A discrete-time queueing network model of a hub-based OBS architecture

Xenia Mountrouidou

IBM Software Group,
Building 503, 4205 S. Miami Blvd.
Durham, NC 27703, USA

Harry Perros

Computer Science Department,
North Carolina State University,
Raleigh, NC 27695, USA

Abstract

We propose and analyze a novel discrete-time queueing network model of a zero loss hub-based Optical Burst Switched (OBS) architecture, consisting of multiple input edge nodes and one destination edge node. The arrival process of bursts is slotted with bulk arrivals as generated by a Time and Burst-Length based burst aggregation algorithm. The queueing network is analyzed by decomposition. We obtain the average end-to-end delay of a burst in the queueing network as well as queueing delays at individual nodes. Our model provides a tight upper bound as shown by comparing the analytical data to simulation results.

1 Introduction

Performance studies point to the fact that in an Optical Burst Switched (OBS) network, the link utilization has to be kept very low in order for the burst loss probability to be within an acceptable level. Various congestion control schemes have been proposed, such as the use of converters, fiber delay lines, and deflection routing. However, these schemes do not alleviate this problem. It is our position that in order for OBS to become commercially viable, new

Email addresses: {xeniamount}@us.ibm.com (Xenia Mountrouidou), {hp}@ncsu.edu (Harry Perros).

schemes have to be devised that will either guarantee zero burst loss or a very low burst loss at high utilization. In this paper we model an architecture which provides zero burst loss for a hub-based OBS topology. We evaluate the performance of this scheme using a discrete time queueing network with blocking and bulk arrivals. The queueing network under study is versatile and can be used to analyze other systems such as Websphere [5] application servers that communicate with databases.

OBS networks have been extensively modeled using a single Erlang loss queue. In [3], [4] and [14] a bufferless OBS network with Poisson burst arrivals and multiple input and output wavelengths has been analyzed. The OBS network is modeled as a multiple server system and the Erlang B formula is applied to calculate the burst loss probability. Assuming that burst arrivals follow a Poisson distribution is unrealistic as shown in [9] since the burst arrival process depends on the aggregation algorithm that is used. More specifically if the Time based aggregation is used then the burst arrival process is slotted with a burst size that has an Erlang distribution, since the burst is a convolution of exponential packet sizes. If the Burst-Length based aggregation is used, then the burst inter-arrival time follows an Erlang distribution and the burst sizes are fixed to a predefined maximum B_{max}. Finally, if the Time and Burst-Length based aggregation is used, then the burst arrival process is a slotted bulk arrival process with fixed burst sizes except for the last burst in the bulk that has a uniform size distribution within the interval $[B_{min}, B_{max})$, where B_{min} and B_{max} are pre-specified minimum and maximum burst sizes.

The performance of OBS has been studied by several authors. In [15] the authors developed a number of queueing network models that provide insights to the effect of various system parameters on the performance of a core OBS node that interacts with a number of edge nodes according to the Just In Time (JIT) signaling protocol. In this architecture, that provides zero burst loss, it is assumed that the core node allocates resources within its switch fabric for a burst at the moment it decides to accept the setup request of the edge node. The burst arrival process they use is a three-state Markov process, that may be in one of the states: *short burst*, *long burst*, or *idle*. Since a customer may request either a short or a long service (burst), the service time distribution is a two-stage hyper-exponential distribution. The authors analyzed a single-class network of the core OBS node without converters and then they extended this to the case with converters. Finally, they further extended these queueing models to the case of multiple classes.

The three-state Markov considered in [15] above is also used in [1]. In this paper the end-to-end burst loss probability is calculated for a queueing network of K nodes where short and long bursts are transmitted. The short bursts occupy one wavelength per link, whereas the long bursts occupy wavelengths on two adjacent links. This network is decomposed to sub-systems and modeled using an iterative algorithm. Each subsystem is solved numerically.

A closed form solution to an OBS network with Time based burst assembly is given in [13]. This network consists of a scheduler and an OBS switch. Bursts for each destination are formed using the Time based aggregation every LT, where L is the number of destination queues and T is the burst assembly processing time. According to the authors these bursts depart according to a geometric distribution with a mean $1/(1 - e^{-\lambda T})$, where λ is the arrival rate of IP packets from the access network. Thus the burst inter-departure time A will be:

$$Pr[A = kT] = (1 - e^{-\lambda T})e^{-(k-1)\lambda T},\ k \geq 1$$

External burst arrivals are modeled using the Poisson distribution. Thus, the system that is analyzed is a $Geo, M/M/W/W$ system with W wavelengths and a capacity of W bursts. The authors solved this system numerically. With a view to calculating the burst loss probability and the throughput of the network.

In this paper, we study a hub-based OBS network consisting of N input edge nodes and a destination edge node. The input edge nodes and the destination edge node are connected to a bufferless OBS node, hereafter referred to as the core node. Each input edge node is connected to the core node via a single wavelength in the upstream direction, i.e., from the edge node to the core node, and a separate single wavelength in the downstream direction. The core node is connected to the destination edge node via w wavelengths in the upstream direction, i.e., from the core node to the destination edge node, and one wavelength in the downstream direction. As will be explained in the following section, this OBS network has a zero burst loss. Hub traffic is an important class of traffic and it arises in access and metro networks. A metro network is used to serve multiple access networks, where most of the traffic is destined out of the network towards a wide area network. The hub-based OBS network can be seen as serving the needs of such a metro network. Different hub-based optical networks have been proposed in the literature. For instance, in [11] and [2] a new architecture named Dual Bus Optical Ring Network (DBORN) has been proposed and analyzed.

We propose and analyze approximately a novel discrete-time queueing network model with blocking and bulk arrivals that models the hub-based OBS architecture in the upstream direction. (We note that the OBS network in the downstream direction can be easily analyzed approximately using existing decomposition techniques.) The discrete-time queueing network consists of N input queues linked to a single multi-server queue with finite capacity. The arrival process of bursts to the input queue is assumed to be slotted with bulk arrivals, as generated by a Time and Burst-Length aggregation algorithm. Blocking may occur when a customer attempts to move from an input queue to the finite capacity queue. We derive an upper bound on the mean end-to-end delay of a burst from the time it arrives at an input edge node to the time that it is delivered to the destination edge node. Queueing networks with blocking have been extensively analyzed, see [10]. Discrete-time queueing networks with blocking have also been analyzed, see [6], [8], [7], and [16]. To the best of our knowledge, the queueing network with blocking proposed and analyzed in this paper has not been studied before.

We note that modelling an OBS network by a discrete-time queueing network constitutes a departure from the previously published continuous-time queueing network models of OBS networks. As explained above, under the Time and Burst-Length burstification algorithm, the arrival process is a slotted bulk arrival of bursts. The size of each burst is equal to a maximum predefined size B_max, with the possible exception of the last burst in a bulk that may be less than B_max. Ignoring this, we see that all bursts are of constant length and consequently their transmission time is also constant, which necessitates a discrete-time model. In addition, due to the zero-loss feature of the OBS network, a burst in an input edge node is delayed until the core node can transmit it to the destination edge node without the possibility of loss. This necessitates the introduction of a blocking scheme in the queueing network (as understood in queueing networks with blocking, rather than in loss networks) in order to model this important event.

This paper is organized as follows. In Section 2 the queueing network is described in detail. In Section 3 we first provide an analysis of each edge node of the queueing network under study, and then we describe an iterative algorithm through which the results of the analysis of the individual edge nodes are combined in order to model the queueing network. The analytic results of the average end-to-end delay provide a tight upper bound as verified by simulation in Section 4. We conclude in Section 5.

2 The Queueing Network

The hub-based OBS network under study consists of N input and one output edge node, as shown in Figure 1. Time is slotted, and a time slot T_B is equal to the time required to transmit a burst of size B_{max} at a specified link speed. In the upstream direction, each input edge node is connected to the core node by a single wavelength and the core node is connected to the destination edge node by w wavelengths.

The bursts are aggregated according to the Time and Burst-Length based algorithm. The burst arrival process is a slotted bulk arrival process as derived in [9]. To the best of our knowledge, this is a unique feature of our analysis: using an analytical burst arrival process as it results from a burst aggregation

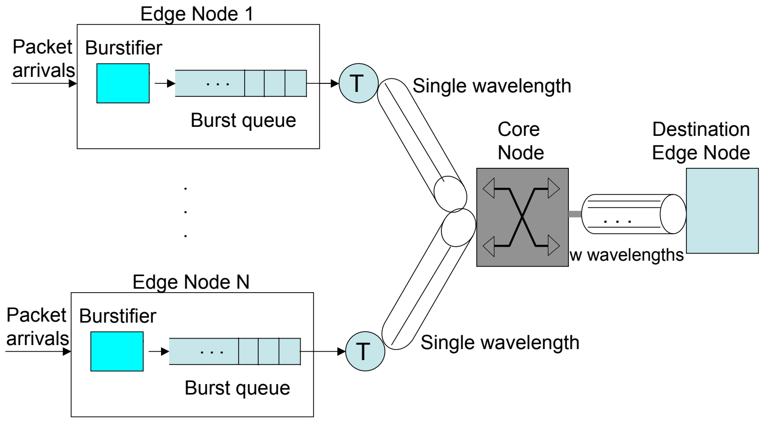


Fig. 1. The OBS network under study

algorithm. In other papers, the burst arrival process was assumed to be Poisson [4], [14], or 3-state Markov process that models short and long bursts [1], [15]. In our model k bursts arrive with probability $P[k]$ at the end of every aggregation period T, where T is assumed to be an integer number of T_B, i.e., $T = s \times T_B$, $s = 1, 2, ...$ The probability $P[k]$ is calculated by Equation 2 as shown below. These bursts are stored in the electronic domain in an infinite queue that resides at each input edge node.

Bursts within a burst queue are served in FIFO. Only one burst from each queue (the one at the top of the queue) can be served at a time. The core node scans the burst queues at the beginning of each time slot T_B in order to identify how many queues have a burst to transmit. (This is equal to the number of busy burst queues.) The time to scan the queues is assumed to be zero. (This in fact can be done prior to the beginning of each time slot.) If there are fewer (or equal) bursts to transmit than the number of wavelengths w, then they are all transmitted in the same time slot each on a different wavelength (We assume that the core node is equiped with w converters). Otherwise, if there are more than w bursts, a selection mechanism such as Horizon that gives priority to the earliest request or a round-robin scheme, can be used to select w bursts that will be transmitted in the time slot. We note that at any time slot up to w bursts are transmitted. In view of this, this scheme has a zero burst loss, since no more than one burst can use the same wavelength at any time slot.

We set the ratio of $\frac{N}{w}$ to a constant value in order to limit the delay a burst may suffer before it is transmitted to a maximum of one time slot. Specifically, let us assume that $\frac{N}{w} = 2$. At the beginning of time slot i, each edge node has one burst to transmit. Since we assume w wavelengths on the downlink from the core node to the output edge node, only w of these bursts can be transmitted in time slot i. The remaining $\frac{N}{2}$ bursts will be transmitted in the next $(i+1)^{st}$ time slot. Thus, a burst at the top of the burst queue of an input edge node can be delayed a maximum of one time slot before it is successfully transmitted. The same applies if $1 < \frac{N}{w} \leq 2$. In general, this delay is directly affected by the ratio of the $\frac{N}{w}$. If $\frac{N}{w} \leq 1$ then a burst at the top of the burst queue is transmitted without any delay. If $d-1 < \frac{N}{w} \leq d$, then a burst at the top of the burst queue may wait a maximum of $d-1$ time slots before it is transmitted.

The above described OBS network is modeled by the discrete time queueing network with blocking shown in Figure 2. Nodes 1 to N represent the N burst queues (one per input edge node) hereafter referred to as *input queues*. Node $N+1$ represents the WDM downlink from the core node to the destination edge node, as described in the following paragraph. We shall refer to this node as the *output queue*. The queueing network is slotted, and the length of a slot is equal to T_B. A customer in this queueing network represents a single burst. Bursts are all equal to B_{max}. (We assume here that the last burst produced during an aggregation period T is also equal to B_{max}). Bulk arrivals occur every $T = s \times T_B$ slots, where s takes integer values. The probability $P[k]$ that k bursts arrive in a single bulk is calculated from Equations 1 and 2:

$$P[0] = \int_0^{B_{min}-1} f_B(x)dx \tag{1}$$

$$P[k] = \int_{B_{min}+(k-1)B_{max}}^{B_{min}+kB_{max}-1} f_B(x)dx, \;\; k >= 1 \tag{2}$$

where:

$$f_B(x) = \sum_{n=1}^{\infty} \frac{e^{-\lambda T}(\lambda T)^n b(bx)^{n-1} e^{-bx}}{n!(n-1)!} \tag{3}$$

as shown in [9] where packet size is exponentially distributed with a mean $1/b$ and the arrival process of packets is Poisson with rate λ. Customers in the input queue j, $j = 1, 2, ..., N$, are served in FIFO manner. The service time represents the time it takes to transmit a burst.

The output queue, i.e. node $N+1$, consists of a finite capacity queue served by w servers. Each server represents a wavelength on the downlink from the core node to the output edge node. The service time at each server is equal to one time slot (T_B), and it represents the time that the core node has to block the appropriate downlink wavelength in order to allow the burst to come through the switch fabric without a loss. Now, let us consider that at the beginning of time slot i, a customer, i.e. a burst, is at the top of the input queue j, $j = 1, 2, ..., N$. Let us also assume that $\frac{N}{w} = 2$, and that the propagation delay

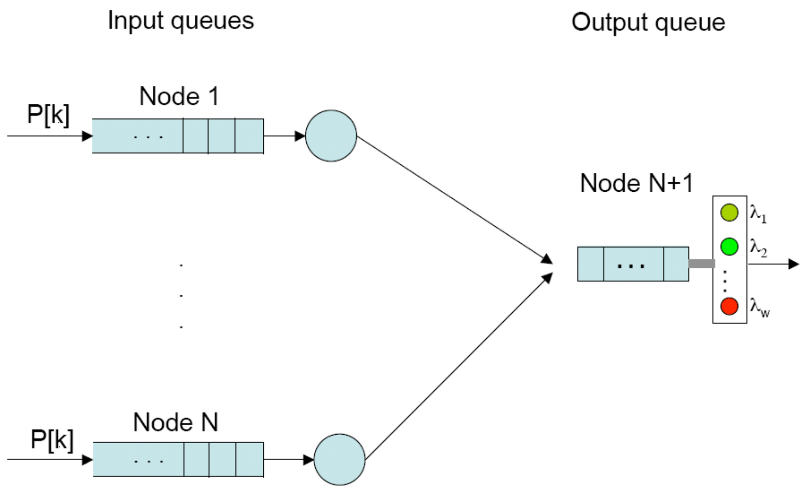


Fig. 2. The discrete-time queueing network with blocking

from an input edge node to the core node is one time slot T_B. Depending upon the total number of bursts that are ready to be transmitted out at this time slot, the burst may or may not be transmitted during the time slot i. As described above, if there are fewer bursts than w, then they are all transmitted in time slot i. Otherwise, the burst may be delayed by one time slot. This is because we have assumed that $\frac{N}{w} = 2$. If $\frac{N}{w} = d$, then it may be delayed up to d time slots.

In the queueing network model we assume that a customer at the top of input queue j, $j = 1, 2, ..., N$, receives its service and then moves to the output queue at the end of the time slot. If there is a free server in the output queue it occupies it in the next time slot, otherwise it joins the output queue. If $\frac{N}{w} = 2$, it will occupy the server in the next time slot. If $\frac{N}{w} = d$ it may be delayed up to d time slots. During the time it is in this queue, its corresponding server at node j is *blocked* in the sense that it cannot transmit any other bursts. The time the server remains blocked is known as the *blocking delay* and this type of blocking mechanism is referred to as *blocking after service*, see [10]. We analyze this queueing network by decomposition with a view to calculating the average end-to-end delay. The decomposition algorithm is described in detail in the Section 3.

We assume that the arrival process to each input queue j, $j = 1, 2, ..., N$, is identical. Also, we note that if k bursts, $k > w$, arrive at the output queue at time slot i, w randomly selected of these arrivals will be allocated to the w servers, and the others will be queued. This is equivalent to saying that if the number of bursts to transmit in the OBS network is greater than the number of available wavelengths on the downlink (OBS to output edge node), then the bursts to be transmitted are randomly selected.

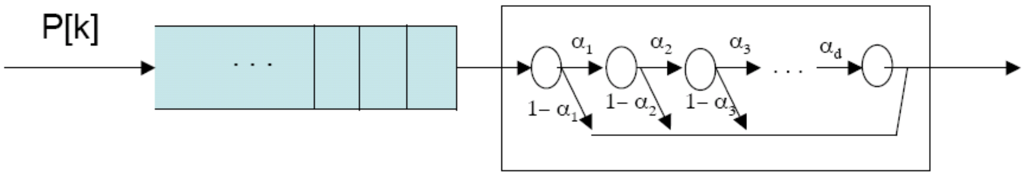


Fig. 3. The discrete-time input queue analyzed in isolation: service time consists of a maximum of d time slots, bulk arrivals every s time slots.

Finally, we note that this queueing network is a valid model under the assumption that the propagation delay from an input edge node to the core node is equal to or greater than one time slot T_B. If it is less than a time slot, then a different type of queueing network model is required that captures the notion of simultaneous resource possession, see [1]. In this queueing network model, the propagation delay from an input edge node to the core node is assumed to be one time slot T_B. The end-to-end delay of a burst defined from the moment it joins a burst queue at an input edge node to the moment it arrives at the destination edge node can be obtained by adding the balance of the end-to-end propagation delay to the delay of a burst in the input queue.

3 The Decomposition Algorithm

The queueing network described above does not have a product form solution. In view of this, we analyze it by decomposition. Specifically, each input queue j, $j = 1, 2, ..., N$ is analyzed independently as a discrete-time queue, as shown in Figure 3 by assuming that its service time is as follows. A burst receives service for one time slot and then with probability $1-\alpha_1$ it leaves the queue or with probability α_1 it receives another one time slot of service. Then the burst either leaves the queue with a probability $1 - \alpha_2$ or with a probability α_2 it receives another time slot of service. This is repeated until the burst leaves the queue after receiving service for $d-1$ time with a probability $1-\alpha_d$ or with a probability α_d receives the last d^{th} time slot. This service mechanism is similar

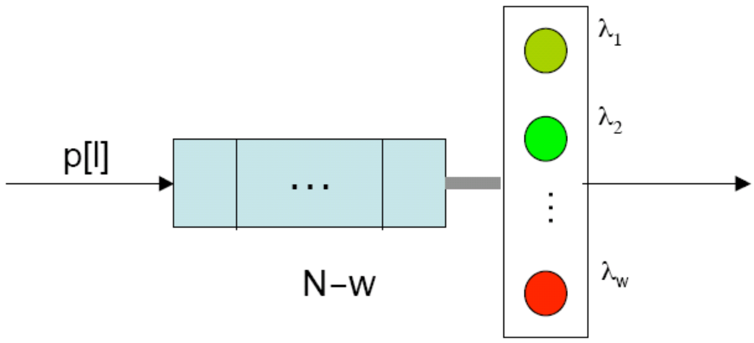


Fig. 4. The discrete-time output queue analyzed in isolation: finite capacity, w servers, slotted bulk arrivals.

to a Coxian distribution only the distribution at each phase is constant, equal to a time slot. The burst arrival process is the same as described previously. That is, it is a bulk arrival every s time slots, and the bulk size has a probability of $P[k]$ given by Equation 2. Since we assume that the arrival process to each input queue j, $j = 1, 2, ..., N$, is the same, it suffices to analyze only one of these N queues.

The output queue is modeled as a discrete-time finite capacity queue served by w servers. The queue size is equal to $N - w$ as shown in Figure 4. The arrival process is a slotted bulk arrival process that occurs every time slot T_B with probability $p[l|n]$ of having l bursts in a bulk, where $l = 0, 1, ..., N$, given that there are n bursts in the output queue, $n = 0, 1, ..., N$. In this decomposition method we need to know $\alpha_1, \alpha_2, ..., \alpha_d$ and $p[l|n]$, $l = 0, 1, ..., N$, $n = 0, 1, ..., N$ in order to analyze these queues in isolation. The probabilities $\alpha_1, \alpha_2, ..., \alpha_d$ are obtained from the analysis of the output queue and the probabilities $p[l|n]$ from the input queues. In view of this, we construct an iterative scheme whereby we first analyze an input queue assuming initial values for $\alpha_1, \alpha_2, ..., \alpha_d$. Based on the results obtained we construct an initial value of the bulk size conditional probabilities $p[l|n]$, $l = 0, 1, ..., N$, $n = 0, 1, ..., N$ and then we analyze the output queue. From this we construct updated values of $\alpha_1, \alpha_2, ..., \alpha_d$ and repeat the above process.

In subsection 3.1 we analyze an input queue, assuming we know $\alpha_1, \alpha_2, ..., \alpha_d$. In subsection 3.2 we analyze the output queue assuming that we know $p[l|n]$, $l = 0, 1, ..., N$, $n = 0, 1, ..., N$. Finally, in subsection 3.3 we describe the iterative scheme.

Table 1
State Transitions of the input queue

Transition States	**Rate**	**Condition**
$(0,0,0) \rightarrow (k,1,1)$	$P[k>0]$	
$(0,0,0) \rightarrow (0,0,1)$	$P[k=0]$	
$(n_q,d_q,0) \rightarrow (0,0,1)$	$P[k](1-\alpha_{d_q})$	$n_q+k-1=0$
$(n_q,d_q,0) \rightarrow (n_q+k-1,1,1)$	$P[k](1-\alpha_{d_q})$	$0<n_q+k-1 \leq MaxInQ$
$(n_q,d_q,0) \rightarrow (MaxInQ,1,1)$	$P[k](1-\alpha_{d_q})$	$n_q+k-1>MaxInQ$
$(n_q,d_q,0) \rightarrow (n_q+k,d_q+1,1)$	$P[k]\alpha_{d_q}$	$n_q+k \leq MaxInQ$
$(n_q,n_q,0) \rightarrow (MaxInQ,d_q+1,1)$	$P[k]\alpha_{d_q}$	$n_q+k>MaxInQ$
$(n_q,d,0) \rightarrow (0,0,1)$	$P[k]$	$n_q+k-1=0$
$(n_q,d,0) \rightarrow (n_q+k-1,1,1)$	$P[k]$	$n_q+k-1 \leq MaxInQ$
$(n_q,d,0) \rightarrow (MaxInQ,1,1)$	$P[k]$	$n_q+k-1>MaxInQ$
$(n_q,d_q,i) \rightarrow (0,0,(i+1)\ mod(s))$	$(1-\alpha_{d_q})$	$n_q-1=0$
$(n_q,d_q,i) \rightarrow (n_q-1,1,(i+1)\ mod(s))$	$(1-\alpha_{d_q})$	$n_q-1>0$
$(n_q,d_q,i) \rightarrow (n_q,d_q+1,(i+1)\ mod(s))$	α_{d_q}	
$(n_q,d,i) \rightarrow (n_q-1,1,(i+1)\ mod(s))$	1	$n_q-1>0$
$(n_q,d,i) \rightarrow (0,0,(i+1)\ mod(s))$	1	$n_q-1=0$

3.1 Analysis of an Input Queue

This is a discrete time queue as depicted in Figure 3. The service time is one time slot (T_B) followed by d time slots of blocking delay with probability $\sum_{i=1}^{d}\alpha_i$. Bulk arrivals occur every period $T=s\times T_B$, where $s=1,2,...$ and the probability that the bulk size is k is $P[k]$, calculated as described in Section 2. Bursts are served individually in a FIFO manner.

This queue is analyzed numerically. The state of the system is described by a three-tuple vector (n_q,d_q,i), where n_q is the number of bursts in the queue, including the one that is in service at the beginning of a time slot T_B, d_q indicates the state of the server, and it takes the values $1,2,...,d$ (server in the d_q^{th} service time slot), and $i=0,1,...,s-1$ is the current time slot. The input queue is infinite, but in order to solve the system numerically we assume that $n_q=0,1,...,MaxInQ$, where $MaxInQ$ is the maximum number of bursts in the input queue. The $MaxInQ$ is selected so that the probability of having $n_q>MaxInQ$ is almost zero.

In order to calculate the steady-state probability vectors, we need to generate the rate matrix $P_{j,InQ}$ for each input queue $j = 1, 2, ..., N$. Since all input edge nodes are the same, we omit the index j for simplicity. Every timeslot i, where $i\ mod(s) = 0$, there are $k = 0, 1, ..., k_{max}$ arrivals with probability $P[k]$. Immediately after these arrivals occur, the core node scans for requests at the input queues and schedules the bursts that are on the top to be served or stored at the output queue. We observe the system at time $T_B + \Delta t$, where Δt indicates a small time interval after the burst arrivals and departures have occurred. It is obvious that there are no arrivals if $i\ mod(s) \neq 0$. The state transition table of the input queue is given in Table 1. In this table we assume that when $i = 0$ we have k arrivals with a probability $P[k]$ and that these arrivals occur every s time slots, i.e. $i\ mod(s) = 0$. We observe that at time slot zero we may have the following state transitions:

- If $n_q > 0$ and the burst that is on service is at the first time slot of service, i.e. $(n_q, d_q, 0), d > d_q \geq 1$, and there are k arrivals with probability $P[k]$ then:
  - · The number of bursts in the queue will become n_q+k-1 with a probability $1 - \alpha_{d_q}$, which means that the burst being served is not blocked for an additional time slot.
  - · The number of bursts in the queue will become $n_q + k$ with a probability α_{d_q}, which means that the burst at the top of the queue will be blocked for an additional time slot. Also $d_q = d_q + 1$ which indicates that the burst that is on top of the queue at the current time slot is blocked for an additional time slot.
- If $n_q > 0$ and the burst on service is blocked, i.e. $(n_q, d, 0)$, and there are k arrivals with probability $P[k]$, then the number of bursts at the input queue will become $n_q + k - 1$ after Δt with a probability $P[k]$. Also the blocked burst will depart with probability one, which means it cannot be blocked for more time slots.

The state transitions when there are no arrivals, i.e. $i \neq 0$ are more straight forward. If the burst at the top of the queue is at the d_q^{th} service time slot ($d_q \geq 1$) it will either be blocked for an additional time slot $(n_q, d_q + 1, (i + 1)\ mod(s))$, or it will depart with a probability $1 - \alpha_{d_q}$ and the next state will be $(n_q - 1, 1, (i + 1)\ mod(s))$ if $n_q - 1 > 0$, or $(0, 0, (i + 1)\ mod(s))$ if there are no bursts left in the input queue. When the burst on top of the queue is at the d^{th} service time slot ((n_q, d, i) then it will depart with probability equal to one.

Table 1 indicates that the analysis of the input queue depends on the burst blocking probabilities $\alpha_1, \alpha_2, ..., \alpha_d$. These probabilities are derived by the analysis of the output queue as described in Section 3.3.

Based on Table 1 we generate the rate matrix P_{InQ}. Then we solve the following system of linear equations numerically using the Gauss-Seidel method [12]:

$$\pi_{(n_q,d,i)} P_{InQ} = \pi_{(n_q,d,i)} \tag{4}$$

where $\pi_{(n_q,d,i)} = (\pi_{(0,0,0)}, \pi_{(0,0,1)}, ...\pi_{(MaxInQ,d,s-1)})$ is the probability of being at state (n_q, d, i). The system space consists of the following states:

$$(0,0,0),(0,0,1),...,(0,0,s-1)$$

$$\begin{gathered}(1,1,0),(1,1,1),...,(1,1,s-1)\\(1,2,0),(1,2,1),...,(1,2,s-1)\\...\\(1,d,0),(1,d,1),...,(1,d,s-1)\\...\\(MaxInQ,1,0),(MaxInQ,1,1),...,(MaxInQ,1,s-1)\\(MaxInQ,2,0),(MaxInQ,2,1),...,(MaxInQ,2,s-1)\\...\\(MaxInQ,d,0),(MaxInQ,d,1),...,(MaxInQ,d,s-1)\end{gathered}$$

Thus the total number of states: $TotalNumStates = MaxInQ \times s \times d$. The marginal probability distribution of having n_q bursts in the input queue is:

$$\pi_{n_q} = \sum_{d_q=1}^{d} \sum_{i=0}^{s-1} \pi_{(n_q,d_q,i)}$$

From the steady-state probabilities of this system we can calculate the probability $p[l]$ of having l arrivals at the output queue, as described in Section 3.3.

3.2 Analysis of the Output Queue

The output queue is a discrete-time $D^{[l]}/D/w/N-w$ queue, and it has slotted bulk arrivals with probability $p[l|n]$ every time slot T_B, slotted departures and consists of w servers. The service time is one time slot equal to T_B. The sequence of events for the output queue is: every timeslot i, there are $l = 0, 1, ..., N$ arrivals with probability $p[l|n]$ given that there are n bursts in the output queue, $n = 0, 1, ..., N$ (including those in service). These l bursts are either stored at the output queue or assigned to downlink wavelengths depending on their availability. If n bursts were already at the output queue at time slot i, then w bursts will be served and $l+n-w$ bursts will be stored and served on the next time slot $i+1$. We observe the system at time $T_B + \Delta t$, where Δt indicates a small time interval after the bulk arrival and departures have occurred.

The rate matrix P_{OutQ} for the $D^{[l]}/D/w/N-w$ queue has elements:

$$p_{ij} = Prob[l = j|n = i] \tag{5}$$

When there are n customers already in the output queue this means that there can only be a maximum $l = N - n$ input queues that can transmit a burst to the output queue. The solution of the output queue is obtained by solving numerically the following system of linear equations using the Gauss-Seidel method [12]:

$$\pi P_{OutQ} = \pi \tag{6}$$

where $\pi = (\pi_0, \pi_1, ...\pi_N)$ is the probability of having $n = 0, 1, ..., N$ bursts at the output queue.

3.3 The Iterative Algorithm

In this Section we use the solutions to the two single nodes described above to construct an iterative algorithm that calculates the average end-to-end delay of a burst. Initially we use a guess value for the blocking probabilities $\alpha_1, \alpha_2, ..., \alpha_d$. We use these value to construct the rate matrix P for the input queue according to Table 1. Then we use the Gauss-Seidel method to calculate the steady state probabilities $\pi_{(n_q,d_q,i)}$ as discussed in Section 3.1.

Now we can calculate the conditional probabilities $p[l|n]$ that are used to construct the rate matrix of the output queue. First we need to calculate the probability of an arrival at the output queue of a burst that resides at the input queue. This means that the burst is at the top of the input queue and on its first time slot of service, i.e. $d = 1$. This probability is calculated by the following sum:

$$q = \sum_{i=0}^{s-1} \sum_{n_q=0}^{MaxInQ} \pi_{(n_q,1,i)} \tag{7}$$

Then the number of bursts that are already in the output queue, i.e. they are blocked, is calculated similarly:

$$\beta = \sum_{i=0}^{s-1} \sum_{d_q=2}^{d_q=d} \sum_{n_q=0}^{MaxInQ} \pi_{(n_q,d_q,i)} \tag{8}$$

Finally the probability that an edge node has no bursts to transmit to the output queue at a time slot is:

$$c = \sum_{i=0}^{s-1} \pi_{(0,0,i)} = 1 - q - \beta. \tag{9}$$

Using these probabilities we calculate the conditional probability $p[l|n]$ as follows:

$$p[l|n] = \begin{pmatrix} N \\ n \end{pmatrix} \beta^n \begin{pmatrix} N-n \\ l \end{pmatrix} q^l c^{N-n-l}. \tag{10}$$

Using these probabilities and the rate matrix P_{OutQ} we analyze numerically the output queue with the Gauss-Seidel method.

The blocking probabilities $\alpha_1, \alpha_2, ..., \alpha_d$ are calculated as the sum of having more than w bursts at the output queue, since we can only serve w bursts within one time slot. Thus any arrival will be blocked when there are already w bursts in the output queue. Also, bursts may be blocked when there are n bursts in the output queue, where $n < w$, and l bursts arrive from the input edge nodes, where $l + n > w$. In this case the blocked bursts are the ones that are not served according to FCFS. This means that they will not be served if they are not $1^{st},\ 2^{nd}, ..., (w-n)^{th}$ in a bulk of l bursts. Thus the probability of a burst to be blocked is given by:

$$\alpha_1 = \sum_{l=0}^{N} \sum_{i>w} \pi_i +$$

$$\sum_{i \leq w} \pi_i Prob[not\ 1^{st},\ 2^{nd}, ..., (w-n)^{th}] p[l]$$

$$\alpha_2 = \sum_{l=0}^{N} \sum_{i>w} \pi_i +$$

$$\sum_{i \leq w} \pi_i Prob[(w+1)^{th}, ..., (2w-n)^{th}] p[l]$$

$$...$$

where π_i is the steady-state probability calculated by solving the system of linear equations in 6. Since bursts have the same probability to be at the $1^{st},\ 2^{nd}, ..., (w-n)^{th}$ position, the $Prob[not\ 1^{st}, not\ 2^{nd}, ...,\ not\ (w-n)^{th}]$ is calculated as follows:

$$Prob[not\ 1^{st}] = 1 - \frac{1}{l}$$

$$Prob[not\ 1^{st}\ and\ not\ 2^{nd}] = (1 - \frac{1}{l})(1 - \frac{1}{l-1})$$

$$...$$

$$Prob[not\ 1^{st}\ and\ not\ 2^{nd}, ...,\ and\ not\ (w-n)^{th}] =$$

$$(1 - \frac{1}{l})(1 - \frac{1}{l-1})...(1 - \frac{1}{l-(w-n)-1})$$

In order to calculate the average end-to-end delay of a burst that arrives in a bulk of k with probability $P[k]$ we calculate the average delay of a burst within a bulk of size k. Subsequently, we uncondition on k. The delay of the first burst is given by:

$$D_1 = Prob[burst\ in\ first\ service\ slot] \times \sum_{i=2}^{d} \alpha_i +$$

$$Prob[burst\ in\ 2^{nd}\ service\ slot] \times \sum_{i=3}^{d} \alpha_i + ...$$

$$+Prob[burst\ in\ d^{th}\ service\ slot] \times 1 + D_{other\ bursts\ in\ queue}$$

$$D_1 = \sum_{n_q=1}^{MaxInQ} \pi_{(n_q,1,0)}(1 + \sum_{i=2}^{d} \alpha_i) +$$

$$\pi_{(n_q,2,0)}(1 + \sum_{i=3}^{d} \alpha_i) + ... + (n_q - 1)(1 + \sum_{i=1}^{d} \alpha_i)$$

where $1 + \alpha$ is the average service time of a burst at the input queue. Now the delay of the other bursts within the bulk of k bursts is given by:

$$D_2 = D_1 + (1 + \sum_{i=1}^{d} \alpha_i)$$

$$D_3 = D_1 + 2(1 + \sum_{i=1}^{d} \alpha_i)$$

$$...$$

$$D_k = D_1 + (k - 1)(1 + \sum_{i=1}^{d} \alpha_i)$$

Adding these we calculate the average per burst delay:

$$\overline{D}_k = \frac{1}{k} \sum_{i=1}^{k} D_i$$

Finally the average end-to-end delay is given by:

$$D = \sum_{k=0}^{k_{max}} \overline{D}_k P[k] \qquad (11)$$

where k_{max} is the maximum number of bursts that arrive at the input edge node within an aggregation period.

The iterative algorithm is given below:

Initial Step
Guess value for $\alpha_1, \alpha_2, ..., \alpha_d$
Iterative Step
//Generate P_{InQ} *for* $T^{[k]}/G/1$ *queue*
use Table 1, $\alpha_1, \alpha_2, ..., \alpha_d$

$// \; Solve \; for \; steady - state \; probability \; T^{[k]}/G/1 \; queue$
$\pi_{(n_q,d_q,s)}$ use Equation 4
$// \; Calculate \; p[l|n]$
$q, \; \beta, \; c$ use Equation 10
$// \; Generate \; P \; for \; D^{[l]}/D/w/N - w \; queue$
use $q, \; \beta, \; c$ and Equation 5 for P_{OutQ}
$// \; Solve \; for \; steady - state \; probability \; D^{[l]}/D/w/\frac{N}{d} \; queue$
π_i use Equation 6
$// \; Generate \; \alpha, \; D_1, \; D$
use Equation 11, $\alpha_1, \alpha_2, ..., \alpha_d$
Repeat until blocking probability α_{d_q} converges

We compare α_{d_q} to its value from the previous iteration and if their difference is less than ϵ, then the iterative algorithm converges. The values of the probability $p[l]$ can also be used as a condition of convergence of the iterative algorithm. We have observed empirically that it takes the same number of total iterations for the algorithm to converge using either α_{d_q} or $p[l]$ as a convergence condition.

4 Numerical Results

In this Section we present numerical results and compare these to simulation results. We show that the average end-to-end delay calculated by our iterative algorithm provides a tight upper bound. We define the end-to-end delay of a burst as the number of time slots T_B that elapse from the moment it joins an input queue to the moment it departs from the output queue.

The decomposition algorithm described in Section 3.3 provides an upper bound to the end-to-end delay. This is because we overestimate the bulk size that arrives at the output queue. Specifically, assume that there are $n > w$ bursts in the output queue at time slot i. At time $i + 1$, $n - w$ bursts will be served and thus $n - w$ input queues will be unblocked. If $n < w$ then all n bursts that reside at the output queue are served and n input queues are unblocked. Once an input queue is unblocked, it will take a minimum of one time slot before it can complete a new service. That is before a new burst is ready to move into the output queue. This is not taken into consideration in our iterative algorithm. For example if there are n bursts in the output queue during the current time slot i we assume that $N - n$ input queues are able to send a burst to the output queue, which is not always the case.

The simulation program for the queueing network was written in C. We use the method of batch means to calculate the steady state delay in each queue, the blocking probability α_{d_q} and the probability $p[l]$ of having l arrivals at the output queue every time slot T_B. The confidence intervals are very small and not discernible in the graphs. The decomposition algorithm was implemented in Matlab. The analytical model runs in a fraction of a second compared to the simulation which, depending on the number of edge nodes N, takes 5-10 minutes to run for 30 batches with 100,000 bursts per batch.

We obtained results for the average end-to-end burst delay for varying traffic intensity. The variables that affect the traffic intensity are: *i*) the number of slots s within an aggregation period; *ii*) the maximum number of bursts that arrive at the input edge node k_{max}; *iii*) the burst distribution, $P[k]$ and *iv*) the number of input edge nodes, N. The latter does not affect our results, as long as the ratio $\frac{N}{w}$ remains constant. That is, if the number of input edge nodes increases, then the number of downlink wavelengths has to increase accordingly.

We have the following condition of stability for the queueing network:

$$k_{avg} \times N \leq s \times w$$

where k_{avg} is the average number of bursts that arrive every aggregation period T, N is the number of input edge nodes, s is the number of time slots T_B per aggregation period T and w the number of output wavelengths. This means that the maximum number of bursts that arrive at each input edge node every aggregation period T, i.e. the average arrival rate to the system, should be less than the maximum number of bursts that can depart from the queueing network during an aggregation period. The average arrival rate to an input burst queue at each time slot T_B is:

$$Average\ Arrival\ Rate = \frac{1}{s} \sum_{k=0}^{k_{max}} k \times P[k]$$

and the average departure rate from an input queue at each time slot T_B:

$$Average\ Departure\ Rate = \frac{1}{1 + \alpha_{d_q}}$$

The traffic intensity of an input queue is given by the ratio:

$$\rho = \frac{Average\ Arrival\ Rate}{Average\ Departure\ Rate} = \frac{\frac{1}{s}\sum_{k=0}^{k_{max}} k \times P[k]}{\frac{1}{1+\alpha_{d_q}}}$$

Assume that there are $N = 16$ input edge nodes and $w = 8$ output wavelengths so that $\frac{N}{w} = 2$. Also assume that the distribution of the number of bursts that arrive at an input edge node is uniform, with $E[k] = \sum_{k=0}^{k_{max}} k \times P[k] = \frac{k_{max}}{2}$. Figures 5 (a) and 5 (b) show the average end-to-end delay versus the maximum

Table 2
Traffic Intensity ρ for Analytical and Simulation Model, $N = 16$, $k_{max} = 4$, Uniform Distribution of Bulk Arrivals

Slots/Aggregation Period	ρ **Analytical**	ρ **Simulation**
8	0.9999	0.9999
16	0.2013	0.1876
32	0.0948	0.0938
64	0.047	0.0469
128	0.0234	0.0235

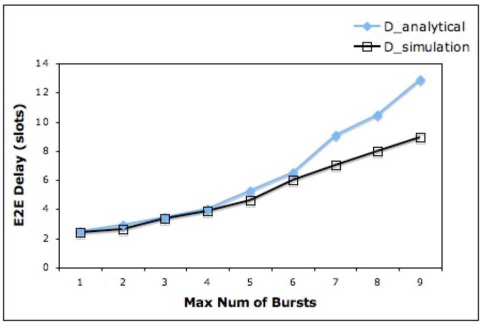
Table 3
Traffic Intensity ρ for Analytical and Simulation Model, $N = 16$, $s = 16, 32$

k_{max}	$s = 16$		$s = 32$	
	ρ Analytical	ρ Simulation	ρ Analytical	ρ Simulation
1	0.0942	0.0938	0.0469	0.0469
2	0.1269	0.125	0.0627	0.0625
3	0.1618	0.1563	0.0786	0.0782
4	0.2013	0.1876	0.0948	0.0938
5	0.2499	0.2189	0.1115	0.1095
6	0.3169	0.2502	0.1287	0.1251
7	0.6129	0.61	0.1468	0.1408
8	0.8945	0.845	0.1663	0.1565
9	0.912	0.9886	0.1872	0.1716

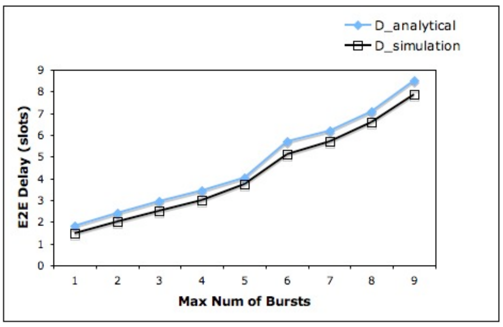
number of bursts k_{max} when the number of slots per aggregation period is $s = 16, 32$ respectively. The first observation from these graphs is that the upper bound given by our analytical model is very tight. It is also easily observed that when the traffic intensity increases, there are deviations between our model and the simulation results. Table 2 gives the traffic intensity for these two graphs. When the number of slots per aggregation period decreases, the traffic intensity ρ increases. Thus, we observe higher traffic intensities in Figure 5 (a) where $s = 16$ than in Figure 5 (b) where $s = 32$. Moreover, when the maximum number of bursts k_{max} increases the traffic intensity increases. Therefore, we observe deviations of the analytical model when $s = 16$ and $k_{max} > 6$, as shown in Figure 5 (a). This is justified in Table 2 where a traffic intensity close to one is observed for $s = 16$ and $k_{max} > 6$. In Table 3 that depicts the traffic intensity for this scenario, we observe that the deviations of the analytical results from the simulation results for $s = 8,\ 16$ correspond to high values of ρ. Figure 6 (a) and 6 (b) use a geometric and a shifted

Table 4
Traffic Intensity ρ for Analytical and Simulation Model, $N = 16$, $k_{max} = 4$

Slots per	Geometric Distribution		Shifted Geometric Distribution	
Aggregation Period	ρ Analytical	ρ Simulation	ρ Analytical	ρ Simulation
8	0.9999	0.9999	0.9999	0.9999
16	0.2399	0.2179	0.1651	0.158
32	0.1102	0.1089	0.0796	0.079
64	0.0544	0.0544	0.0395	0.0395

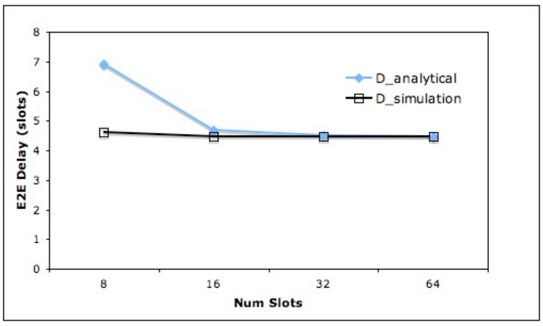


(a) $s = 16$

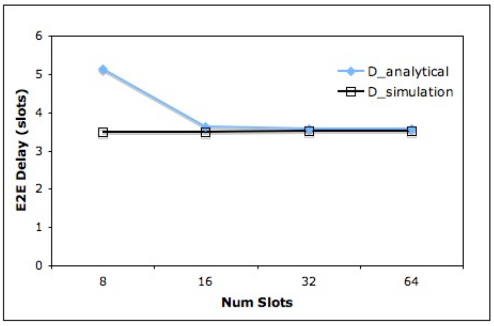


(b) $s = 32$

Fig. 5. Average end-to-end burst delay vs. maximum number of bursts per bulk. Uniformly distributed bulk size, $N = 16$.



(a) Geometric distribution of bulk arrivals



(b) Shifted Geometric distribution of bulk arrivals

Fig. 6. Average end-to-end burst delay vs. number of slots s, $N = 16$, $k_{max} = 4$, $d = 2$.

geometric distribution respectively for the bulk size, with $k_{max} = 4$, $N = 16$, $p = 0.03$ and $\frac{N}{w} = 2$ for the geometric distribution. The pdf of the geometric distribution is given by:

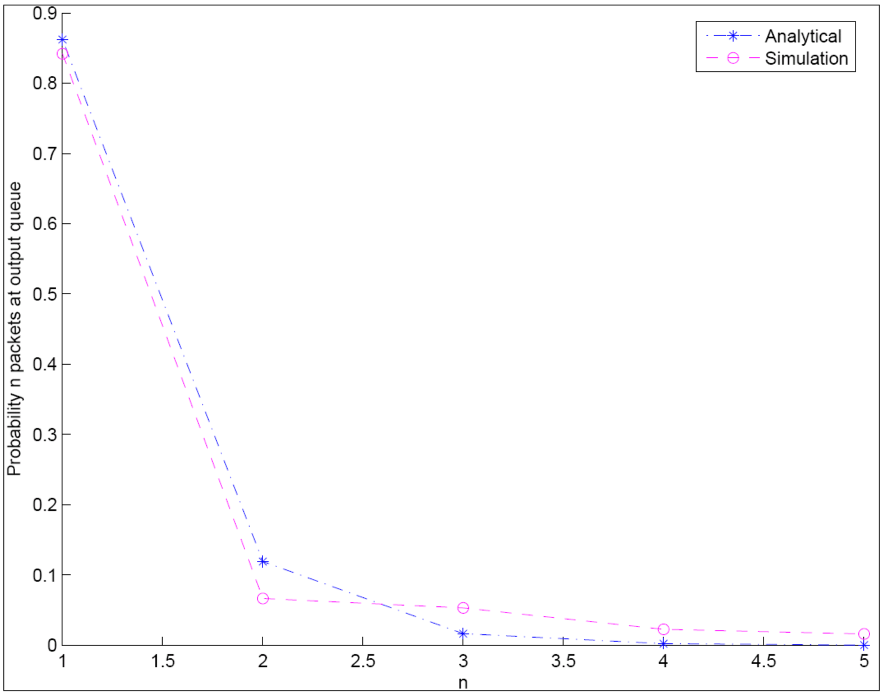
$$P(k) = p(1-p)^{k-1}, k = 1, 2, ..., k_{max} \text{ } or \text{ } 0 \text{ } otherwise$$

and the shifted geometric is:

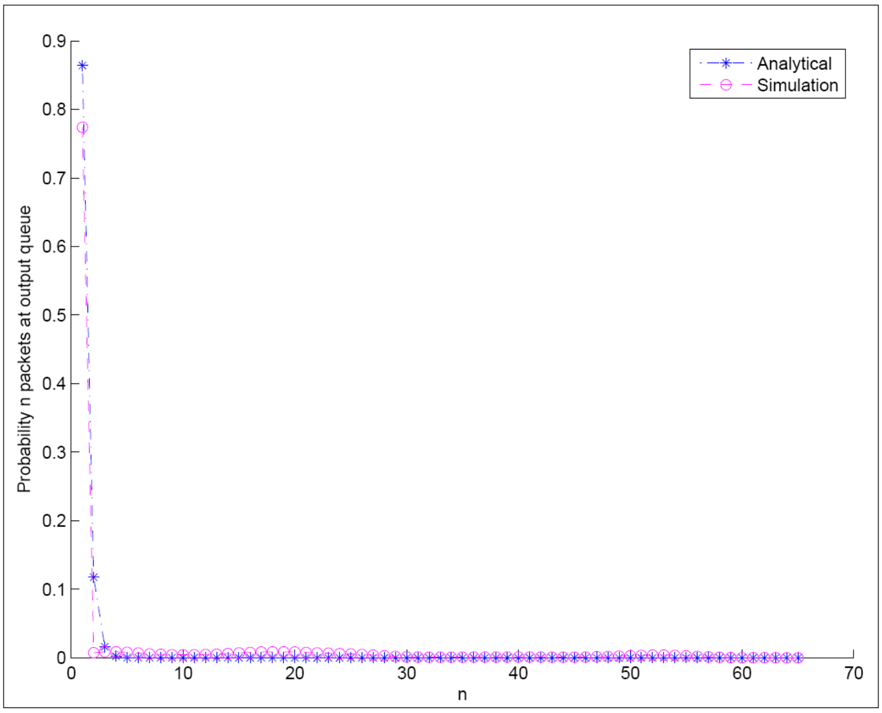
$$P(k_{max} - k + 1) = p(1-p)^{k-1}, k = 1, 2, ..., k_{max} \text{ } or \text{ } 0 \text{ } otherwise$$

In Figures 6 (a) and 6 (b) we vary the number of slots per aggregation period and observe that the bound provided by the analytical model is a tight upper bound. When the number of slots increases, the traffic intensity decreases as shown in Table 4. Therefore, even when the number of bursts k in a bulk is geometrically distributed, we observe the same behavior in our system. That is the analytical results are almost identical to simulation for $\rho < 30\%$. The bound becomes less tight as ρ increases. Now when the number of input edge nodes varies, the probability π_i of having i bursts in the output queue varies as well. This is shown in Figures 7 (a) and (b) when $N = 4,\ 64$ respectively, $\frac{N}{w} = 2$, $s = 32$ and $k_{max} = 4$ for uniformly distributed bulk sizes. It is observed that the accuracy of our analytical model increases when the number of input edge nodes increases. The traffic intensities for this example are given in Table 3 (3^{rd} line where $s = 32$ and $k_{max} = 4$). The average end-to-end delay for varying number of edge nodes is shown in Figure 8. We also observe that the average end-to-end delay does not change significantly when the number of edge nodes increases. This is justified since we have assumed that the ratio $\frac{N}{w} = 2$ is constant and therefore the stability condition: $k_{avg} \times N \leq s \times w$ is not affected when the bulk arrival distribution and the number of slots per aggregation period remain the same.

Figures 9 (a) and (b) show the probability π_i of having i, $i = 0, 1, ..., 16$ bursts in the output queue when $N = 16$, $w = 8$, $s = 32$, assuming a uniform



(a) $N = 4$



(b) $N = 64$

Fig. 7. Probability of having i bursts in the output queue, $k_{max} = 4$, $s = 32$.

burst size distribution, with $k_{max} = 4, 10$ respectively. These graphs indicate that when the traffic intensity increases, i.e. k_{max} increases, the approximation given by our analytical model is less close to the simulation (see Figure 9 (b)).

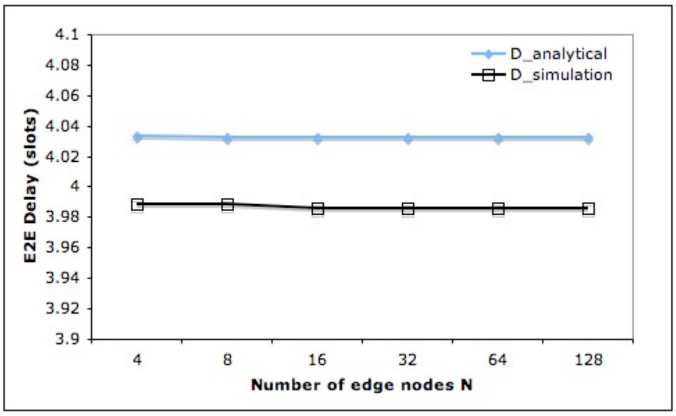


Fig. 8. Average end-to-end burst delay vs. number of edge nodes N in aggregation period T. Uniformly distributed bulk size, $s = 32$, $k_{max} = 4$

This is because the systems traffic intensity is very high when $k_{max} = 10$. The traffic intensities for this example are given in Table 2.

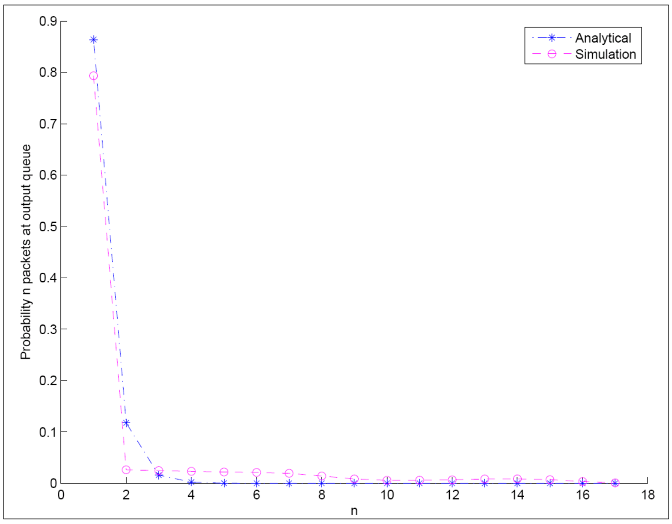
5 Conclusions

We proposed and analyzed a novel discrete-time queueing network model of a hub-based OBS network. We obtained approximately the average end-to-end delay of a burst in the queueing network, the blocking probability of a burst, and the probability $p[l]$ of having l arrivals at the output queue every time slot T_B. The approximate analysis requires a fraction of time to run compared to simulation. Through numerical validations we showed that our iterative algorithm provides an upper bound for different bulk burst arrival distributions, such as uniform and geometric distribution. We also showed that it provides an upper bound when the distribution $P[k]$ is kept the same and the number of slots per aggregation period s, or the maximum number of bursts per bulk k_{max} vary. The upper bound is very tight when the traffic intensity $\rho < 0.30$.

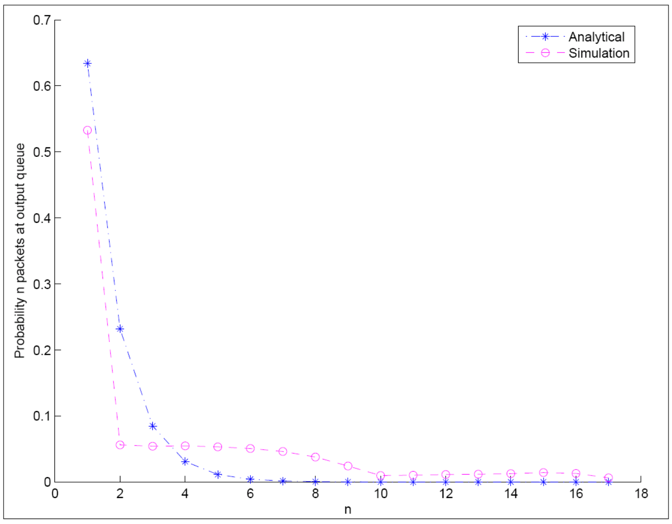
References

[1] L. Battestilli and H. Perros. A performance study of an optical burst switched network with dynamic simultaneous link possession. *Computer Networks Journal*, 50:219–236, 2006.

[2] N. Bouabdallah, G. Pujole, and H. Perros. Multipoint-to-point lightpath in all-optical networks: Dimensioning and cost analysis. *Performance Evaluation*, 65:262–285, 2007.



(a) $k_{max} = 4$



(b) $k_{max} = 10$

Fig. 9. Probability of having i bursts in the output queue, $N = 16$, $s = 32$.

[3] H. M. Chaskar, S. Verma, and R. Ravikanth. A framework to support IP over WDM using optical burst switching. In *IEEE/ACM/SPIE Optical Network Workshop*, January 2000.

[4] K. Dolzer, C. Gauger, J. Spath, and S. Bodamer. Evaluation of reservation mechanisms for optical burst switching. *AEU International Journal of Electronics and Communications*, 55(1), January 2001.

[5] IBM. Websphere Software. Available at: http://www.csc.ncsu.edu/faculty/perros/books.html.

[6] D. Kouvatsos, N. Tabet-Aouel, and S. Denazis. Approximate analysis of discrete-time queueing networks with or without blockig. *IFIP Transactions, special issue on High-Speed Networks and their Perfromance, North Holland*, C-21:399–434, 1994.

[7] T. Morris and H. Perros. Approximate analysis of a discrete-time tandem network of cut-through queues with blocking and bursty arrivals. *Performance Evaluation*, J-17:207–223, 1993.

[8] T. Morris and H. Perros. Performance analysis of a multi-buffered banyan atm switch under bursty traffic. In *IEEE Transactions of Communications*, volume 42, pages 891–895, 1994.

[9] X. Mountrouidou and H. G. Perros. Characterization of the burst aggregation process in optical burst switching. In *Proc. of IFIP Networking 2006*, pages 752–764, 2006.

[10] H. G. Perros. *Queueing Networks with Blocking, Exact and Approximate Solutions.* Oxford University Press, 1994.

[11] N. L. Sauze, A. Dupas, E. Dotaro, L. Ciavaglia, M. Nizam, A. Ge, and L. Dembeck. A novel, low cost optical packet metropolitan ring architecture. In *Proceedings of ECOC '01*, volume 1, pages 66–67, October 2001.

[12] W. Stewart. *Introduction to the Numerical Solution of Markov Chains.* Princeton University Press, 1994.

[13] T. Tachibana, T. Ajima, and S. Kasahara. Round-robin burst assembly and constant transmission scheduling for optical burst switching networks. In *GLOBECOM'03*, number 1, pages 2772–2776, December 2003.

[14] J. Wei and R. McFarland. Just-in-time signaling for WDM optical burst switching networks. *Journal of Lightwave Technology*, 18(12):2019–2037, December 2000.

[15] L. Xu and H. Perros. A queueing network model of an optical burst switching node. *Elsevier Performance Evaluation*, 64(6):315–346, May 2007.

[16] A. Zaghloul and H. Perros. Approximate analysis of a shared-medium atm switch under bursty arrivals and nonuniform destinations. *Performance Evaluation*, J-21:111–129, 1994.