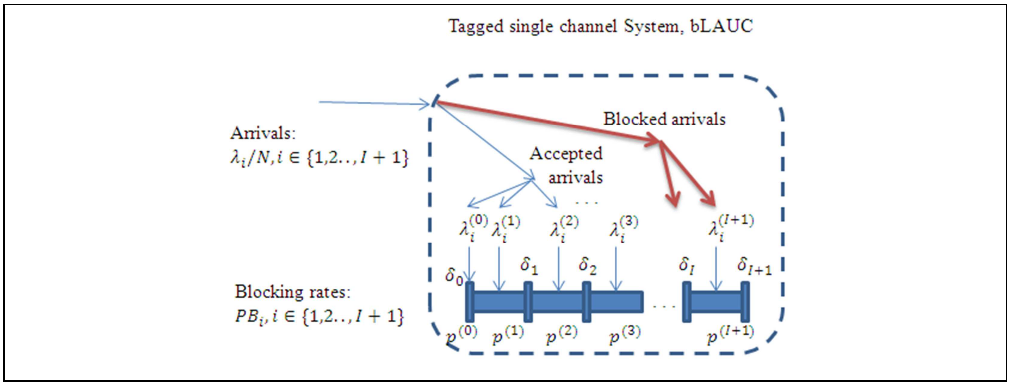


Figure 25. Arrivals that are blocked in the tagged system

Complete iterative procedure can be seen in Table 8.

1) Start with $\lambda_i^{(k)} = \frac{\lambda_i}{N}, i \in \{1, .., I\}\ k \in \{0, \ldots, I+1\}$

2) Find K using described method

3) Solve single channel fluid queue with regime specific arrival rates and obtain, $p^{(k)}, k \in \{0, \ldots, I+1\}$

4) Update regime specific arrival rates $\lambda_i^{(k)}$'s (based on bLAUC's algorithm) by using $p^{(k)}$'s and $P_i^{(k)}$'s $k \in \{0, \ldots, I+1\}$

If system converges then keep on going with step 5, otherwise return to step 3.

5) Compute loss rates by using $PB_i = 1 - (\sum_{k=0}^{i} p^{(k)} \lambda_i^{(k)}) / (\lambda_i / N)$.

Table 8. Complete iterative procedure to approximate per-class loss rates for our problem

[20] Differences can be explained as using K which is insensitive to the offset scheme and even though our way of designing bLAUC is to have almost constant regime specific arrival rates, those are still not piecewise constant functions. See Figure 23

In this chapter, we proposed an iterative method, which utilizes a simpler iterative method to solve an Erlang loss system, to approximate blocking probabilities of different QoS classes in a setting where bLAUC is used. Given method is quite important since it gives very satisfying results with really low complexity compared to discrete event simulators. Also due to the method used to reflect multiplexing effects and focus on only one tagged system, solution method does not suffer severely from dimensionality problems due to increasing number of channels and classes unlike methods depending on discretization of several dimensions.

Chapter 4

Numerical Results

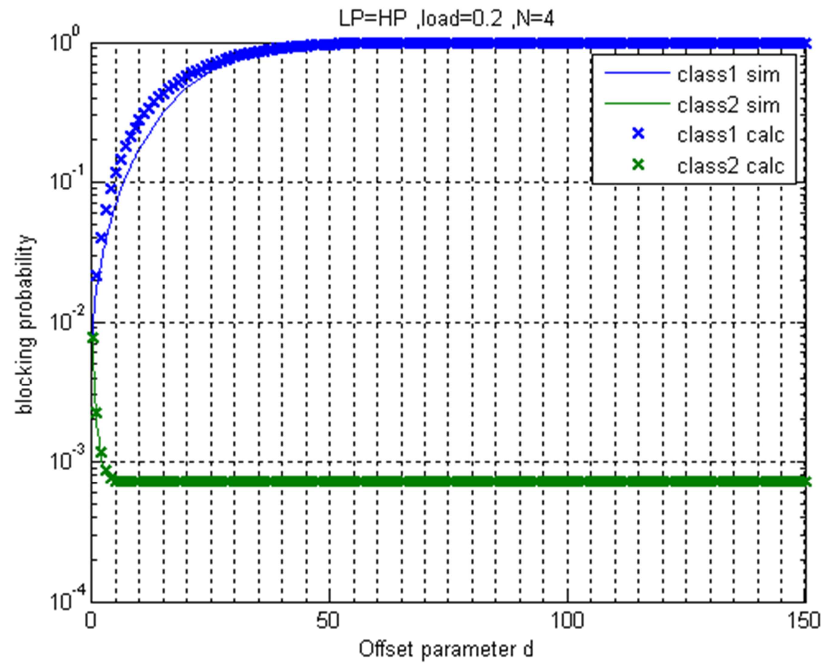
In this chapter, results -mostly in the form of plots- will be presented to compare the approximations obtained by our iterative procedure and discrete event simulations as a way to validate the suggested approach. Then using our iterative procedure, we will solve an optimization problem which would take a very long time period if discrete event simulations were to be used.

4.1 Comparisons with Discrete Event Simulations under Various Settings

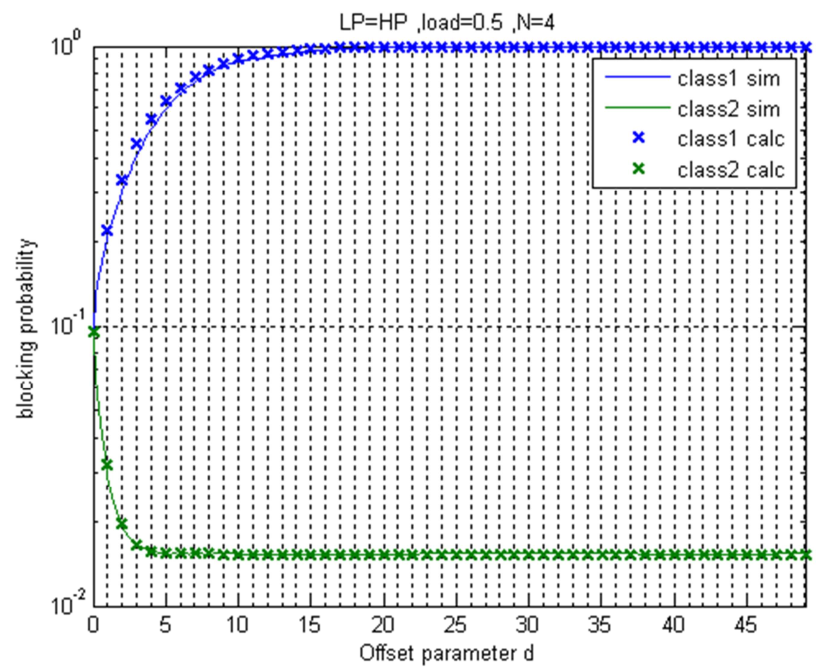
We will provide various scenarios with two, three and ten QoS classes. Within this chapter, solid lines denote simulations and dashed lines or crosses denote results of iterations. Three load scenarios are investigated for a 2-class scenario (0.2, 0.5, and 0.8), two load scenarios (0.5 and 0.8) for a 3-class scenario and only a single load scenario (0.5) for a 10 classes setting. For the 2-class scenario, also different mixtures of HP and LP arrivals are used; if they are not equal then one has an arrival rate which is 4 times of the other one. In other cases, load is always distributed evenly between classes. Systems with 4 or 16 channels are investigated for the 2-class system; 4 and 8 channels are used for the 3-class scenario and only an 8 channel system is used for the 10-class scenario. Offset schemes are parameterized by a single parameter d. In two class scenarios, $\delta_1 = 0$ and $\delta_2 = d$. In the 3-class case, $(\delta_1, \delta_2, \delta_3) = (0, d, 2d)$ and for the 10-class case $(\delta_1, \delta_2, \dots, \delta_{10}) = (0, d, \dots, 9d)$. In plots using logarithmic scaling for its y axis, sometimes, solid lines become missing, which is due to having no realizations in simulations and obtaining a 0 probability value[21]. Comments are left to the Subsection 4.1.4.

[21] Simulations took more than 4 weeks; however, it is still possible to observe "noise" on them. For each point of the simulations, 5 million arriving packets are used and simulations are repeated 7 times. However, confidence interval of the results of simulation are so large that we observe the "noise". Other than that, to obtain very accurate results with narrow confidence intervals in cases where probabilities are really small (See Figure 26-d3), we would need more than couple of years.

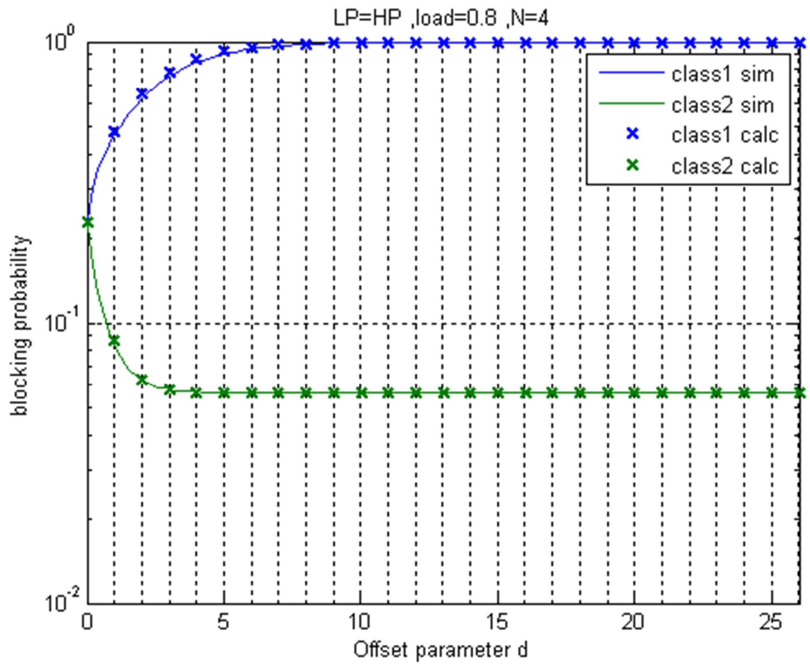
4.1.1 Two Classes



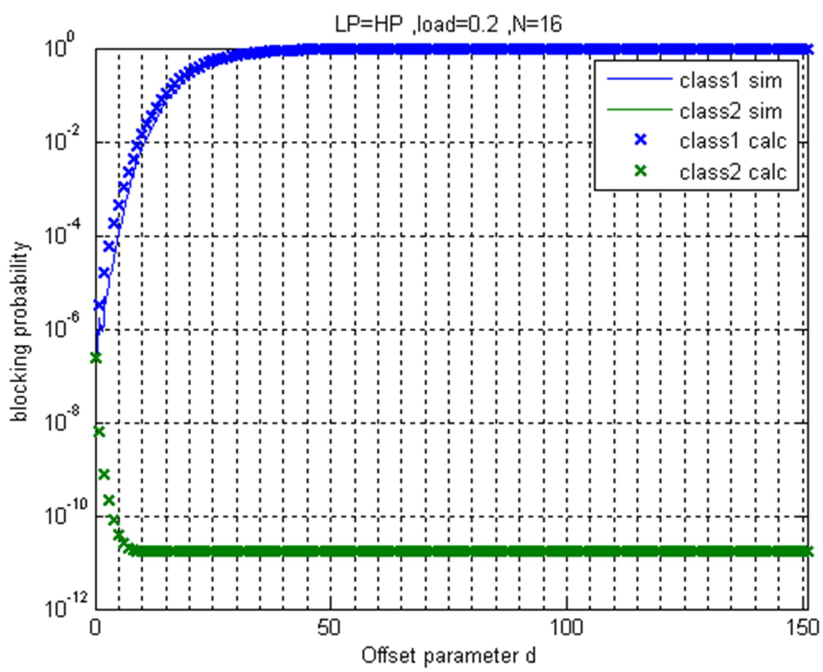
(a1)



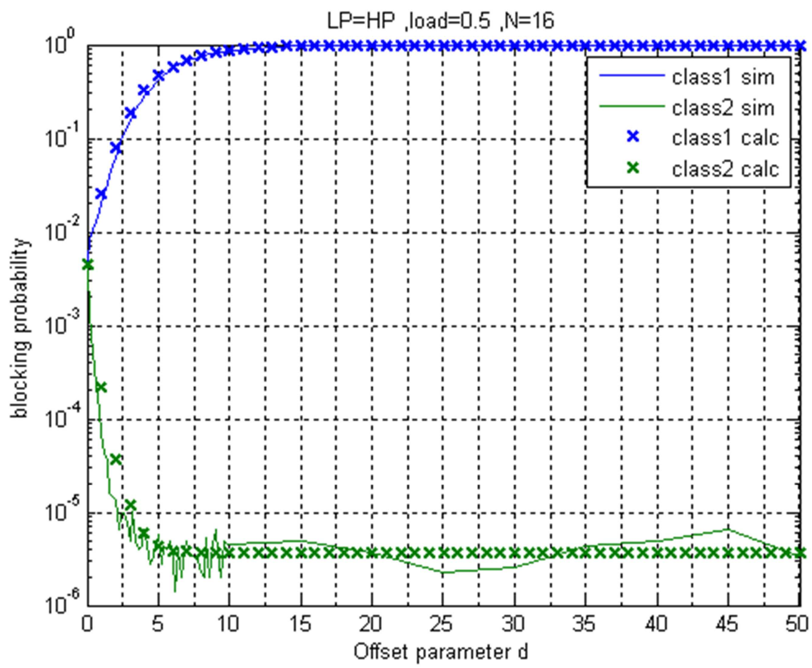
(b1)


LP=HP ,load=0.8 ,N=4
blocking probability
Offset parameter d
class1 sim
class2 sim
class1 calc
class2 calc

(c1)

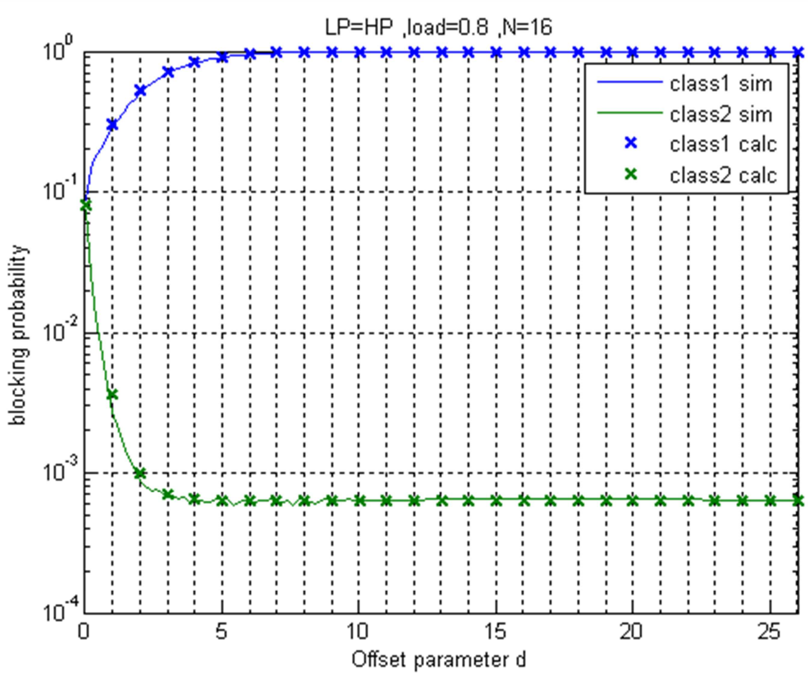

LP=HP ,load=0.2 ,N=16
blocking probability
Offset parameter d
class1 sim
class2 sim
class1 calc
class2 calc

(d1)



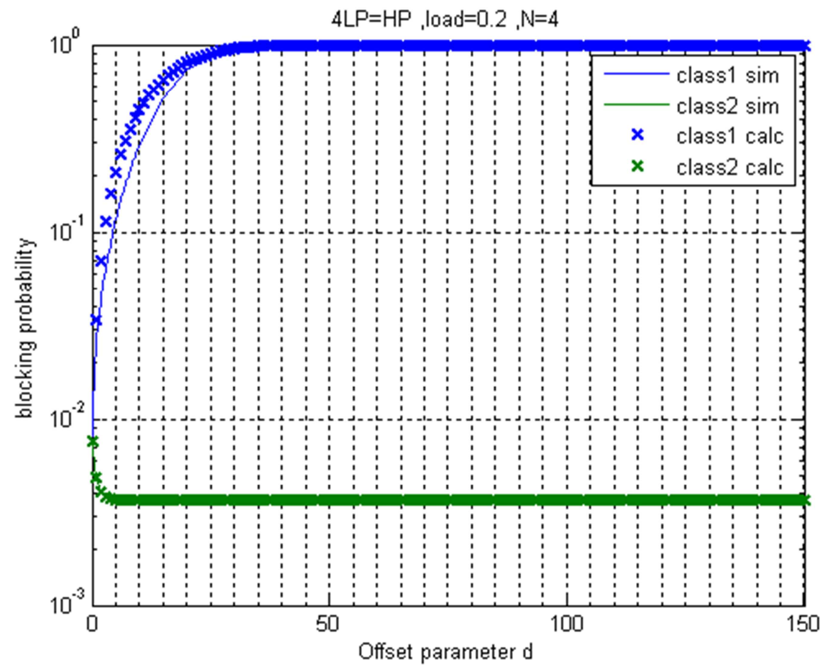
LP=HP ,load=0.5 ,N=16
blocking probability
Offset parameter d
class1 sim
class2 sim
class1 calc
class2 calc

(e1)

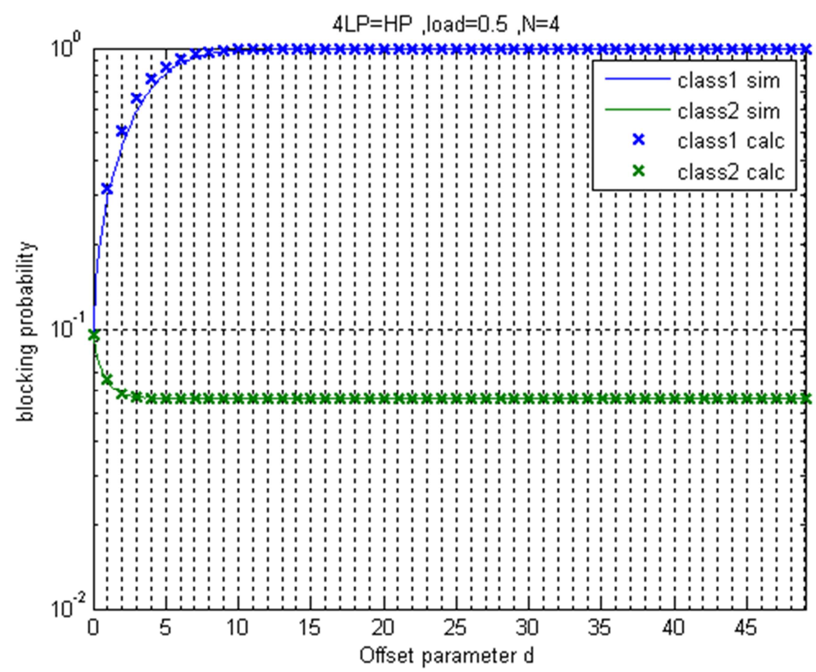


LP=HP ,load=0.8 ,N=16
blocking probability
Offset parameter d
class1 sim
class2 sim
class1 calc
class2 calc

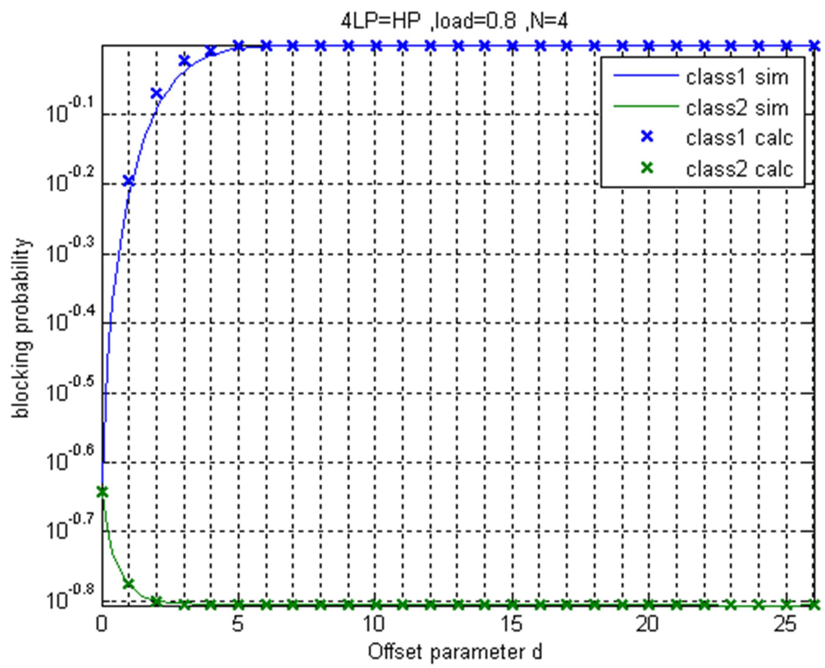
(f1)


4LP=HP ,load=0.2 ,N=4
class1 sim
class2 sim
class1 calc
class2 calc
blocking probability
Offset parameter d

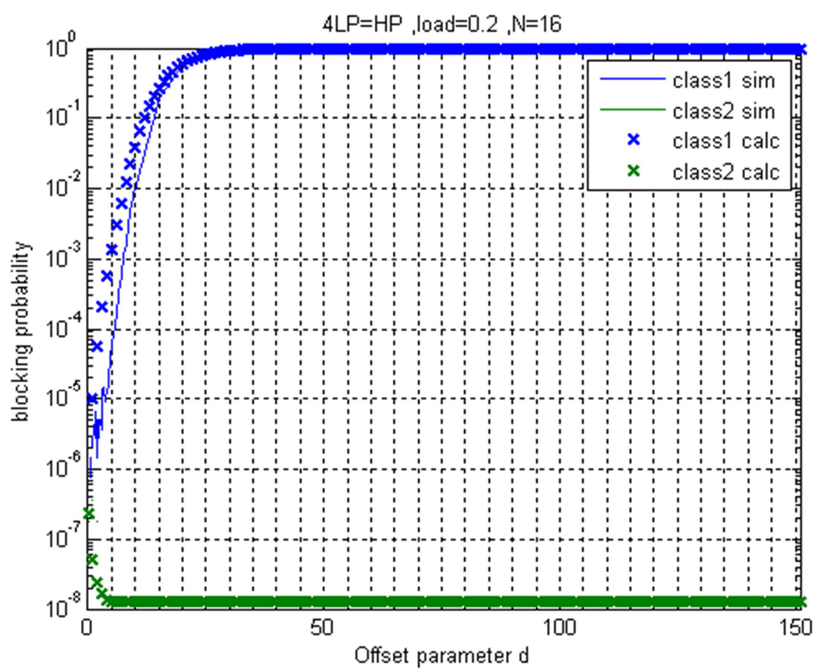
(a2)


4LP=HP ,load=0.5 ,N=4
class1 sim
class2 sim
class1 calc
class2 calc
blocking probability
Offset parameter d

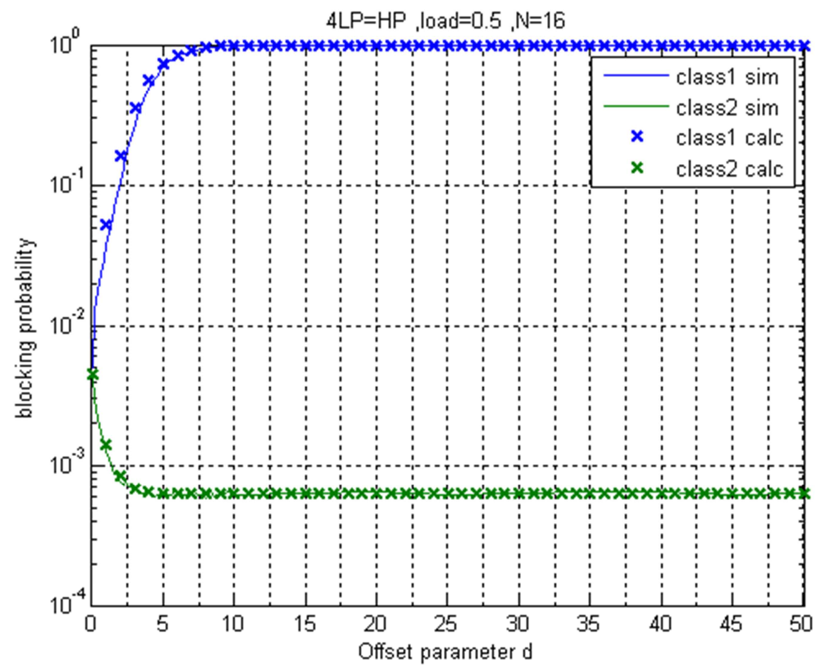
(b2)


4LP=HP ,load=0.8 ,N=4
class1 sim
class2 sim
class1 calc
class2 calc
$10^{-0.1}$
$10^{-0.2}$
$10^{-0.3}$
$10^{-0.4}$
$10^{-0.5}$
$10^{-0.6}$
$10^{-0.7}$
$10^{-0.8}$
blocking probability
0
5
10
15
20
25
Offset parameter d

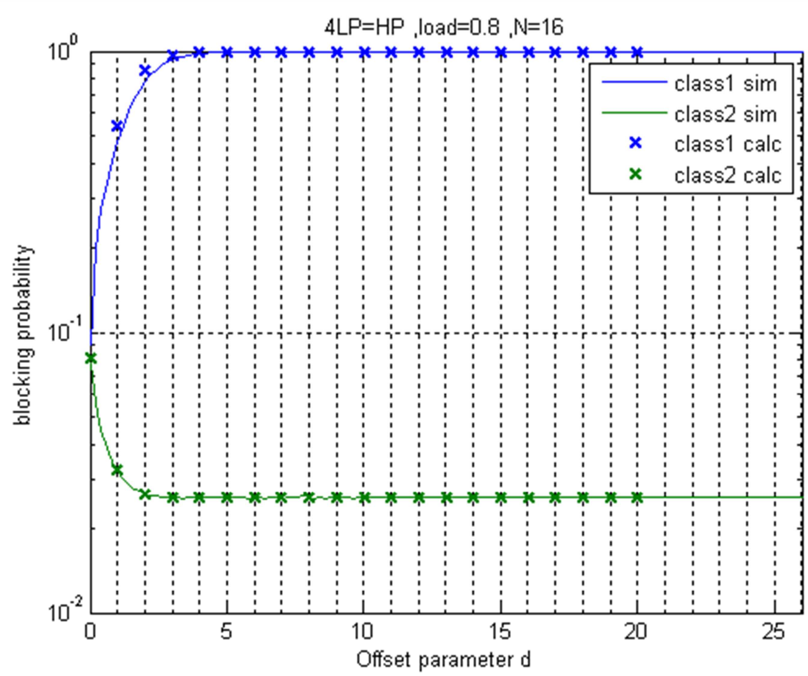
(c2)


4LP=HP ,load=0.2 ,N=16
class1 sim
class2 sim
class1 calc
class2 calc
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
blocking probability
0
50
100
150
Offset parameter d

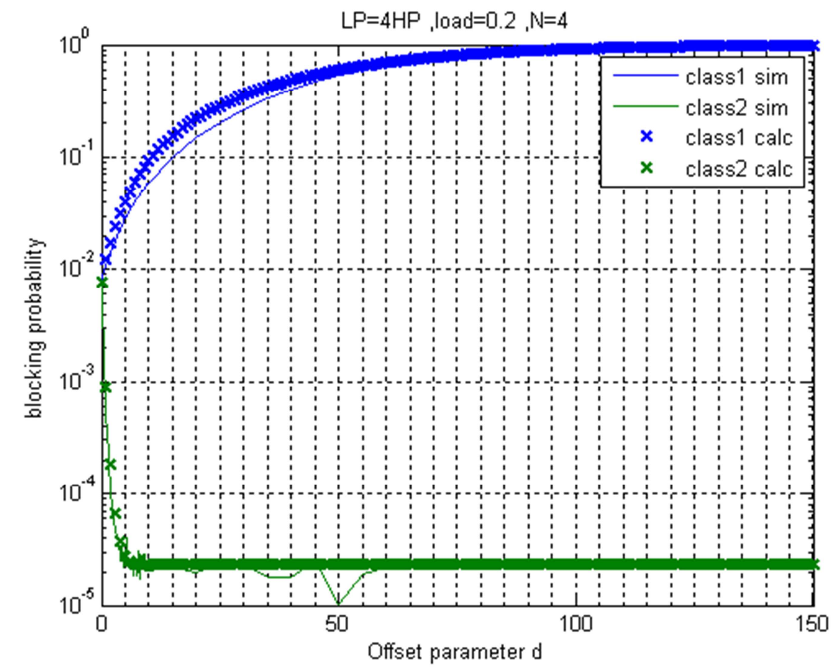
(d2)


4LP=HP ,load=0.5 ,N=16
class1 sim
class2 sim
class1 calc
class2 calc
blocking probability
Offset parameter d

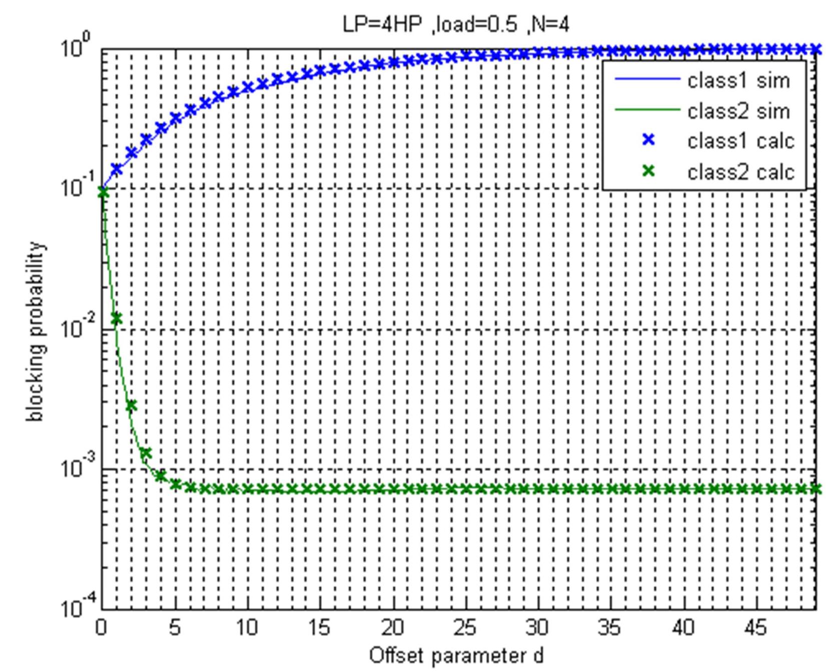
(e2)


4LP=HP ,load=0.8 ,N=16
class1 sim
class2 sim
class1 calc
class2 calc
blocking probability
Offset parameter d

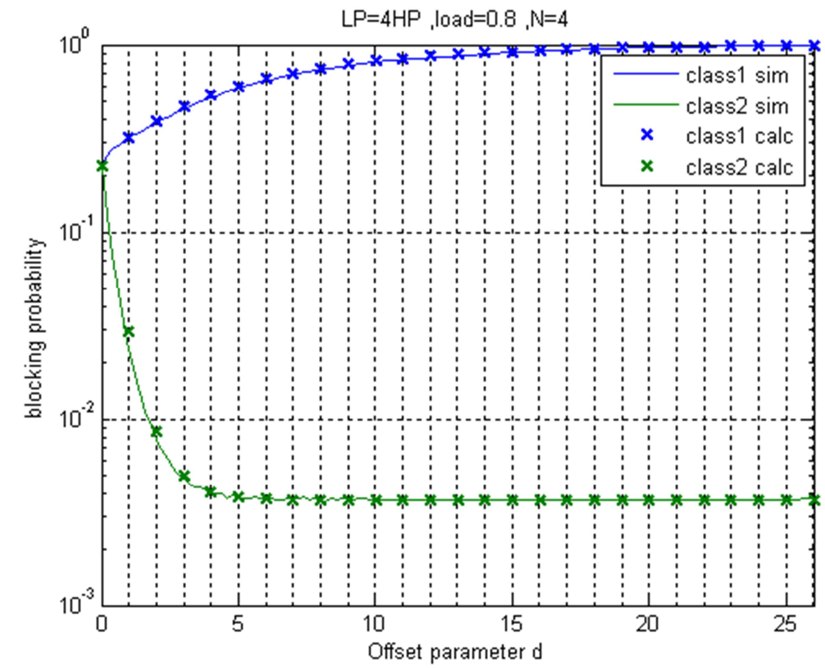
(f2)


LP=4HP ,load=0.2 ,N=4
blocking probability
Offset parameter d
class1 sim
class2 sim
class1 calc
class2 calc

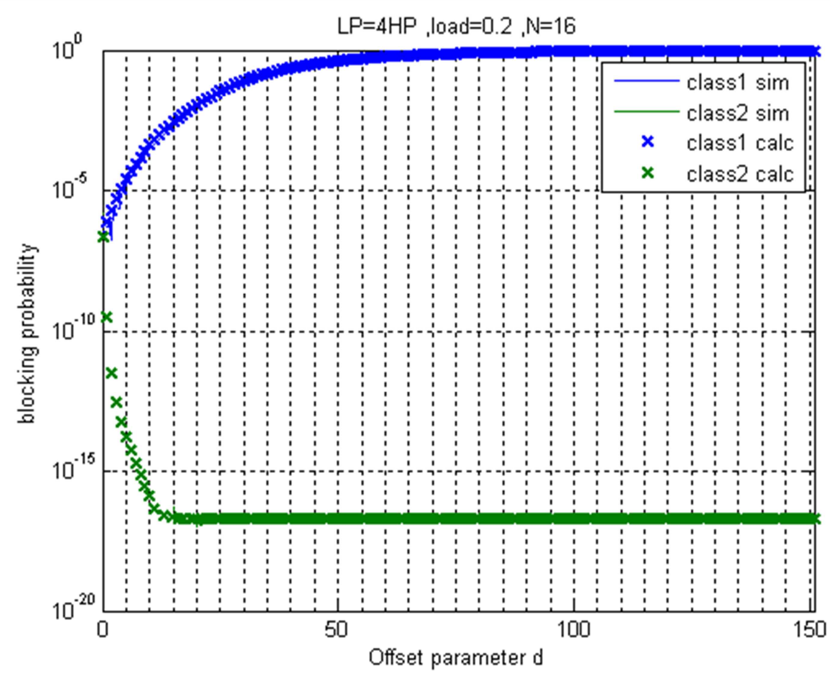
(a3)


LP=4HP ,load=0.5 ,N=4
blocking probability
Offset parameter d
class1 sim
class2 sim
class1 calc
class2 calc

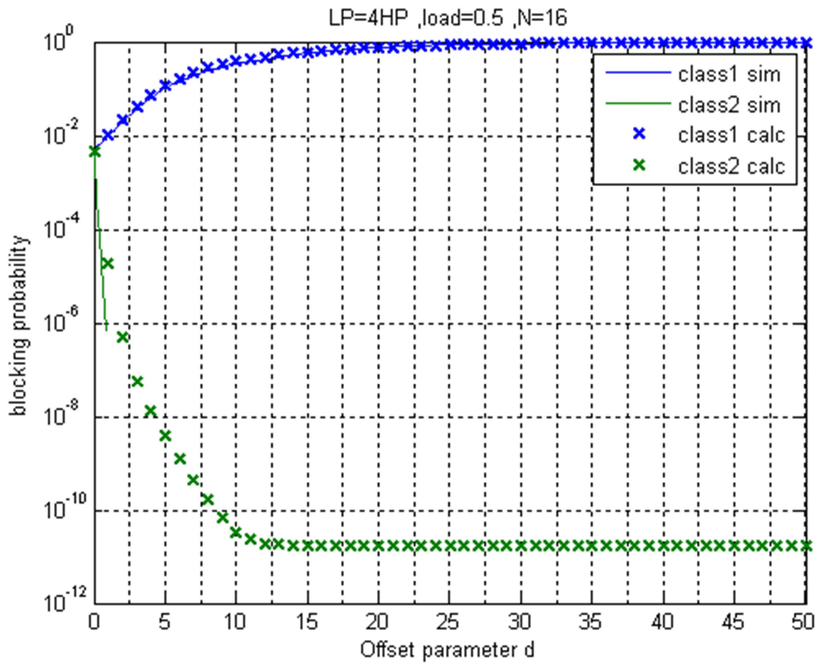
(b3)


LP=4HP ,load=0.8 ,N=4
class1 sim
class2 sim
class1 calc
class2 calc
blocking probability
Offset parameter d

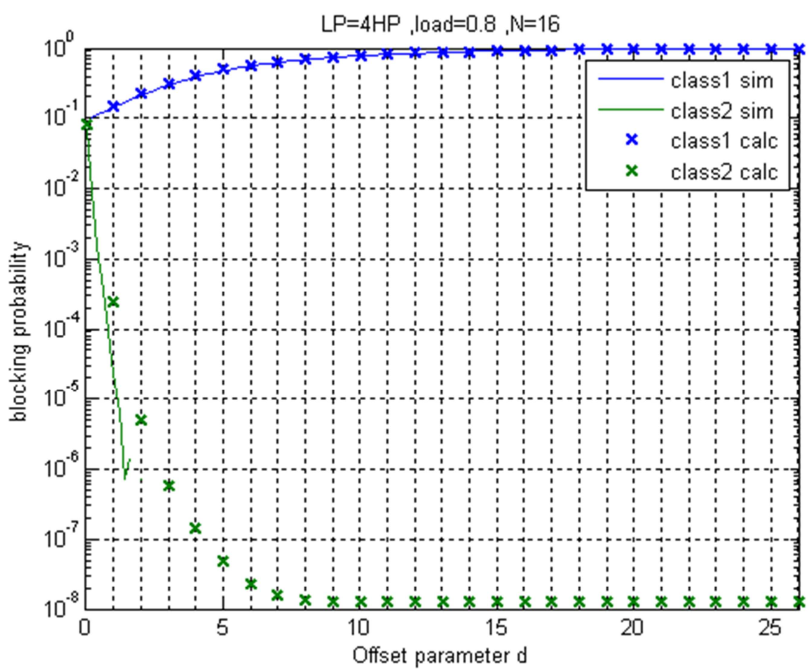
(c3)


LP=4HP ,load=0.2 ,N=16
class1 sim
class2 sim
class1 calc
class2 calc
blocking probability
Offset parameter d

(d3)



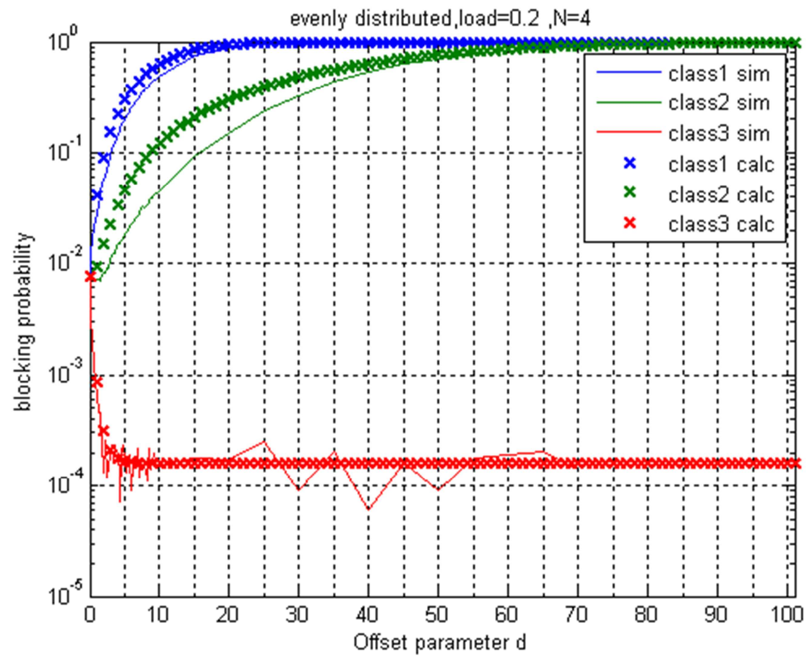
(e3)



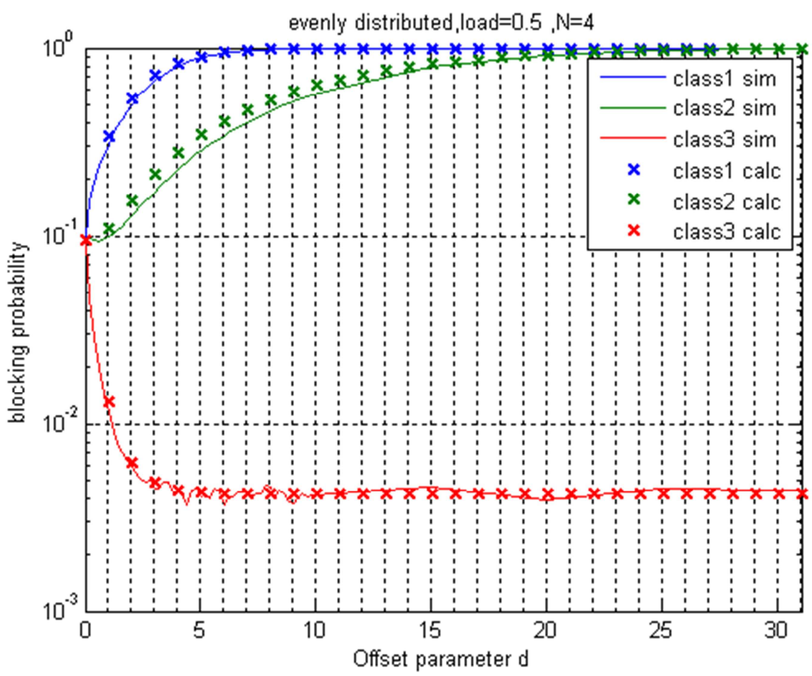
(f3)

Figure 26. Two class scenarios for different traffic mixtures. "a,b,c"s are for 4 channels and "d,e,f"s are for 16 channels. Mixture information is given in the titles of the plots

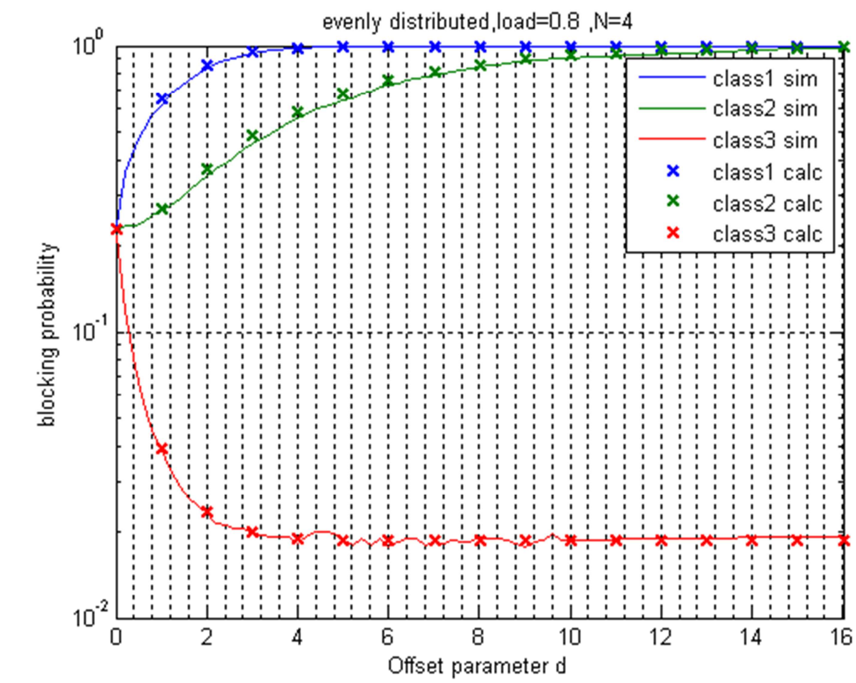
4.1.3 Three Classes



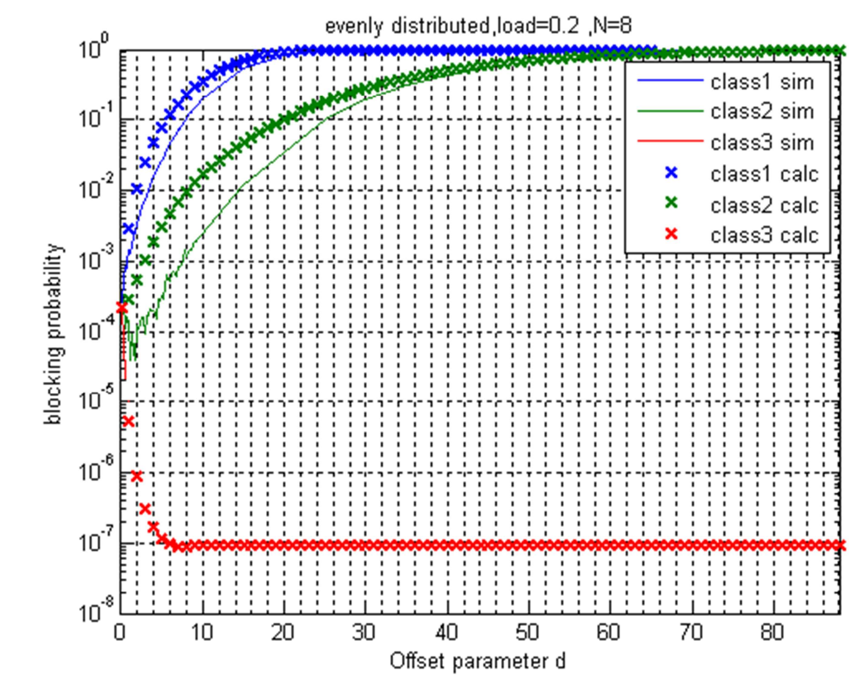
(a)



(b)


evenly distributed,load=0.8 ,N=4
blocking probability
Offset parameter d
class1 sim
class2 sim
class3 sim
class1 calc
class2 calc
class3 calc

(c)


evenly distributed,load=0.2 ,N=8
blocking probability
Offset parameter d
class1 sim
class2 sim
class3 sim
class1 calc
class2 calc
class3 calc

(d)

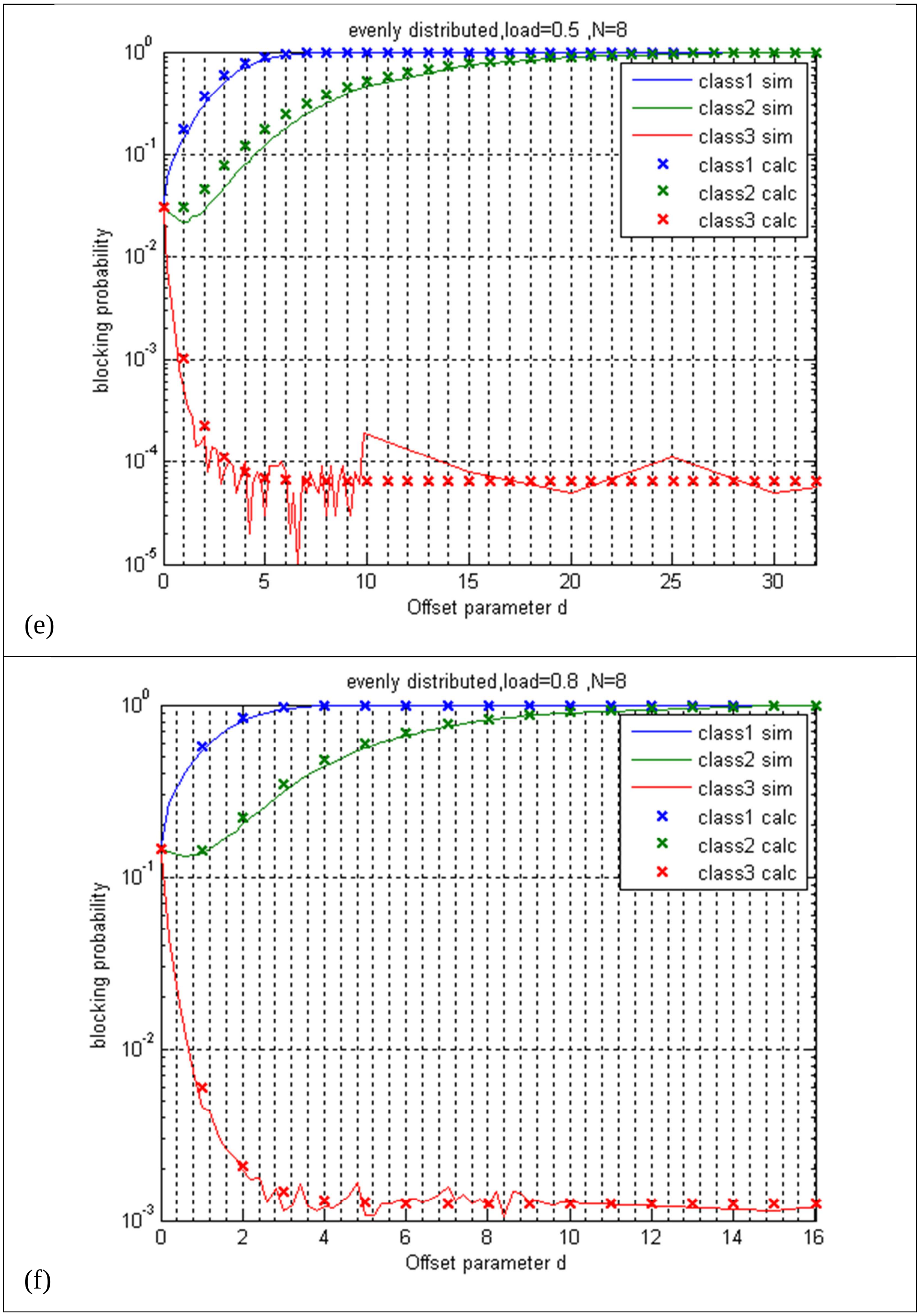


Figure 27. 3 class scenarios with logarithmic scaling on Y axis: (a-c) 4 channels (d-f) 8 channels. Approximation error does not increase with increasing number of channels.

4.1.3 Ten Classes

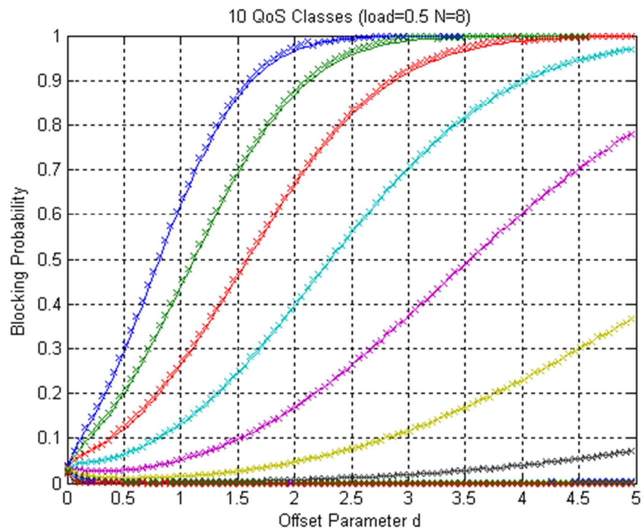


Figure 28 10 QoS Classes. Suggested method works for arbitrary number of classes

4.1.4 Overview

Considering the given plots (Figure 26Figure 28), it is possible to say that our solution method captures the general properties of the system. Furthermore, our iterative method presents very accurate approximations in moderate to high load scenarios. Additionally, accuracy of the proposed method does not decrease as the number of channels and classes increase when load is fixed. In fact, our method provides exact solutions at the limiting cases, in which the problem can be modeled by an M/M/c/c. Knowing this, it is possible to say that the more **accepted traffic** to the system is dominated by a "single class", the more accurate results our method provides. As it can be easily observed, our method captures system behavior completely for no offset case, after that point our approximation introduces some error and that error diminishes as per-class loss rates reach steady-state with respect to delay parameter. Notice that our method finds the point where loss probabilities are saturated very accurately. Finally, it is possible to use our method for various numbers of classes (see Figure 28) and it still offers acceptable approximations.

4.2 A Case Study: Solving an Optimization Problem

In this subsection, we will attempt to solve a throughput optimization problem with certain constraints using our method. Constraints can be summarized as: $N = 4$, $I = 2$, $\lambda_1 = 4\alpha$, $\lambda_2 = \alpha$, $\alpha \in [0,1]$, $\delta_2 - \delta_1 \in [0,20]$, $PB_1 < 10^{-1}$ and $PB_2 < 10^{-4}$. We are to maximize total throughput of the system. In solving this problem, we deployed a binary search method on a grid until a predetermined neighborhood condition is satisfied (Details are omitted since searching in a space is a very deep problem in itself [26].). Solution of this problem takes less than 10 minutes of computing where simulations may require a time period between a week and two weeks[22]. Plots giving results can be seen in Figure 29 and Figure 30

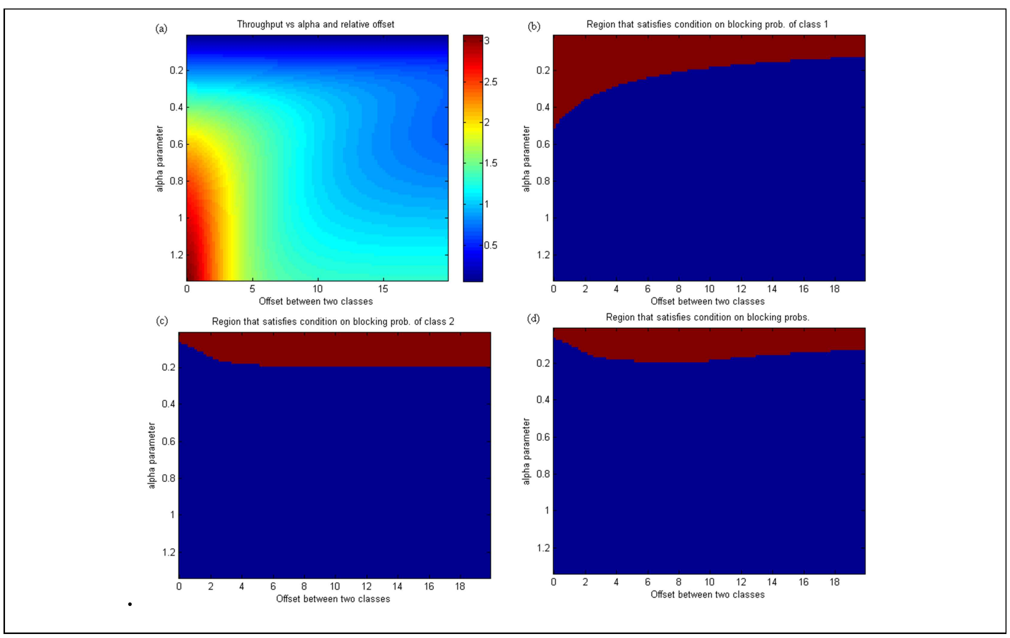


Figure 29. Initial search grid results.(a) shows throughput wrt. delay and alpha parameters. (b), (c): delay and alpha parameters satisfying loss rate criteria LP and HP. (d) for both classes

[22] A crude estimation based on previous simulations

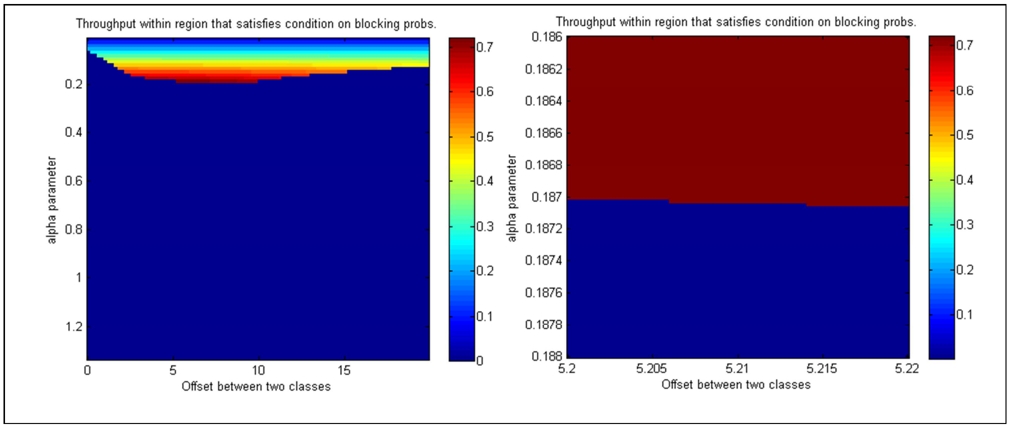


Figure 30. Throughput in region satisfying criteria

Results of our search using iterations are summarized in Table 9. Results are satisfying enough. Considering that our procedure is slightly pessimistic compared to simulations it can be used for determining some guidelines for designing a system using bLAUC.

	Iterative Procedure	Simulations
Alpha	0.1871	0.1871
Offset Difference	5.2141	5.2141
PB_1	0.0448	0.0291
PB_2 [23]	$9.99\text{x}10^{-5}$	0.0001
Throughput	0.7230	0.7318

Table 9. Comparison of iterative procedure and corresponding simulation results in the case study.

[23] PB_2 of the iterative procedure is within numerical sensitivity step of the programming language being used.

Chapter 5

Conclusions

This thesis provides an iterative method to accurately approximate blocking probabilities of QoS classes based upon bLAUC algorithm in an output port with multiple channels, this is important due to the fact that, for multiple channel case, there is no method or solution that can provide per class loss probabilities.

Furthermore, our method achieves this with a very low complexity. Although, the provided framework does not include any details regarding methods to obtain numerical solutions, considering the number of classes the complexity of our iterative procedure is linear (exploiting almost block tridiagonal structure formed by boundary conditions) and constant with number of classes and offset scheme. Even though bLAUC is a relaxation on LAUC and applicability of bLAUC may be relatively limited, our approach can be viewed as a first step to model LAUC. Another contribution of this work is providing an iterative procedure which may be used for other multi-server system performance modeling problems in which servers are dependent. Though, methods that estimate the effective number of independent servers (denoted by K in this thesis) and associated rate update equations are needed.

Solving multiple channel bLAUC system with FDLs or multiple channel LAUC system (with or without FDLs) are still open problems which will be in the scope of our future work.

BIBLIOGRAPHY

[1] Biswanath Mukherjee, *Optical WDM Networks*. New York: Springer, 2006.

[2] B. Ramamurthy, H. Feng, J.P. Heritage D. Data, and B. Mukherjee, "Transparent vs. opaque vs, translucent wavelength routed optical networks," in *OFC '99*, San Diego, CA, 1999, pp. 59-61.

[3] A. Willner, M.C. Cardakli, O.H. Adamczyk, Y. Song, and D. Gurkan, "Key building blocks for all-optical networks," *IEICE Transactions in Communications vol. E83-B*, October 2000.

[4] R. Ramaswami and K.N. Sivarajan, "Routing and wavelength assignment in all-optical networks," *IEEE/ACM Trans. Networking, vol.3,* p. 1996, 858-867.

[5] P. Gambini et al., "Transparent optical packet switching: network architecture and demonstrators in the KEOPS project," *IEEE J. Select. Areas Commun. vol. 16* , pp. 1245–1259, 1998.

[6] C. Qiao and M. Yoo, "Optical burst switching (OBS) - a new paradigm for an optical Internet," *Jour. High Speed Networks, vol. 8, no. 1* , pp. 69-84, 1999.

[7] K. Sivalingam and S. Subramaniam, *Optical packet switching in Emerging Optical Network Technologies: Architectures, Protocols, and Performance*. Norwell, MA: Springer, 2004, pp. 111-127.

[8] M. Klinkowski, "Offset Time-Emulated Architecture for Optical Burst Switching - Modelling and Performance Evaluation: Ph D. thesis," 2007.

[9] Y. Chen, C. Qiao, and X. Yu, "Optical Burst Switching: A new area in optical networking research," *IEEE Network, vol. 18, no. 5*, pp. 16-23, 2004.

[10] J.Y. Wei and R.I. McFarland, "Just-in-time signaling for WDM optical burst switching networks," *J. Lightwave Techn., vol. 18, no. 12,* pp. 2019-2037, 2000.

[11] J.S. Turner, "Terabit burst switching," *J. High Speed Networks, vol. 8, no. 1*, p. 1999, 3-16.

[12] J. Teng and G.N. Rouskas, "A Detailed Analysis and Performance Comparison of Wavelength Reservation Schemes for Optical Burst Switched Networks ," *Photonic Network Communications, vol. 9, no. 3*, p. 2005, 311-315.

[13] Y. Xoing, M. Vandenhoute, and C. Cankaya, "Control architecture in

optical burst-switched wdm networks," *IEEE J. Sel. Areas Commun., vol. 18*, pp. 1838–1851, 2000.

[14] M. Ljolje, R. Inkret, and B. Mikac, "A comparative analysis of data scheduling algorithms in optical burst switching networks," *Proceedings of Conference on Optical Network Design and Modeling*, pp. 493-500, 2005.

[15] N. Akar et al., "A survey of quality of service differentiation mechanisms for optical burst switching networks," *Optical Switching and Networking, Vol. 7, Issue 1*, pp. 1-11, 2010.

[16] K.C. Chua, M. Gurusamy, Y. Liu, and M.H. Phung, *Quality of Service in Optical Burst Switched Networks*.: Springer, 2007.

[17] M. Klinkowski, D. Careglio, S. Spadaro, and J. Sole-Pareta, "Impact of burst length differentiation on QoS performance in OBS networks," *in Proceedings of IEEE ICTON*, pp. 91-94, 2005.

[18] M. Yoo, C. Qiao, and S. Dixit, "Optical burst switching for service differentiation in the next generation optical internet," *IEEE Commun. Mag., vol. 39, no. 2*, pp. 98-104, 2001.

[19] A.M. Kaheel, H. Alnuweiri, and F. Gebali, "A New Analytical Model for Computing Blocking Probability in Optical Burst Switching Networks," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 24, NO. 12*, pp. 120-128, 2006.

[20] N. Barakat and E.H. Sargent, "An accurate model for evaluating blocking probabilities in multi-class obs systems," *IEEE Communications Letters, vol. 8*, pp. 119-121, 2004.

[21] H.E. Kankaya, "Performance Evaluation of Telecomunication Networks Based on Multi-regime Fluid Queues: PhD. Dissertation," 2009.

[22] H.E. Kankaya and N. Akar, "Exact analysis of offset-based differentiation in single-channel multi-class obs," *IEEE Comm. Letters, vol. 13,* , pp. 148-150, 2009.

[23] W. Scheinhardt, "Markov Modulated and Feedback Fluid Queues. PhD thesis," 1998.

[24] M. Mandjes, D. Mitra, and W. Scheinhardt, "Models of network access using feedback fluid queues," *Queueing Syst. Theory Appl., vol. 44, no. 4* , pp. 2989–3002, 2003.

[25] D.P. Kroese and W.R Scheinhardt, "Joint Distributions for Interacting Fluid Queues," *Queueing Systems, 37*, pp. 99-139, 2001.

[26] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization.," *Journal of the Operational Research Society*, vol. 10, no. 57, pp. 1143-1160, 2006.