

A Queueing Network Model of an Edge Optical Burst Switching Node

Lisong Xu, Harry G. Perros and George N. Rouskas

Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7534

Abstract— We consider an edge optical burst switching (OBS) node with or without converters, and with no buffering. The OBS node serves a number of users, each connected to the switch over a fiber link that supports multiple wavelengths. Each wavelength is associated with a 3-state Markovian burst arrival process. The arrival process permits short and long bursts to be modeled. We model the edge OBS node as a closed non-product-form queueing network, with multiple heterogeneous classes, and we develop a suite of approximate decomposition algorithms to analyze it. Our approximate algorithms have a good accuracy, and they provide insight into the effect of various system parameters on the performance of the edge OBS node.

I. INTRODUCTION

Optical burst switching (OBS) is a WDM-based technology positioned between wavelength routing (i.e., circuit switching) and optical packet switching. The unit of transmission is a burst whose length in time is arbitrary. The transmission of each burst is preceded by the transmission of a control packet, which usually takes place on a separate signaling channel. Unlike wavelength routing, a source node does not wait for confirmation that an end-to-end connection has been set-up. Instead it starts transmitting a data burst after a delay (referred to as *offset*), following the transmission of the control packet. The purpose of the control packet is to inform each intermediate node of the upcoming data burst so that it can configure its switch fabric in order to switch the burst to the appropriate output port. In this work, we assume that an OBS node has no buffers. In view of this, in case of output port conflict, it may drop a burst.

The performance of OBS has been studied by several authors. To the best of our knowledge, most performance studies of OBS networks are based on either simulation or simple analytical models under the assumption that the burst arrival process is Poisson. In [1]–[4], an output port of an OBS node is analyzed assuming Poisson arrivals and no buffering. Under these assumptions, an output port can be modeled by a finite number of servers, each representing a wavelength, with no queue. Then, the probability that a burst destined to this output port is lost can be obtained from the Erlang B formula. In [5], [6], an output port is analyzed assuming Poisson arrivals and buffering. It is then modeled by an $M/M/m/K$ queue, where m is the number of wavelengths and $K - m$ is the capacity of the buffer. Wei *et al.* [7] considered multiple classes of bursts, each of which is a Poisson arrival process. It is well-known that the Poisson process is not a good model for wide

area traffic, and it is unlikely that the burst arrival processes in future optical networks will be accurately characterized by the Poisson model. Therefore, more sophisticated models are required in order to advance our understanding of the performance and the potential of OBS networks.

In this paper, we develop for the first time a queueing network model of an edge OBS node with burst arrival processes described by more general Markov processes. The edge OBS node serves a number of users, each connected to the node by a fiber link which can support multiple wavelengths. Each wavelength is associated with a burst arrival process described by a 3-state Markovian model; the parameters of the model can be selected to capture a wide range of scenarios of the arrival stream. We consider an OBS edge node both with and without converters, and we model it as a closed non-product-form queueing network which we analyze by decomposition. We develop algorithms for both the single-class case, in which all users have the same arrival process, and the multi-class case, whereby each user has a different arrival process. Finally, we use our algorithms to gain new insight into the performance of an edge OBS node.

Following this introduction, we describe briefly the operation of an edge OBS node in Section II. In Section III, we present the burst arrival process used in the queueing network model described in this paper. In Section IV, we describe a queueing network model of the edge OBS node. Sections V and VI describe an algorithm for analyzing this queueing network without and with wavelength converters, respectively, assuming a single-class of customers. In Section VII, we present a new decomposition method for analyzing a multi-class generalization of this queueing network. We validate the accuracy of the approximation algorithm in Section VIII by comparing it to simulation results, and we conclude the paper in Section IX.

II. THE EDGE OBS NODE

The authors participate in the JumpStart project [8], a joint NCSU/MCNC research effort addressing the design, specification, performance evaluation, and hardware implementation of a signaling protocol for OBS networks. The signaling protocol follows the just-in-time (JIT) approach, and is based on the work by Wei and McFarland [9]. Part of this signaling scheme has been reported by Baldine *et al.* [10]. The signaling protocol has been implemented in FPGA and it has been deployed in the ATDNet testbed in Washington, DC. In this paper, we model an edge OBS node employing

This work was supported by the Intelligence Technology Innovation Center under contract MDA904-00-C-2133.

the JumpStart JIT signaling protocol. We now describe the aspects of the JumpStart signaling protocol that are necessary for modeling an edge OBS node; for full details, the interested reader is referred to [10].

We consider an OBS network consisting of OBS nodes (switches) interconnected by bidirectional fiber links. Each fiber link between a user and an OBS edge node, or between two adjacent OBS nodes, can support $W + 1$ wavelengths. Of these, one wavelength (referred to as *control wavelength*) is used to transmit control packets, and the other W wavelengths (referred to as *burst wavelengths*) are used to transmit data bursts. A user is equipped with $W + 1$ pairs of optical transceivers, each fixed tuned to one of the $W + 1$ wavelengths. (Alternatively, a user may be equipped with fewer than $W + 1$ pairs of tunable or fixed transceivers; in this case, however, it is not possible for a user to access all wavelengths at the same time.)

Following the JumpStart JIT signaling protocol [10], a user first sends a *setup* message to its edge OBS node. The *setup* message includes the source and destination addresses, the wavelength on which the source prefers to transmit the burst, and other information. We assume that an OBS node consists of a non-blocking space-division switch fabric, with no optical buffers. If the edge node can switch the burst on the specified wavelength, it returns a *setup ack* message to the user. The *setup ack* message contains the *offset* field that informs the user how long it should wait before transmitting its burst. It is possible, however, that a *setup* message be refused if the preferred wavelength on the destination output port is busy, or in the case of full wavelength converters, if all the wavelengths on the destination output port are busy. In this case, the edge node returns a *reject* message. The user undergoes a random delay, and then re-transmits the *setup* message. In our model, we assume that the user continues to re-transmit the *setup* message until it receives a *setup ack* message, although this assumption can be easily removed. We have considered the case where a burst is dropped if its *setup* message is rejected in [16].

We assume that the node allocates resources within its switch fabric for a burst at the moment that it decides to accept the *setup* message. An alternative approach is to allocate the necessary resources near the time the edge node expects the burst to arrive. Assuming that the estimate regarding the burst arrival time is accurate, the latter approach minimizes the holding time of the resource for a burst. On the other hand, it also requires a complex scheduling algorithm, as well as significant amounts of memory on the signaling board that processes the OBS signals [10]. Therefore, in JumpStart we have decided to follow the former approach, which is the one we model in this work.

Another design issue is related to the time when the node frees the resources allocated to a burst. One way of resolving this problem is for the source to indicate the length of the transmission of the burst in the *setup* message. Assuming that the node knows when the burst will start to arrive, it can then calculate the time when it will free its resources.

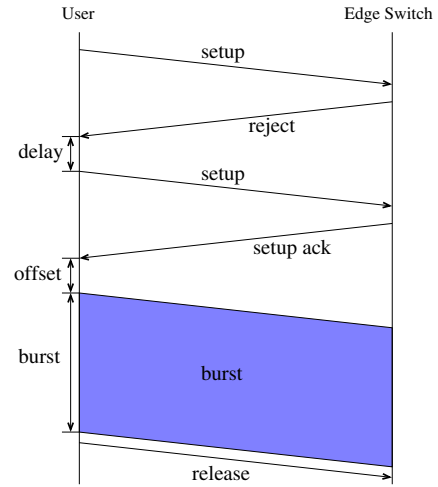


Fig. 1. Signaling messages in JumpStart

Alternatively, the user does not communicate to the edge node the length of its burst, but it simply sends a *release* message to the node to indicate the end of its transmission. Upon receipt of the *release* message, the node frees the resources allocated to the burst. The latter solution seems to be easier to implement, but it gives rise to a larger number of signaling messages. Our model can take into account either method, due to the inherent abstractions in the underlying queueing network.

The sequence of messages exchanged between a user and its edge node is shown in Figure 1. A user can be seen as being in one of the following three states: (1) *idle*, i.e., no bursts to transmit; (2) *busy* transmitting a burst; or (3) *blocked*, i.e., undergoing a delay before it re-transmits a *setup* message. If a user can simultaneously transmit bursts on different wavelengths, then it can be in a different state for each burst wavelength.

III. THE BURST ARRIVAL PROCESS

Each burst wavelength from a user to an OBS edge switch is associated with a burst arrival process. We use the three-state Markov process shown in Figure 2 to model arrivals on a given burst wavelength. The arrival process may be in one of three states: *short burst*, *long burst*, or *idle*. If it is in the *short burst* (respectively, *long burst*) state, then the user is in the process of transmitting a *short* (respectively, *long*) burst on this wavelength. If it is in the *idle* state, then the user is not transmitting any burst on this wavelength. The duration of a burst, whether short or long, and the time spent in the *idle* state, are assumed to be exponentially distributed. In this model, we assume that the source becomes *idle* after the transmission of each burst. That is, the source does not transmit bursts back-to-back. This assumption can be easily removed by modifying the three-state Markov process in Figure 2 to include transitions between the *short burst* and *long burst* states. Also, more complicated burst arrival processes can be modeled

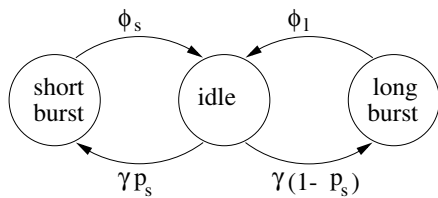


Fig. 2. The burst arrival process

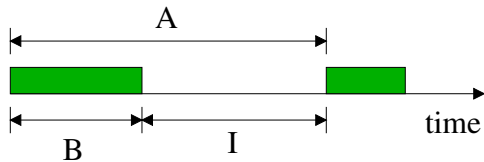


Fig. 3. Relationship between random variables A , B , and I

by introducing additional states and appropriate transitions between them. For instance, instead of using only two burst lengths (short and long), we may introduce $l > 2$ different burst lengths, each associated with a different state of the Markov process. Non-exponentially distributed burst lengths can also be accounted for by describing the length of a burst by a Coxian distribution. The analysis of the queueing network model that represents an edge OBS node, which we develop in the following sections, can be extended in a straightforward manner to these more general burst arrival processes. However, incorporating more general arrival processes in the model does introduce additional complexities in the expressions we derive, therefore, to keep the analysis simple we only consider the three-state Markov process in Figure 2.

The burst arrival process of Figure 2 is characterized completely by the following parameters:

- $1/\gamma$, the mean duration of the idle state,
- $1/\phi_s$ and $1/\phi_l$, the mean durations of the short burst and long burst states, respectively,
- p_s , the probability that a burst is a small burst, and

We also let p_i denote the probability that a burst from the user has output port i , $i = 1, \dots, P$, as its destination, where P is the number of output ports of the edge OBS node.

Let A , B , and I be random variables denoting the burst interarrival time, burst duration, and idle time, respectively. Their relationship is shown in Figure 3. Let $L_A(s)$, $L_B(s)$, and $L_I(s)$ denote their Laplace transform, respectively. We have that:

$$L_A(s) = L_B(s)L_I(s) = \left(p_s \frac{\phi_s}{\phi_s + s} + (1-p_s) \frac{\phi_l}{\phi_l + s} \right) \frac{\gamma}{\gamma + s} \quad (1)$$

By differentiating $L_A(s)$, we obtain the first two moments of the interarrival time A as follows:

$$E[A] = p_s \frac{1}{\phi_s} + (1-p_s) \frac{1}{\phi_l} + \frac{1}{\gamma} \quad (2)$$

$$E[A^2] = p_s \left(\frac{1}{\phi_s^2} + \frac{1}{\gamma\phi_s} + \frac{1}{\gamma^2} \right) + (1-p_s) \left(\frac{1}{\phi_l^2} + \frac{1}{\gamma\phi_l} + \frac{1}{\gamma^2} \right) \quad (3)$$

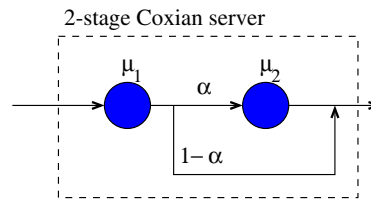


Fig. 4. 2-stage Coxian server

Then, the squared coefficient of variation of the inter-arrival time of successive bursts (short or long), $c^2(A)$, is given by:

$$c^2(A) = \frac{E(A^2)}{E^2(A)} - 1 \quad (4)$$

The squared coefficient of variation $c^2(A)$ is a measure of the burstiness of the arrival process. Unlike the Poisson process which is smooth ($c^2(A) = 1$), one may introduce any degree of burstiness into the arrival process of Figure 2 by appropriately selecting the parameters of the three-state Markov process.

We note that the inter-arrival times of successive bursts are i.i.d. It is possible to introduce correlation among the inter-arrival times by allowing bursts to arrive back-to-back, as we explained above. In this work we do not consider correlated inter-arrival times.

Since a customer may request either a short or a long service (burst) with probabilities p_s and $1-p_s$, respectively, the service time distribution is a two-stage hyperexponential distribution. It is well-known that this distribution is equivalent to a two-stage Coxian distribution with $c^2(B) \geq 1$ (see Figure 4). We will let μ_1 and μ_2 denote the service rate of the first and second stages of the corresponding Coxian server, respectively, and a denote the probability that, upon completion of the first service stage, the customer in the Coxian server will proceed to the second stage. The values of μ_1 , μ_2 , and a are uniquely determined by the values of $1/\phi_s$, $1/\phi_l$, and p_s as follows [11]:

$$\mu_1 = \phi_s, \quad \mu_2 = \phi_l, \quad a = \frac{(1-p_s)(\phi_s - \phi_l)}{\phi_s} \quad (5)$$

The mean $E(B)$ and the squared coefficient of variation $c^2(B)$ of the burst duration can be calculated as:

$$E(B) = \frac{p_s}{\phi_s} + \frac{1-p_s}{\phi_l} \quad (6)$$

$$c^2(B) = \frac{2}{E^2(B)} \left(\frac{p_s}{\phi_s^2} + \frac{1-p_s}{\phi_l^2} \right) - 1 \quad (7)$$

IV. A QUEUEING NETWORK MODEL OF AN EDGE OBS NODE

In general, an edge OBS node is connected to a number of users and to a number of other OBS nodes. Consequently, it receives bursts from users and other OBS nodes. In this work, we assume that there is no traffic from other OBS nodes to the edge OBS node, and we only consider the burst traffic

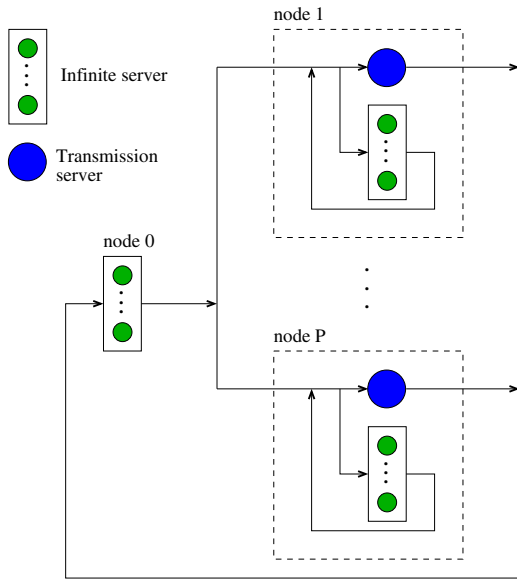


Fig. 5. Queueing network model of a sub-system of an edge switch without converters

from the users ¹. Let P and N denote the number of input (or output) ports of an edge node and the number of the users connected to the edge node, respectively. Note that, $P \geq N$.

The traffic on each incoming wavelength from a user to the edge node is generated by the burst arrival process described in Section III. Since each user can simultaneously transmit bursts on all its W burst wavelengths, the user is associated with W different burst arrival processes. Therefore, an edge node with N users has a total of NW burst arrival processes. Recall that a setup message is refused if, at the time it arrives at the edge node, the output port is busy transmitting another burst. In this case, the corresponding burst arrival process undergoes an exponential delay, and then the user re-transmits its setup message. Thus, at any time, there may be a number of burst arrival processes undergoing an exponential delay for each output wavelength.

A. Edge OBS Node Without Converters

Let us first consider an edge OBS node with no converters. In this case, a burst on an incoming wavelength can only be switched to the *same* wavelength on each output port, and user bursts arriving to the edge switch on different wavelengths do not interfere with each other. Consequently, the edge node can be decomposed into W sub-systems, one per burst wavelength. This decomposition is exact. Each sub-system $w, w = 1, \dots, W$, is a $P \times P$ switch with N users, but each input and output port has a single wavelength, which corresponds to wavelength w of the original edge switch. Therefore, each sub-system has N burst arrival processes.

¹This is a reasonable assumption for an edge OBS node. Most traffic from other OBS nodes to the edge node is in the direction *from* the OBS network *to* the users, while in this work we are interested in modeling the performance of an edge node in the direction *from* the users *to* the OBS network.

The queueing network model of a sub-system is shown in Figure 5; it consists of $P + 1$ nodes numbered $0, 1, \dots, P$. Node 0 is an infinite server node, and it represents the burst arrival processes which are in the *idle* state. Node $i, i = 1, \dots, P$, represents the (single) wavelength on output port i . Each node i consists of a single *transmission server* and an *infinite server*. The customer (if any) occupying the transmission server represents the burst arrival process whose burst is being transmitted by output port i . The customers (if any) in the infinite server represent those burst arrival processes which are undergoing a delay before their users re-transmit the corresponding setup messages. The total number of customers in this closed queueing network model of a sub-system is equal to N (i.e., it is equal to the total number of burst arrival processes in the sub-system).

Let us now follow the path of a customer through the queueing network model in Figure 5. Let us assume that the customer starts in the *idle* state, i.e., it is in node 0. The time it spends in the *idle* state is exponentially distributed with mean $1/\gamma$. Upon completion of its service at node 0, it moves to node i with probability p_i ; this corresponds to the transmission of a setup message for a burst with output port i . If the single transmission server at node i is free, the customer enters service immediately. The service time is exponentially distributed with a mean of $1/\phi_s$ or $1/\phi_l$ with probabilities p_s or $1 - p_s$, corresponding to the transmission of short or long burst, respectively. If the transmission server is busy (i.e., output port contention occurs), the customer enters the infinite server at node i , where it undergoes an exponential delay with mean $1/\omega$; this delay models the delay until the retransmission of the setup message. Upon completion of the exponential delay, the customer again tries to seize the transmission server. If the transmission server is busy, the customer joins the infinite server again, and it undergoes another delay, and so on, until it succeeds to get hold of the transmission server. The customers in the infinite server are often referred to in the literature as *orbiting* customers. Note that it is possible for the transmission server to become idle while there are one or more customers orbiting. In this case, it is possible that a new customer arrives from node 0 and starts service immediately.

In the case where all N customers have the same burst arrival process, the closed queueing network model can be seen as consisting of a single class of N customers and $P + 1$ nodes. If each customer has a different burst arrival process, then we have a multi-class queueing network with $P + 1$ nodes and N classes, where each class contains exactly one customer.

B. Edge OBS Node With Converters

Let us now consider an edge OBS switch with converters. In this case, a setup message for output port i of the switch is accepted as long as at least one wavelength of this output port is free. Otherwise, the setup message is rejected, and the user undergoes a delay before retransmitting the message. Clearly, the above decomposition of an edge switch into sub-systems per wavelength is no longer possible, since user bursts

TABLE I
NOTATION USED IN THE ANALYSIS

Parameter	Description
N	number of users connected to edge switch
P	number of input (output) ports of edge switch
W	number of burst wavelengths in a fiber
$1/\gamma$	mean duration of the idle state
$1/\phi_s$	mean duration of the short burst
$1/\phi_l$	mean duration of the long burst
$1/\omega$	mean orbiting time of a user
p_s	prob. that a burst is a short burst
p_i	prob. that destination output port of burst is i
$E(B)$	mean duration of a burst (short or long)
μ_1, μ_2, a	param. of 2-stage Coxian distr. of burst size

arriving on different wavelengths may interfere with each other. However, the edge switch *as a whole* can be modeled by a closed queueing network very similar to the one shown in Figure 5. The new queueing network consists of $P + 1$ nodes and a total of NW customers (since there are now NW arrival processes). Node 0 in the new queueing network is identical to node 0 in the network of Figure 5. Similarly, each node $i, i = 1, \dots, P$, in the new queueing network corresponds to each of the output ports of the edge switch. The main difference is that each node $i, i = 1, \dots, P$, consists of an infinite server and W (rather than one) transmission servers, each corresponding to one of the W wavelengths of output port i .

In the following sections, we describe a technique for solving the queueing network in Figure 5. We note that, despite the rich literature in queueing network analysis, this particular queueing network with orbiting customers has not been analyzed before. The notation used in the analysis is summarized in Table I. In Section V, we analyze the queueing network assuming a single class of customers (i.e., all burst arrival processes are identical) and no wavelength converters. In Section VI, we analyze the network assuming a single class of customer and wavelength converters. Finally, the analysis of the multi-class network is presented in Section VII.

V. ANALYSIS OF THE SINGLE-CLASS QUEUEING NETWORK WITHOUT CONVERTERS

The queueing network in Figure 5 is a non-product-form queueing network with Coxian service times. We analyze it using Marie's algorithm [12], [13]. The idea in Marie's method is to replace each non-BCMP node by a *flow equivalent node* with a load-dependent exponential service rate, obtained by calculating the conditional throughput of the non-BCMP node in isolation under a load-dependent arrival rate. Therefore, we need to construct a flow equivalent server for each node $i, i = 1, \dots, P$. Node 0 is an infinite server (a BCMP node), so we do not need to construct a flow equivalent node for it. To the best of our knowledge, Marie's method has not been applied to nodes with orbiting customers. Consequently, the derivation in the next subsection of a flow equivalent server for such a node is a new contribution.

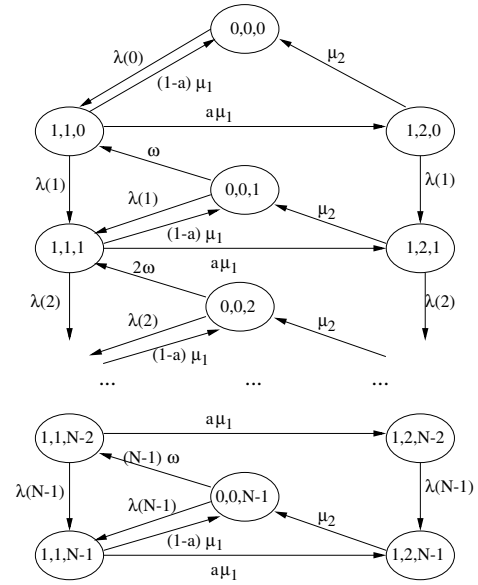


Fig. 6. State transition rate diagram of node $i, i = 1, \dots, P$, of the queueing network of Figure 5

A. The Flow Equivalent Server

Let us consider node $i, i = 1, \dots, P$, of the queueing network shown in Figure 5. Let $\lambda_i(n_i)$ be the arrival rate into this node when there are a total of n_i customers in the node. We also assume that the service time is a two-stage Coxian distribution with parameters $\mu_{(i,1)}, \mu_{(i,2)}$, and a_i . The state of node i can be described by the triplet: (n_i^t, k_i, n_i^o) , where $n_i^t = 0, 1$, indicates whether the transmission server is busy or not, $k_i = 0, 1, 2$, is the Coxian phase of the transmission server ($k_i = 0$ if and only if $n_i^t = 0$), and $n_i^o = 0, 1, \dots, N - 1$, gives the number of orbiting customers occupying the infinite server. The state transition diagram of node i is shown in Figure 6. In order to simplify the notation, and since we are only concerned with the analysis of node i in isolation, we drop the index i in Figure 6 and throughout the rest of this subsection.

Let $p(n^t, k, n^o)$ be the steady-state probability of the state (n^t, k, n^o) . From Figure 6, we have the following global balance equations:

$$p(0, 0, n)(\lambda(n) + n\omega) = p(1, 1, n)(1 - a)\mu_1 + p(1, 2, n)\mu_2, \quad 0 \leq n < N \quad (8)$$

$$p(1, 1, n - 1)(\lambda(n) + \mu_1) = p(0, 0, n)n\omega + p(1, 1, n - 2) \times \lambda(n - 1) + p(0, 0, n - 1)\lambda(n - 1), \quad 0 < n \leq N \quad (9)$$

$$p(1, 2, n - 1)(\lambda(n) + \mu_2) = p(1, 1, n - 1)a\mu_1 + p(1, 2, n - 2)\lambda(n - 1), \quad 0 < n \leq N \quad (10)$$

Let $p(n)$ denote the steady-state probability that there are a total of n customers in the node. We have that:

$$p(n) = \begin{cases} p(0, 0, 0), & n = 0 \\ p(0, 0, n) + p(1, 1, n - 1) + p(1, 2, n - 1), & 0 < n < N \\ p(1, 1, N - 1) + p(1, 2, N - 1), & n = N \end{cases} \quad (11)$$

Let $v(n)$ denote the conditional throughput of the node, calculated as follows:

$$v(n) = \begin{cases} 0, & n = 0 \\ \frac{p(1,1,n-1)}{p(n)}(1-a)\mu_1 + \frac{p(1,2,n-1)}{p(n)}\mu_2, & n > 0 \end{cases} \quad (12)$$

We now have the following two theorems:

Theorem 1: $p(n-1)\lambda(n-1) = p(n)v(n)$, $0 < n \leq N$.

Proof: By adding equations (8), (9), and (10) together, and after simplifying the result using equations (11) and (12), we obtain:

$$p(0)\lambda(0) = p(1)v(1), \quad n = 0 \quad (13)$$

$$p(n)\lambda(n) + p(n)v(n) = p(n-1)\lambda(n-1) + p(n+1)v(n+1), \quad 0 < n < N \quad (14)$$

$$p(N)v(N) = p(N-1)\lambda(N-1), \quad n = N \quad (15)$$

Using equations (13) and (14) recursively, we finally get:

$$p(n-1)\lambda(n-1) = p(n)v(n), \quad 0 < n \leq N \quad (16)$$

completing the proof of the theorem. \blacksquare

Theorem 2: The conditional throughput $v(n)$ of the node is given by the expressions:

$$\begin{aligned} v(0) &= 0 \\ v(1) &= \frac{\mu_1\omega(\lambda(1)-a\lambda(1)+\mu_2)}{(\lambda(1)+\omega)(\lambda(1)+a\mu_1+\mu_2)} \\ v(n) &= \frac{n\mu_1\omega(\lambda(n-1)+(n-1)\omega)(\lambda(n)-a\lambda(n)+\mu_2)}{Z}, \quad 1 < n \end{aligned} \quad (17)$$

where $Z = (\lambda(n) + n\omega)((n-1)\omega(\mu_1 + \mu_2 + \lambda(n) - v(n-1)) + \lambda(n-1)(a\mu_1 + \mu_2 + \lambda(n) - v(n-1)))$.

Proof: By means of expression (12), we can rewrite expression (8) as follows:

$$\begin{aligned} p(0,0,n)(\lambda(n) + n\omega) &= p(1,1,n)(1-a)\mu_1 + p(1,2,n)\mu_2 \\ &= p(n+1)v(n+1) \\ &= p(n)\lambda(n) \end{aligned} \quad (18)$$

We can also rewrite expression (9) as:

$$\begin{aligned} &p(1,1,n-1)(\lambda(n) + \mu_1) \\ &= p(0,0,n)n\omega + p(1,1,n-2)\lambda(n-1) + p(0,0,n-1)\lambda(n-1) \\ &= p(0,0,n)n\omega + \frac{p(1,1,n-2) + p(0,0,n-1)}{p(n-1)}p(n-1)\lambda(n-1) \\ &= p(0,0,n)n\omega + \frac{p(1,1,n-2) + p(0,0,n-1)}{p(n-1)}p(n)v(n) \end{aligned} \quad (19)$$

Using expressions (11), (12), (18), and (19), we get the following group of equations for $n > 1$:

$$\begin{cases} p(n) = p(0,0,n) + p(1,1,n-1) + p(1,2,n-1) \\ p(n-1) = p(0,0,n-1) + p(1,1,n-2) + p(1,2,n-2) \\ v(n) = \frac{p(1,1,n-1)}{p(n)}(1-a)\mu_1 + \frac{p(1,2,n-1)}{p(n)}\mu_2 \\ v(n-1) = \frac{p(1,1,n-2)}{p(n-1)}(1-a)\mu_1 + \frac{p(1,2,n-2)}{p(n-1)}\mu_2 \\ p(0,0,n)(\lambda(n) + n\omega) = p(n)\lambda(n) \\ p(0,0,n-1)(\lambda(n-1) + (n-1)\omega) = p(n-1)\lambda(n-1) \\ p(1,1,n-1)(\lambda(n) + \mu_1) = p(0,0,n)n\omega \\ \quad + \frac{p(1,1,n-2) + p(0,0,n-1)}{p(n-1)}p(n)v(n) \end{cases} \quad (20)$$

Now, assuming that $p(0,0,n)$, $p(0,0,n-1)$, $p(1,1,n-1)$, $p(1,1,n-2)$, $p(1,2,n-1)$, $p(1,2,n-2)$, and $v(n)$ are unknown variables, we can solve the group of equations (20) to obtain the expression for $v(n)$, $n > 1$, as in (17).

Similarly, we can solve the following group of equations for $p(0,0,1)$, $p(1,1,0)$, $p(1,2,0)$, and $v(1)$:

$$\begin{cases} p(1) = p(0,0,1) + p(1,1,0) + p(1,2,0) \\ v(1) = \frac{p(1,1,0)}{p(1)}(1-a)\mu_1 + \frac{p(1,2,0)}{p(1)}\mu_2 \\ p(0,0,1)(\lambda(1) + \omega) = p(1)\lambda(1) \\ p(1,1,0)(\lambda(1) + \mu_1) = p(0,0,1)\omega + p(1)v(1) \end{cases} \quad (21)$$

to obtain the expression for $v(1)$ as in (17). Since $v(0)$ is obviously equal to 0, the proof is complete. \blacksquare

We use the conditional throughput $v(n)$ as the load-dependent service rate $\mu(n)$ of the node in the iterative algorithm described in the next subsection.

B. The Iterative Algorithm

We use Marie's algorithm [12] to analyze the queueing network of Figure 5. The main steps of the algorithm:

- **Step 1.** Initialize the service rate $\mu_i(n_i)$ of flow equivalent server i , $i = 1, \dots, P$, to $1/E(B)$, for $n_i > 0$, and set the service rate $\mu_0(n_0)$ of flow equivalent server 0 to γn_0 .
- **Step 2.** For each node i , $i = 1, \dots, P$, do:
 - **Step 2.1.** Calculate the arrival rate $\lambda_i(n_i)$ of node i by short-circuiting node i in the substitute product-form closed queueing network, where each node j has an exponential service time of $\mu_j(n_j)$.
 - **Step 2.2.** Calculate the conditional throughput $v_i(n_i)$ of node i using Theorem 2.
 - **Step 2.3.** Calculate the steady-state probability $p_i(n_i)$ of node i using Theorem 1.
- **Step 3.** Check the following two convergence conditions. If both are satisfied, then stop. Otherwise, set $\mu_i(n_i)$ to $v_i(n_i)$ for all $i = 1, \dots, P$, and go back to Step 2.

- 1) The first convergence condition ensures that the sum of the mean number of customers at all nodes is equal to the number of customers in the queueing network:

$$\left| \frac{N - \sum_{i=0}^P \sum_{j=0}^N j p_i(j)}{N} \right| < \epsilon \quad (22)$$

- 2) The second convergence condition makes sure that the conditional throughputs of each node are consistent with the topology of the queueing network:

$$\left| \frac{r_i - \frac{1}{P+1} \sum_{j=0}^P r_j}{\frac{1}{P+1} \sum_{j=0}^P r_j} \right| < \epsilon, \quad i = 0, 1, \dots, P \quad (23)$$

$$r_i = \begin{cases} \frac{1}{p_i} \sum_{j=0}^N p_i(j) \mu_i(j), & i = 1, \dots, P \\ \sum_{j=0}^N p_i(j) \mu_i(j), & i = 0 \end{cases} \quad (24)$$

VI. ANALYSIS OF THE SINGLE-CLASS QUEUEING NETWORK WITH CONVERTERS

As we discussed in Section IV, the only difference between the queueing network model of an edge switch with wavelength converters and the queueing network of Figure 5 (which models a single wavelength sub-system of an edge switch without wavelength converters) is that, in the former model, each node $i, i = 1, \dots, P$, has W transmission servers, while in the latter model each node i has a single transmission server. Unfortunately, when each node i has multiple transmission servers, we cannot obtain a closed-form solution for the conditional throughput of the node. Instead, we solve each node i numerically using the Gauss-Seidel method [11], to get $p_i(n_i)$, the steady probability that node i has n_i customers. Then, we calculate the conditional throughput $v_i(n_i)$ as follows:

$$v_i(n_i) = \frac{p_i(n_i - 1)\lambda_i(n_i - 1)}{p_i(n_i)} \quad (25)$$

Finally, we use the same iterative algorithm described in Section V-B, to analyze this more general queueing network.

VII. ANALYSIS OF THE MULTI-CLASS QUEUEING NETWORK WITH OR WITHOUT CONVERTERS

In this section, we present a new decomposition technique for solving the multi-class version of the queueing network in Figure 5. The technique is quite general and it can be applied to a wide class of multi-class queueing networks. The notation used in the analysis is summarized in Table I.

Consider an edge OBS switch where each customer has a different burst arrival process. This feature is taken into account by associating each customer with a different class. The resulting queueing network is a closed non-product-form queueing network with multiple classes, each of which has only a single customer. The number of classes is $C = NW$, where N is the number of users connected to the edge OBS node and W is the number of wavelengths per fiber. We note that, for realistic values of N and W , the number of classes can be very large (i.e., in the order of 100s).

This type of multi-class closed non-product-form has been studied in the literature [14], [15], mainly by extending Marie's algorithm [12]. Neuse and Chandy [14] proposed an algorithm called the *heuristic aggregation method* (HAM) to solve such a queueing network. HAM is a natural extension of Marie's method, but it involves two time-consuming computations which limit its applicability to networks with a very small number of classes only. First, it requires the numerical analysis of a node with multi-class load-dependent arrivals and a two-stage Coxian service time. Second, it requires the computation of the normalization constant in a multi-class queueing network. Baynat and Dallery [15], presented an alternative extension of Marie's method to multi-class queueing networks. Specifically, to avoid the computation of the normalization constant of a multi-class network, they decompose a C -class network into C single-class networks. The interaction of the customers in different classes is taken into account in the analysis of each node in isolation. They also proposed a class

aggregation technique that reduces significantly the complexity of the analysis of a node. However, the arrival rate to the aggregate class is calculated by aggregating the arrival rate to each individual class [15]. This aggregation process takes time that increases exponentially with the number of classes. Consequently, while this method is faster than HAM, it cannot be used in networks with a large number of classes.

Based on these observations, we first show in Section VII-A how to use HAM to solve networks with only two classes, and in Section VII-B we present a new method for solving networks with more than two classes of customers. Specifically, we decompose a network with multiple classes of customers into a set of two-class networks, each of which is solved using HAM. We also employ a class aggregation technique to reduce the complexity of the analysis of a node. However, we use the convolution algorithm to calculate the arrival rate to the aggregate class and not Baynat and Dallery's method [15] since, as we mentioned above, the latter is not scalable.

A. The Two-Class Queueing Network

In this subsection, we assume that there are only two classes of customers in the queueing network of Figure 5, namely, class 0 and 1. Also, we assume that each node $i, i = 1, \dots, P$, of the queueing network consists of $W \geq 1$ transmission servers. Therefore, the analysis applies to edge OBS switches with converters ($W > 1$) or without ($W = 1$).

We first construct a flow equivalent node for each node $i, i = 1, \dots, P$, of the queueing network in Figure 5. Let $\lambda_{i0}(n_{i0}, n_{i1})$ and $\lambda_{i1}(n_{i0}, n_{i1})$ denote the arrival rate of class 0 and class 1 customers, respectively, to node i when there are n_{i0} class 0 customers and n_{i1} class 1 customers in the node. We also assume that the service time of class $j, j = 0, 1$, is a two-stage Coxian distribution with parameters $\mu_{(ij,1)}, \mu_{(ij,2)}$, and a_{ij} . The state of node i can be described by the vector $(n_{i0}^{(t1)}, n_{i0}^{(t2)}, n_{i0}^o, n_{i1}^{(t1)}, n_{i1}^{(t2)}, n_{i1}^o)$ where

- $n_{i0}^{(t1)}$ and $n_{i1}^{(t1)}$ are random variables representing the number of class 0 and class 1 customers, respectively, being served by the transmission servers in phase one,
- $n_{i0}^{(t2)}$ and $n_{i1}^{(t2)}$ are random variables representing the number of class 0 and class 1 customers, respectively, being served in phase two, and
- random variables n_{i0}^o and n_{i1}^o represent the number of orbiting customers of class 0 and 1, respectively.

Let $p_i(n_{i0}^{(t1)}, n_{i0}^{(t2)}, n_{i0}^o, n_{i1}^{(t1)}, n_{i1}^{(t2)}, n_{i1}^o)$ be the steady state probability that node i is in state $(n_{i0}^{(t1)}, n_{i0}^{(t2)}, n_{i0}^o, n_{i1}^{(t1)}, n_{i1}^{(t2)}, n_{i1}^o)$. We use the Gauss-Seidel method to calculate the steady-state probability numerically. We then obtain the conditional throughput $v_i(n_i)$ of node i as:

$$v_i(n_i) = \frac{p_i(n_i - 1)\lambda_i(n_i - 1)}{p_i(n_i)} \quad (26)$$

where $p_i(n_i)$ and $\lambda_i(n_i)$ are calculated by:

$$p_i(n_i) = \sum_{n_{i0}^{(t1)} + n_{i0}^{(t2)} + n_{i0}^o + n_{i1}^{(t1)} + n_{i1}^{(t2)} + n_{i1}^o = n_i} p_i(n_{i0}^{(t1)}, n_{i0}^{(t2)}, n_{i0}^o, n_{i1}^{(t1)}, n_{i1}^{(t2)}, n_{i1}^o) \quad (27)$$

$$\lambda_i(n_i) = \frac{1}{p_i(n_i)} \sum_{n_{i0}+n_{i1}=n_i} p_i(n_{i0}^{(t1)}, n_{i0}^{(t2)}, n_{i0}^o, n_{i1}^{(t1)}, n_{i1}^{(t2)}, n_{i1}^o) \times [\lambda_{i0}(n_{i0}, n_{i1}) + \lambda_{i1}(n_{i0}, n_{i1})] \quad (28)$$

where $n_{i0} = n_{i0}^{(t1)} + n_{i0}^{(t2)} + n_{i0}^o$, and $n_{i1} = n_{i1}^{(t1)} + n_{i1}^{(t2)} + n_{i1}^o$. We use $v_i(n_i)$ as the load-dependent service rate $\mu_i(n_i)$ of the flow equivalent server of node i .

We solve this two-class product-form network consisting of the flow equivalent servers using the convolution algorithm to obtain the arrival rates $\lambda_{i0}(n_{i0}, n_{i1})$ and $\lambda_{i1}(n_{i0}, n_{i1})$ to each node $i, i = 1, \dots, P$. This process is repeated until convergence, as dictated by Marie's algorithm [12].

B. Iterative Algorithm for Analyzing More Than Two Classes

As we observed above, the complexity of HAM increases exponentially with the number of classes, thus it can only be applied to networks with a small number of classes. We now introduce a new method for solving queueing networks with a large number of classes. The main idea of our algorithm is to approximate the original multi-class network with a set of two-class networks, each of which is solved using HAM. Below, we first describe a mechanism for aggregating a number of classes into a single class, and subsequently we describe an iterative algorithm for analyzing multi-class queueing networks.

1) *Class Aggregation*: Let C denote the number of classes in a network, $C = NW$. For a network with C classes, we create C two-class networks. For each two-class network $c, c = 1, \dots, C$, the first class is class c in the original network, and the second class is the aggregate class of all the other classes. To apply our algorithm, we need to specify the parameters of the service rate of the aggregate class $\mu_{i,1}^{(agg)}$, $\mu_{i,2}^{(agg)}$, and $\alpha_i^{(agg)}$ at node $i, i = 0, 1, \dots, P$.

Let e_{ic} denote the visit ratio of class c at node i . Assuming that we know the mean response time T_{ic} of class c at node i , we can calculate the throughput H_c of class c in the network and the throughput H_{ic} of class c at node i as follows:

$$H_k = \frac{N_k}{\sum_i T_{ik} \times e_{ik}}, \quad H_{ik} = H_k \times e_{ik} \quad (29)$$

where $N_c (= 1 \forall c)$ is the number of class c customers.

We employ the class aggregation technique in Baynat and Dallery's method [15] to obtain the parameters of the service time distribution of the aggregate class at node i . Since the distribution of each class is a two-stage hyper-exponential distribution, the distribution of the aggregate class is also a hyper-exponential distribution, but with more than two stages. The first two moments $E(Agg)$ and $E(Agg^2)$ of the distribution of the aggregate class can be calculated as follows:

$$E(Agg) = \frac{\sum_{k \neq c} H_{ik} E(B_k)}{\sum_{k \neq c} H_{ik}} \quad (30)$$

$$E(Agg^2) = \frac{\sum_{k \neq c} H_{ik} E^2(B_k)(1 + c^2(B_k))}{\sum_{k \neq c} H_{ik}} \quad (31)$$

$$c^2(Agg) = \frac{E(Agg^2) - E^2(Agg)}{E^2(Agg)} \quad (32)$$

where $E(B_k)$ and $c^2(B_k)$ are the mean and the squared coefficient of variation of the burst duration of class k , respectively. $E(B_k)$ and $c^2(B_k)$ can be calculated as in expressions (6) and (7). We approximate the service time distribution of the aggregate class as a two-stage Coxian distribution using moment matching. We use the expressions in [13] to match the first two moments, as follows:

$$\alpha_i^{(agg)} = \frac{1}{2c^2(Agg)}, \mu_{i1}^{(agg)} = \frac{2}{E(Agg)}, \mu_{i2}^{(agg)} = \mu_{i1}^{(agg)} \alpha_i^{(agg)} \quad (33)$$

2) *The Iterative Algorithm*: As we described in the previous subsection, if we know the mean response time of each class at each node in a network with C classes, then we can decompose the network into C two-class queueing networks. Using HAM, we can solve each of these C two-class queueing networks, and then we can re-calculate the mean response time of each class at each node. We repeat this process until it converges. The following steps summarize our iterative algorithm.

- **Step 1.** Initialize the mean response time T_{ic} to 1 for all $i, c, i = 0, 1, \dots, P, c = 1, \dots, C$. Initialize the load-dependent service rate $\mu_i(n_i)$ of node $i, i = 1, \dots, P$, in each two-class network to $\frac{\sum_i H_{ic}}{\sum_i H_{ic} E(B_c)}$ for $n_i > 0$.
- **Step 2.** For each $i = 0, 1, \dots, P$ and $c = 1, \dots, C$, do
 - **Step 2.1.** Calculate the throughput of each class using expression (29).
 - **Step 2.2.** Aggregate all classes except class c into one class using expression (33), to obtain the two-class queueing network c , which consists of class c and the aggregate class.
 - **Step 2.3.** Solve the two-class product-form queueing network c using the convolution algorithm to obtain the arrival rate to node i of both class c and the aggregate class.
 - **Step 2.4.** Solve node i numerically using the Gauss-Seidel method to obtain the steady-state probabilities $p_i(n_{i0}^{(t1)}, n_{i0}^{(t2)}, n_{i0}^o, n_{i1}^{(t1)}, n_{i1}^{(t2)}, n_{i1}^o)$.
 - **Step 2.5.** Calculate the conditional throughput $v_i(n_i)$ of node i using expression (26), and use this value as the load-dependent service rate $\mu_i(n_i)$ of the flow equivalent server of node i in the two-class network c .
 - **Step 2.6.** Set variable T_{ic}^{old} to T_{ic} .
 - **Step 2.7.** Calculate the *new* mean response time T_{ic} of class c at node i .
- **Step 3.** Check if the mean response times $T_{ic}, i = 0, \dots, P, c = 1, \dots, C$, satisfy the following convergence criterion. If so, then stop. Otherwise, repeat from Step 2.

$$\frac{\sqrt{\sum_{i,c} (T_{ic} - T_{ic}^{old})^2}}{\sqrt{\sum_{i,c} T_{ic}^2}} < \epsilon \quad (34)$$

VIII. NUMERICAL RESULTS

We now present results to illustrate how the different parameters affect the performance of the edge OBS node.

We also compare the approximate results to results obtained from simulation. Simulation results are plotted along with 95% confidence intervals; however, most confidence intervals are very narrow and are barely visible in the figures.

A. Single-Class Network

A comprehensive set of results for various performance measures and for a wide range of values of the system parameters listed in Table I can be found in [16]. Due to space constraints, we only show results for $P = 16$ (i.e., a 16×16 edge switch) with $W = 32$ wavelengths per fiber and a hot-spot traffic pattern where 10% of all arriving traffic has output port 16 as its destination, while the remaining traffic is uniformly distributed among the other 15 output ports (i.e., $p_{i,16} = 0.1 \forall i$, and $p_{ij} = 0.06 \forall i, j \neq 16$). We consider three performance measures: *switch throughput*, which is the sum over all output ports of the port throughput, *switch utilization*, i.e., the average across all output ports of the port utilization, and *mean waiting time of a user*, that is, the average waiting time until a user transmits a burst to the switch.

Figures 7-9 plot the three performance measures, respectively, against the number N of users attached to an OBS switch *without* converters (note that, as the number of users increases, the traffic load to the switch also increases accordingly). There are two sets of plots, each corresponding to a different burst arrival process. For both arrival processes, the mean burst size is 1, the mean burst interarrival time is 1.2, and the mean orbiting time is 1. However, the squared coefficient of variation $c^2(A)$ of the burst interarrival times is 1 for one process, and 100 for the other; the process with $c^2(A) = 1$ is very smooth, while the one with $c^2(A) = 100$ is extremely bursty. Each set consists of two plots, one corresponding to simulation results and one corresponding to results obtained using the approximate analytical model we developed in Section V.

From the three figures, we observe that as the number of users increases, the switch throughput, switch utilization, and mean user waiting time all increase. This behavior is expected, since the traffic load also increases with the number of users. We also observe that, for all three performance measures, the approximate analytical results are very close to the results obtained from simulation. (We have observed a good agreement between between the analytical and simulation results across a wide range of system parameters [16]). We emphasize that, in addition to being accurate, our approximate algorithms are orders of magnitude faster than simulation: computing one point for the plots of Figures 7-9 may take hours of simulation time, while it takes the approximate algorithm only a few seconds for the same computation.

From Figures 7-9 we also see the dramatic effect that the burstiness of the arrival process can have on the performance of the edge OBS node. Specifically, for the smooth arrival process ($c^2(A) = 1$), the switch throughput and utilization increase with the number of users, while the mean waiting time remains low. When the arrival process is extremely bursty ($c^2(A) = 100$), on the other hand, increasing the traffic

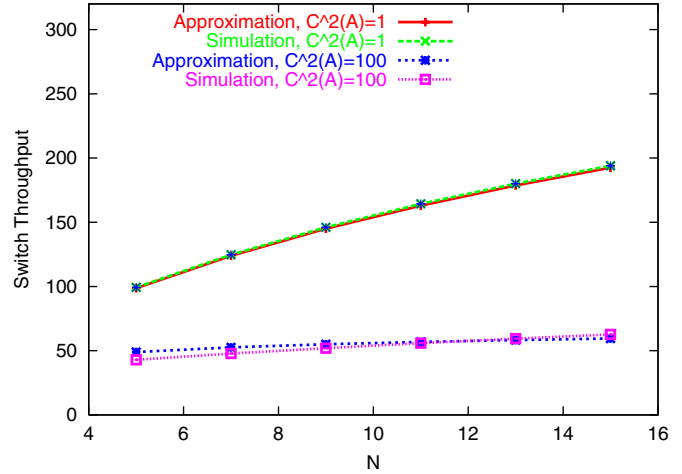


Fig. 7. Switch throughput, no converters

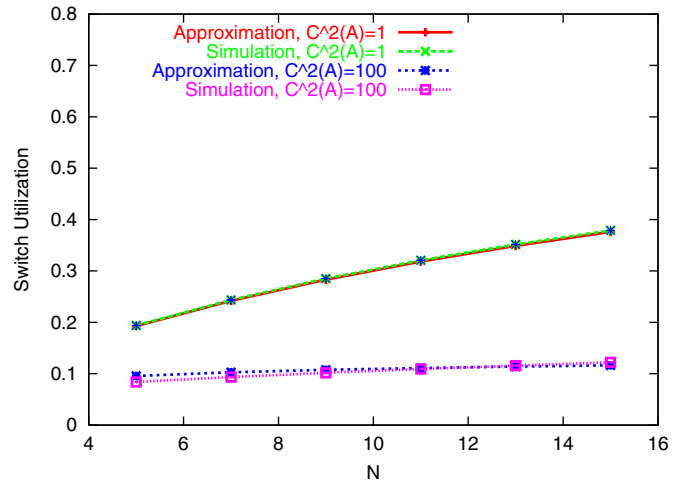


Fig. 8. Switch utilization, no converters

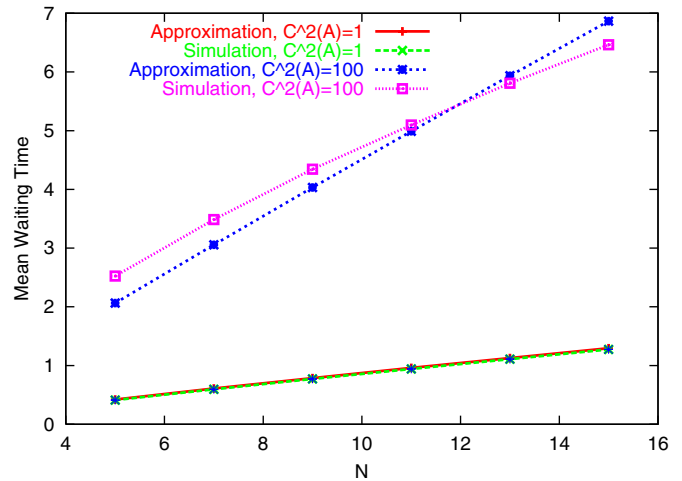


Fig. 9. Mean waiting time, no converters

load by increasing the number of users has minimal effect on switch throughput or utilization, which remain at low levels, while it severely affects the mean waiting time. This result can be explained by noting that, when arrivals are bursty, the first arriving user will be successful in transmitting its traffic, while subsequent users will find their output wavelength busy and will become orbiting customers. While this behavior agrees with intuition, the important point is that using our model, one can study a broad spectrum of switch behaviors by simply using different values for some of the parameters. Coupled with the fact that the approximation algorithms are fast and accurate, our queuing model can be used for extensive “what if” analysis that would not be possible otherwise.

Figures 10-12 are similar, but present results for an edge OBS node *with* converters. We consider two arrival processes as before, with the same mean burst size and mean interarrival time, and squared coefficient of variation equal to 1 or 100. From the figures, we observe the same good agreement between analytical and simulation results. However, we also observe two important differences compared to the results in Figures 7-9 for a switch with no converters. First, for the same traffic load, all performance measures are significantly improved. For instance, when $N = 15$, the switch throughput and utilization are 320 and 0.62, respectively (compared to 190 and 0.37, respectively, when there are no converters and $c^2(A) = 1$), while the mean waiting time drops dramatically to 0.32. Again, this improvement in performance is expected, but our queuing models enable us to quantify the benefits of wavelength conversion. The second important observation is that, for all three measures, there is little difference in the performance when the squared coefficient of variation of the arrival process increases from 1 to 100. This result can be explained by noting that each output port has $W = 32$ wavelengths, therefore, up to 32 simultaneous burst transmissions may take place at any given time *regardless* of the wavelength at which they arrive. Even when arrivals are extremely bursty, all transmissions can be accommodated as long as they are no more than 32. Only when all 32 wavelengths are busy will any transmission attempts undergo a delay; consequently, the switch throughput and utilization are high and waiting time is low. On the other hand, for a switch without converters, if multiple bursts arrive back-to-back requesting the same wavelength on the same output port, only the first one will be transmitted and all the others will have to be delayed. Therefore, the higher the degree of burstiness of the arrival process, the larger the degree to which the switch performance suffers. Overall, the results in Figures 10-12 indicate that, in addition to their well-known benefits, wavelength converters may also mitigate the adverse effects of extremely bursty traffic on switch performance. This observation is important since, by definition, OBS networks will deal with bursty traffic.

B. Multi-Class Network

We now present results for the edge switch when each user is associated with a different arrival process. We study the same 16×16 switch with $W = 32$ wavelengths and

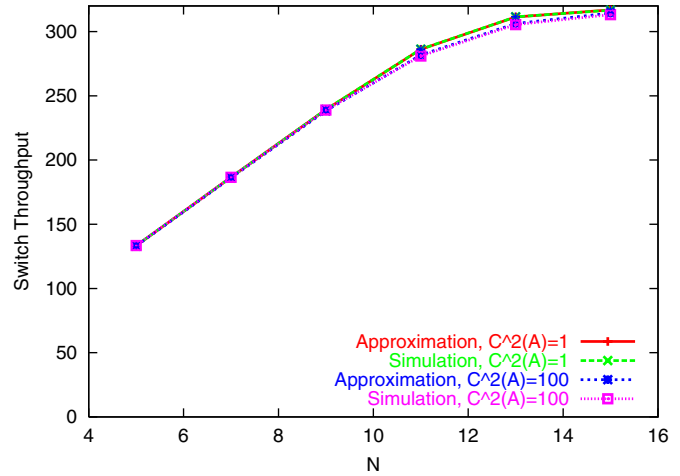


Fig. 10. Switch throughput, with converters

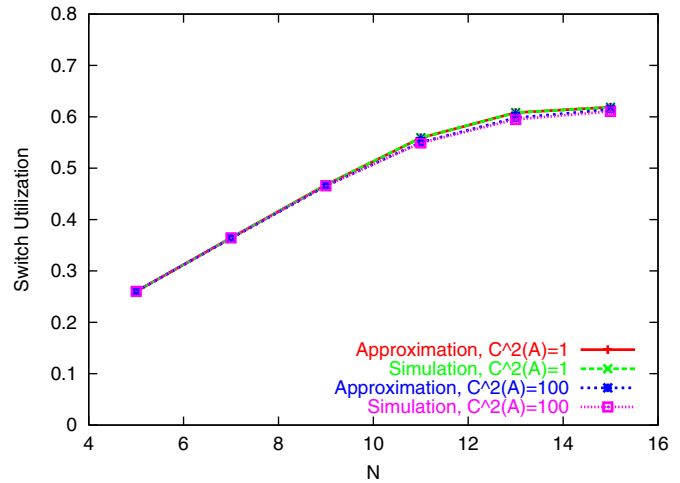


Fig. 11. Switch utilization, with converters

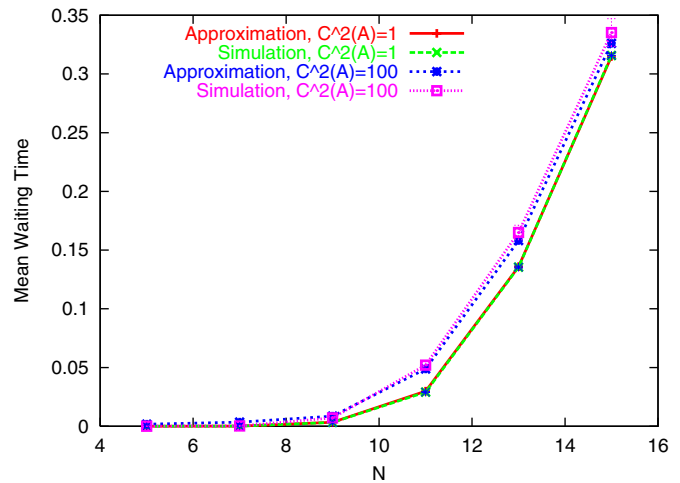


Fig. 12. Mean waiting time, with converters

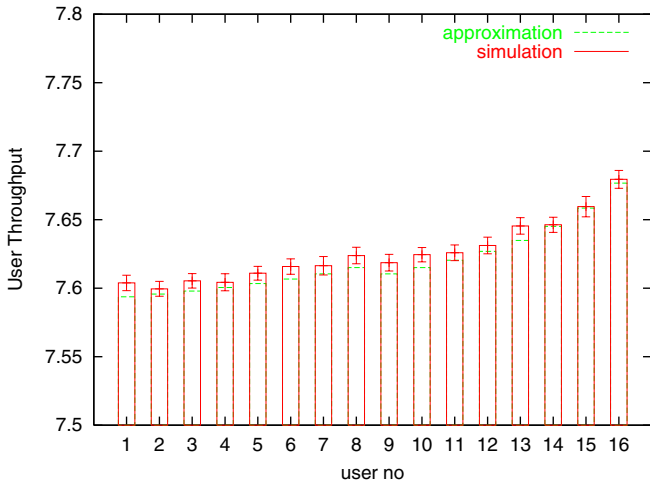


Fig. 13. User throughput, without converters

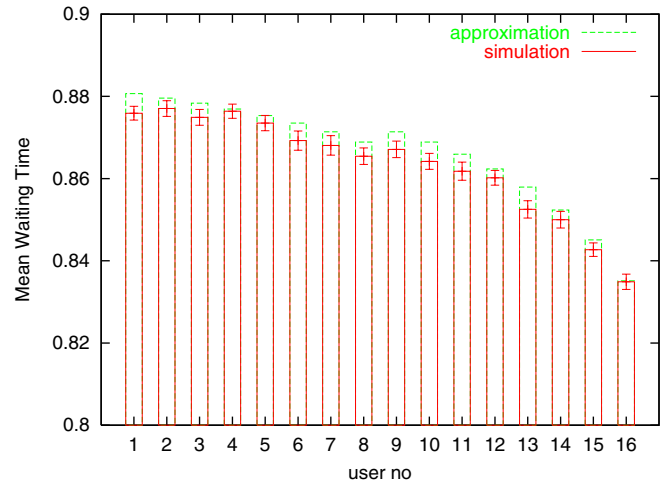


Fig. 14. User mean waiting time, without converters

the same hot-spot traffic pattern as in the last subsection. Since we are interested in the relative performance of different users, for simplicity we only show results for a switch without converters. In this experiment, all users have the same load 30%, the same mean burst size 1, the same ratio 100 of mean long burst size to mean short burst size, and mean orbiting time equal to 1. However, different users have different short burst probabilities p_s ; specifically, the value of p_s of user i , $i = 1, \dots, 16$, is equal to $0.05 + i \times 0.05$. That is, a user with a larger index has more short bursts than a user with a smaller index. For example, user 16 has 85% short bursts, while user 1 has only 10% short bursts. We consider two measures: *user throughput*, defined as the number of bursts transmitted by a user per unit time, and *mean user waiting time*.

Figures 13 and 14 plot the throughput and the mean waiting time of each user, respectively. First, we observe a good agreement between the analytical and simulation results. We also observe that as the index of a user increases, the throughput increases and the waiting time decreases. That is, if two users have the same traffic load, the same mean burst size, and the same burst size ratio, but different short burst probabilities, then the user with more short bursts has a lower waiting time and a higher throughput. Again, our model and approximate algorithms provide insight into the effect of the traffic parameters on user performance. For additional experiments and results for a multi-class OBS node, the reader is referred to [16].

IX. CONCLUDING REMARKS

We have presented a new queueing network model of an edge OBS node. The model is quite general, and it permits us to study the performance of an edge switch under a wide range of traffic and operational scenarios. We have developed approximate algorithms, and we have presented numerical results which demonstrate the accuracy of our approximations. We are currently working on extensions of the queueing network models to analyze a network of OBS nodes.

REFERENCES

- [1] 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.
- [2] K. Dolzer and C. Gauger, "On burst assembly in optical burst switching networks - a performance evaluation of Just-Enough-Time," in *Proceedings of the 17th International Teletraffic Congress*, September 2001, pp. 149–161.
- [3] K. Dolzer, C. Gauger, J. Späth, and S. Bodamer, "Evaluation of reservation mechanisms for optical burst switching," *AEÜ International Journal of Electronics and Communications*, vol. 55, no. 1, January 2001.
- [4] S. Verma, H. Chaskar, and R. Ravikanth, "Optical burst switching: a viable solution for terabit IP backbone," *IEEE Network*, pp. 48–53, November/December 2000.
- [5] J. S. Turner, "Terabit burst switching," *Journal of High Speed Networks*, vol. 8, no. 1, pp. 3–16, January 1999.
- [6] M. Yoo, C. Qiao, and S. Dixit, "QoS performance of optical burst switching in IP-over-WDM networks," *Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 2062–2071, October 2000.
- [7] J. Y. Wei, J. L. Pastor, R. S. Ramamurthy, and Y. Tsai, "Just-in-time optical burst switching for multiwavelength networks," in *IFIP TC6 WG6.2 Fifth International Conference on Broadband Communications*. Kluwer Academic Publishers, November 1999, pp. 339–352.
- [8] "The JumpStart project." <http://jumpstart.anr.mcn.org>.
- [9] J. Y. Wei and R. I. McFarland, "Just-in-time signaling for WDM optical burst switching networks," *Journal of Lightwave Technology*, vol. 18, no. 12, pp. 2019–2037, December 2000.
- [10] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson, "JumpStart: A just-in-time signaling architecture for WDM burst-switched networks," *IEEE Communications*, vol. 40, no. 2, pp. 82–89, February 2002.
- [11] H. Perros, *Queueing Networks with Blocking: Exact and Approximate Solutions*. Oxford University Press, 1994.
- [12] R. Marie, "An approximate analytical method for general queueing networks," *IEEE Transactions on Software Engineering*, vol. 5, no. 5, pp. 530–538, September 1979.
- [13] —, "Calculation equilibrium probabilities for $\lambda(n)/C_k/1/N$ queues," *ACM Sigmetrics Performance Evaluation Review*, vol. 9, no. 2, pp. 117–125, 1980.
- [14] D. Neuse and K. M. Chandy, "HAM: The heuristic aggregation method for solving general closed queueing network models of computer systems," *Performance Evaluation Review*, vol. 11, pp. 195–212, 1982.
- [15] B. Baynat and Y. Dallery, "A product-form approximation method for general closed queueing networks with several classes of customers," *Performance Evaluation*, vol. 24, pp. 165–188, 1993.
- [16] L. Xu, "Performance analysis of optical burst switched networks," Ph.D. dissertation, North Carolina State University, Raleigh, NC, June 2002.