

Performance Analysis of an Ingress Switch in a JumpStart Optical Burst Switching Network ^{*†}

Lisong Xu

Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588-0115

Harry G. Perros [‡]

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

Abstract

We consider an ingress optical burst switching (OBS) node employing the JumpStart signaling protocol. The switch serves a number of users, each connected to the switch with a fiber link that supports multiple wavelengths. Each wavelength is associated with a 3-state Markovian burst arrival process which permits short and long bursts to be modeled. We model the ingress switch as a closed multi-class non-product-form queueing network, which we analyze approximately by decomposition. Specifically, we develop new techniques to analyze the queueing network, first assuming a single class of customers, and subsequently multiple classes of customers. These analytical techniques have applications to general queueing networks beyond the one studied in this paper. We also develop computationally efficient approximate algorithms to analyze an ingress switch in the limiting case where the number of wavelengths is large. The algorithms have a good accuracy, and they provide insight into the effect of various system parameters on the performance of an ingress OBS switch.

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

†Part of this paper has been published in the Proceedings of IEEE INFOCOM 2003.

‡Correspondence author, Phone: 919-515-2041, E-mail:hp@csc.ncsu.edu

1 Introduction

Optical burst switching (OBS) [22, 28, 33, 31] 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 a 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. A review of the main features of OBS can be found in Bettestilli and Perros [6, 7].

An OBS link can be modeled using the Erlang loss queue, see Chaskar *et al.* [12], Dolzer *et al.* [14], Verma *et al.* [29], and Wei *et al.* [32], . This queueing system consists of a finite number of servers with no buffer and Poisson arrivals. Each server represents a wavelength of the OBS link, and the service time represents the time it takes to transmit a burst. The probability that a burst arriving at an OBS link will be lost is calculated using the Erlang B formula. This formula is a function of the rate of arrival, the number of servers, and the mean service time, and it is insensitive to the distribution of the service time. The assumption that bursts arrive in a Poisson fashion is a convenient assumption, as it makes the queueing model easier to analyze, but it has certain disadvantages. For instance, it does not capture the notion of burstiness typically associated with high-speed networks, and also, it does not allow for the fact that during the time a burst is being transmitted, the source cannot start the transmission of another burst. A more appropriate burst arrival process is the on-off process, which has been used by Detti *et al.* [13], and Vu *et al.* [30]. The impact of self-similarity traffic on an OBS node is studied by Rajaduray *et al.* [23]. Finally, we note that the $M/M/m/K$ queue, where m is the number of wavelengths and $K - m$ is the capacity of the buffer, has been used to model a buffered OBS link, see Yoo *et al.* [37], and Yao *et al.* [36]. More complicated queueing models of a buffered OBS switch are analyzed by Callegati [11], Lu *et al.* [16], and Zhang *et al.* [40].

The Erlang loss queue has also been used to model an OBS network, see Rosberg *et al.* [25, 24] and Zalesky *et al.* [39]. In these papers, a network of OBS nodes is represented by a network of Erlang loss queues, where a customer may occupy one server in one or more Erlang loss nodes. Customers arrive in a Poisson manner. This network of Erlang loss queues readily arises in teletraffic and it has been used to calculate call blocking probabilities. Since it does not admit a closed-form solution, the queueing network is analyzed approximately by decomposing it to individual Erlang loss queues. The arrival rate into each queue is iteratively adjusted to account for blocking at other Erlang loss queues. In view of this, this approximation method is known as the *reduced load fixed-point approximation*.

In this paper, we develop a queueing network model of an ingress OBS switch employing the **JumpStart** signaling protocol [3, 1]. Specifically, the ingress OBS switch 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. This arrival process is a generalization of the well-known on-off arrival process, and it can be used to capture a wide range of scenarios of the burst arrival stream. We consider an ingress switch both with and without converters.

In [34], we model an ingress switch as a closed non-product-form queueing network which we analyze by

decomposition. Specifically, we develop an algorithm for the single-class case where all the wavelengths are associated with the same arrival process. This algorithm is then extended to the multi-class case, where each wavelength is associated with a different arrival process. The two algorithms described in [34] are shown to have a good accuracy. However, they are computationally intensive and they cannot be used to analyze an OBS ingress node with a large number of wavelengths, say 64 or more. In view of this, we present a computationally efficient algorithm for the limiting case where the number of wavelengths per port is very large. With the rapid advances in optical technology, such a limiting case is of great practical importance in the design of switches and networks. This algorithm takes only a few minutes to run while simulation may take several hours. Therefore, it can be used for extensive “what-if” analysis that would not be otherwise possible.

The queueing network model analyzed in this paper is a generalization of the Engset model [15] proposed by Detti *et al.* [13]. *First*, The Engset model can be used to represent a single output port of an OBS switch, whereas our model can represent all output ports of an OBS switch. *Second*, in order to model an output port with the Engset model, Detti *et al.* [13] assume that the duration of a dropped burst is zero. However, Zukerman *et al.* [41] demonstrate that this assumption may introduce certain discrepancy, especially with high traffic load. *Third*, in the Engset model, a burst is assumed to be lost if it arrives at a time when all the wavelengths of the OBS link are busy. In our model we assume that a blocked burst is not lost. Rather, the user associated with the burst will undergo a delay and then try again to transmit the same burst to the same OBS link of the ingress node. This is repeated until the burst is successfully transmitted. This is an important mechanism, as it eliminates burst loss at the interface between the user and the OBS ingress switch. However, our proposed queueing model is then significantly different from that described in [13, 41], and much harder to analyze. *Finally*, we note that our model can also be easily modified to represent the case where a blocked burst is simply lost.

Following this introduction, we describe briefly the network architecture and the operation of an ingress OBS node in Section 2. In Section 3, we present the burst arrival process, and in Section 4, we describe a queueing network model of the ingress OBS node. Sections 5 and 6 describe an algorithm for analyzing this queueing network without and with wavelength converters, respectively, assuming a single-class of customers. In Section 7, we introduce a new decomposition method for analyzing a multi-class generalization of this queueing network, and in Section 8, we introduce computationally efficient approximate algorithms for analyzing the ingress node in the limiting case where the number of wavelengths is large. The detailed description of these algorithms is given in Appendices A and B. Section 10 gives a brief analysis of the complexity of our proposed algorithms. In Section 9 we validate the accuracy of the proposed approximate algorithms by comparing to simulation results, and we demonstrate how the different parameters affect the performance of an ingress OBS node. Finally, we conclude the paper in Section 11.

2 JumpStart Project

The **JumpStart** project [3, 1, 21] is a joint MCNC/NCSU research effort addressing the design, specification, performance evaluation, and hardware implementation of a signaling protocol for OBS networks. The signaling protocol is based on the work by Wei and McFarland [31] and it is described in Baldine *et al.* [3, 4] and Zaim *et al.* [38], and it has been deployed in the ATDNet testbed [27, 2] in Washington, DC.

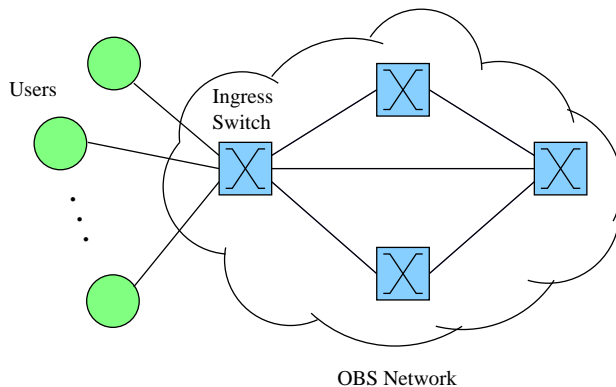


Figure 1: JumpStart Network Architecture

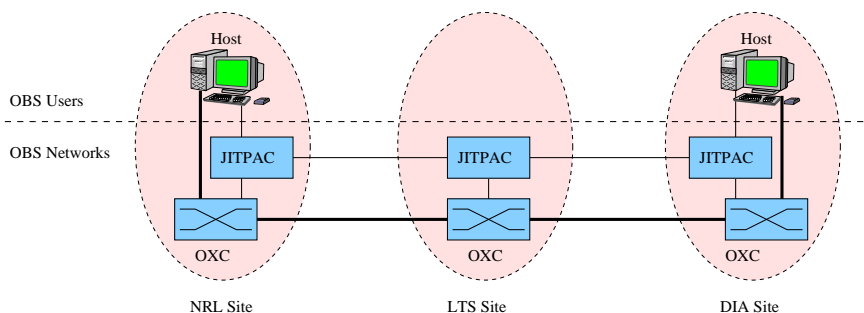


Figure 2: JumpStart Network Testbed

2.1 Network Architecture

Figure 1 shows the OBS network architecture of JumpStart project. A JumpStart OBS network consists of OBS nodes interconnected by bidirectional fiber links. Each fiber carries $W + 1$ wavelengths. One wavelength is used to transmit control packets, and the other W wavelengths are used to transmit data bursts. A user is capable of transmitting on any of these $W + 1$ wavelengths simultaneously. That is, the user is equipped with $W + 1$ transmitters.

An important feature of the JumpStart Project is *data transparency* [3]. There is no O/E conversion on a data channel, and data transport is transparent to a JumpStart OBS network. Furthermore, the JumpStart architecture makes no assumptions about the format of a burst it carries, and instead it schedules a time period for the burst on wavelengths within the network. The JumpStart signaling protocol assists users to negotiate the burst format, however, the format of a burst has no impact on the network. For example, a burst may consist of several IP packets, a few ATM cells, a stream of HDTV frames, or the raw bit streams from remote sensors. Since it is the responsibility of a user to define the format of a burst, a JumpStart ingress switch does not provide any burst assembly process, which is usually associated with ingress switches in other types of OBS networks.

To further illustrate the difference between JumpStart networks and other types of OBS networks, we show the JumpStart OBS testbed [27, 2] in Figure 2. The Advanced Technology Demonstration Network (ATDnet) is a high-speed optical testbed in Washington DC established by DARPA. The JumpStart team

has constructed a JumpStart OBS testbed connecting three ATDnet sites: the Naval Research Laboratory (NRL), the National Security Agency’s Laboratory for Telecommunications Science (LTS), and the Defense Intelligence Agency (DIA). Each site has a high-performance host and an optical crossconnect (OXC), and the OXC is controlled by a JITPAC (Just-in-Time Protocol Acceleration Circuit) developed by the JumpStart team. The host at each site can be considered as a user in Figure 1, and the OBS network consists of the OXC at each site. As described above, the JumpStart network makes no assumption about the format of a burst, and a user can transmit any types of bursts over the testbed. In 2002, the JumpStart team successfully transmitted HDTV (High Definition Television) signals at 1.5 Gbps rate over the testbed. In 2004, the JumpStart team demonstrated bulk file transfer using GridFTP over the testbed. More detailed description of the messages between the host (i.e a user) and the OXC (i.e. the ingress switch) is given in the next subsection.

2.2 An Ingress Switch

Following the JumpStart JIT signaling protocol [3], a user first sends a **setup** message to its ingress OBS node. The **setup** message carries information related to the burst transmission, including the source and destination addresses, and the burst wavelength. If the ingress node can switch the burst, 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 there is a destination port conflict. In this case, the ingress node returns a **reject** message. The user goes through a random delay, and it then re-transmits the **setup** message. This is repeated until the burst is successfully transmitted.

Note that an ingress switch of a JumpStart network always sends a feedback message (either **setup ack** or **reject**) to its user, whereas an ingress switch of other types of OBS networks does not send any feedback message to its user. This is an important mechanism of a JumpStart network, as it eliminates burst loss at the interface between a user and the OBS ingress switch¹. However, since a user continues to re-transmit the **setup** message until it receives a **setup ack** message, the queueing model that we propose for an ingress switch contains some special customers, and is very hard to be analyzed.

In JumpStart [3], the OBS node allocates resources within its switch fabric for a burst at the moment that it decides to accept the **setup** message. The user may indicate the length of its burst transmission in the **setup** message. This information is used by the OBS node to determine when to free the resources allocated to the burst. Alternatively, the user may simply send a **release** message to the node to indicate the end of its transmission. In this case, the OBS node frees the resources allocated to the burst upon receipt of the **release** message. 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 ingress node is shown in Figure 3. A user can be in one of 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. Since a user can simultaneously transmit bursts on different wavelengths, it can be in a different state for each burst wavelength.

¹Alternatively, we can assume that the user drops the burst if it receives a **reject** message. This case can be easily taken into account in our model. For further details, see [35].

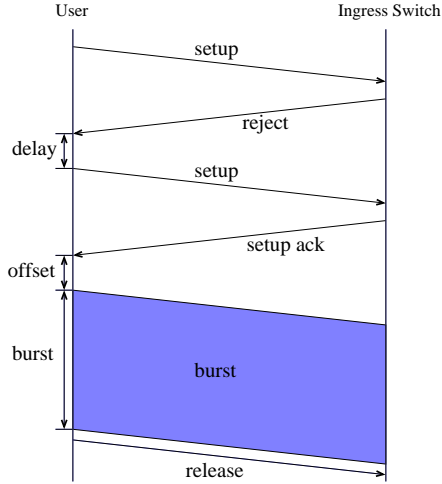


Figure 3: Signaling messages in JumpStart

3 The Burst Arrival Process

3.1 Related Work and Motivation

For the first cut performance analysis, in most of the reported performance modeling studies of an OBS network, the burst arrival process is assumed to be Poisson distributed. This is a convenient assumption, since it makes the queuing model easier to analyze, but it has certain disadvantages. First, it does not capture the notion of burstiness typically associated with high-speed networks, and second, it does not allow for the fact that during the time a burst is being transmitted, the user cannot start the transmission of another burst.

A more appropriate burst arrival process is the ON-OFF process, where an ON period corresponds to the duration of a burst, and an OFF period corresponds to the interval between two consecutive bursts. Detti *et al.* [13] analyze an OBS node with ON-OFF burst arrival processes. To simplify the analysis, they neglect the fact that a dropped burst is still transmitted on its input wavelength for its duration, and assume that the duration of a dropped burst is zero. With this assumption, an OBS node can be analyzed with the Engset formula [15], which is well-known for its insensitivity to the distributions of ON and OFF periods. However, Zukerman *et al.* [41, 30] demonstrate that this assumption may introduce discrepancy, and in certain cases especially with high traffic load, the Engset model is not appropriate for OBS modeling. They give an algorithm to accurately analyze an OBS node with ON-OFF burst arrival processes.

Zukerman *et al.* [41] also demonstrate by simulation that the performance of an OBS node with ON-OFF burst arrival processes is sensitive to the distributions of ON and OFF periods, but not too significant. Our simulation result (e.g. as shown in Figures 10 to 12 in Section 9) shows that the performance of a JumpStart ingress switch is very sensitive to the distributions of ON and OFF periods, especially for a switch with a small number of wavelengths. This is because the signaling protocol of the JumpStart project is different from that used in other types of OBS networks.

As described in the last section that the JumpStart architecture makes no assumptions about the format

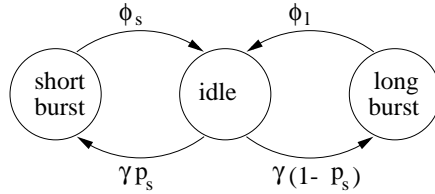


Figure 4: The burst arrival process

of a burst. The JumpStart architecture is also very flexible in the duration of a burst. A burst could be very small as an aggregation of several IP packets lasting for only a few tens of milliseconds, or could be very long like an optical circuit lasting for a few months [1]. Therefore, an exponential distribution may not be appropriate for modeling the duration of a burst. Since the performance of a JumpStart node is sensitive to the distribution of a burst duration, we use a generalization of an ON-OFF process as described in the next subsection,

3.2 Description of the Burst Arrival Process

In this paper, we associate each burst wavelength from a user to an OBS ingress switch with a burst arrival process which is modeled by the three-state Markov process shown in Figure 4. 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 *blocked* state of a user is not modelled in the burst arrival process, and will be modelled later in the queueing network model. The duration of a burst, whether short or long, is assumed to be exponentially distributed. Also the duration of the idle time is assumed to be exponentially distributed.

In this model, we assume that the user becomes idle after the transmission of each burst. That is, the user does not transmit bursts back-to-back. This assumption can be easily removed by modifying the three-state Markov process in Figure 4 to include transitions between the **short burst** and **long burst** states. Also, more complicated burst arrival processes can be modeled by introducing additional states and appropriate transitions between them. For instance, instead of using only two burst lengths (short and long), we may introduce $k > 2$ 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 ingress OBS node can be extended to these more general burst arrival processes. However, incorporating more general arrival processes in the model introduces additional complexities in the expressions. Therefore, in order to keep the analysis simple, we only consider the three-state Markov process shown in Figure 4.

The burst arrival process of Figure 4 can be characterized completely by the four parameters: **(1)** $1/\gamma$, the mean duration of the **idle** state, **(2)** $1/\phi_s$, the mean duration of the **short burst** state, **(3)** $1/\phi_l$, the mean duration of the **long burst** state, and **(4)** p_s , the probability that a burst is a small burst. In this paper, we also use the following four parameters to characterize the arrival process: **(1)** the *load* l of the burst arrival process, defined as the percentage of time that a wavelength is used for transmitting bursts, **(2)** the mean burst size $E(B)$ of both short bursts and long bursts, **(3)** the *ratio* $r = \phi_s/\phi_l$, of the mean

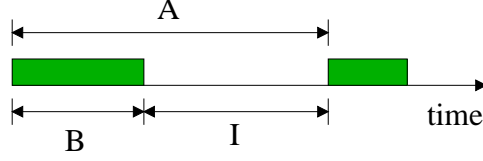


Figure 5: Relationship between random variables A , B , and I

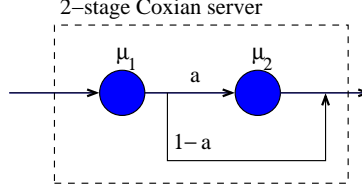


Figure 6: 2-stage Coxian server

long burst duration to the mean short burst duration, and **(4)** the probability p_s that a burst is a small burst. Given the first set of parameters, the second set of parameters can be obtained as follows:

$$l = \frac{p_s/\phi_s + (1-p_s)/\phi_l}{1/\gamma + p_s/\phi_s + (1-p_s)/\phi_l}, \quad E(B) = \frac{p_s}{\phi_s} + \frac{1-p_s}{\phi_l}, \quad r = \frac{\phi_s}{\phi_l}, \quad (1)$$

while given the second set of parameters, the first set can be obtained as:

$$\gamma = \frac{l}{(1-l)E(B)}, \quad \phi_s = \frac{p_s + (1-p_s)r}{E(B)}, \quad \phi_l = \frac{\phi_s}{r} \quad (2)$$

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 5. Let $L_A(s)$, $L_B(s)$, and $L_I(s)$ denote their Laplace transform, respectively. We have that:

$$L_A(s) = L_B(s)L_I(s) = p_s \frac{\phi_s}{\phi_s + s} + (1-p_s) \frac{\phi_l}{\phi_l + s} \frac{\gamma}{\gamma + s} \quad (3)$$

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

$$E[A] = p_s \frac{1}{\phi_s} + (1-p_s) \frac{1}{\phi_l} + \frac{1}{\gamma} \quad (4)$$

$$E[A^2] = p_s \left(\frac{1}{\phi_s^2} + \frac{1}{\gamma\phi_s} + \frac{1}{\gamma^2} \right) + (1-p_s) \left(\frac{1}{\phi_l^2} + \frac{1}{\gamma\phi_l} + \frac{1}{\gamma^2} \right) \quad (5)$$

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

$$c^2(A) = \frac{E(A^2)}{E^2(A)} - 1 \quad (6)$$

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 4 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 explained above. In this work we do not consider correlated inter-arrival times.

Since a customer may request either a short or a long burst with probabilities p_s and $1 - p_s$, respectively, the burst time distribution is a two-stage hyper-exponential distribution. It is well-known that this distribution can be represented by a two-stage Coxian server (see Figure 6), if its squared coefficient of variation c^2 is greater than 1. We will let μ_1 and μ_2 denote the service rate of the first and second stage of the Coxian server, respectively, and a denote the probability that, upon completion of the first service stage, the customer 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:

$$\mu_1 = \phi_s, \quad \mu_2 = \phi_l, \quad a = \frac{(1 - p_s)(\phi_s - \phi_l)}{\phi_s} \quad (7)$$

From the Coxian distribution we can easily calculate the mean $E(B)$ and the squared coefficient of variation $c^2(B)$ of the burst duration. We have:

$$E(B) = \frac{p_s}{\phi_s} + \frac{1 - p_s}{\phi_l} \quad (8)$$

$$c^2(B) = \frac{2}{E^2(B)} \left(\frac{p_s}{\phi_s^2} + \frac{1 - p_s}{\phi_l^2} \right) - 1 \quad (9)$$

4 A Queueing Network Model of an Ingress OBS Node

We consider an ingress OBS node with P input (and output) ports, and with $N, N \leq P$, users, each attached to the node over a fiber carrying $W + 1$ wavelengths. The traffic on each of the W burst wavelengths from a user to the ingress node is generated by the burst arrival process described in Section 3. 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 ingress 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 ingress 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. This is an important mechanism of a JumpStart network, as it eliminates burst loss at the interface between a user and the OBS ingress switch. However, the queueing model for a JumpStart ingress node is then quite different from that for an ingress node in other types of OBS networks, such as the one described in [13, 41]. Therefore, we cannot use the algorithm developed in [13, 41] to analyze our model.

4.1 Ingress OBS Node Without Converters

Let us first consider an ingress OBS node without 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 ingress switch on different wavelengths do not interfere with each other. Consequently, the ingress node can be decomposed into W sub-systems, one per burst wavelength. This decomposition is exact. Each sub-system

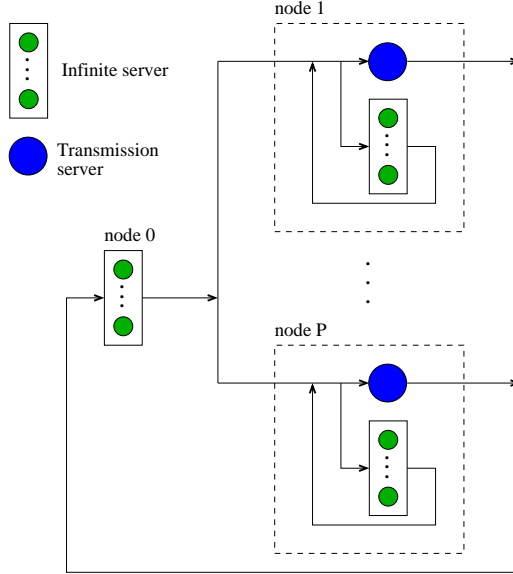


Figure 7: Queueing network model of a sub-system of an ingress switch without converters

$w, w = 1, \dots, W$, 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 ingress switch. Therefore, each sub-system has N burst arrival processes.

The queueing network model of a sub-system is shown in Figure 7; 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., 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 7. 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 q_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. In view of this, it is modeled by a two-stage Coxian distribution described by equation (7). 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.

When all N customers have the same burst arrival process and branching probabilities, the closed queueing network model consists of a single class of N customers and $P + 1$ nodes. If each customer has a different burst arrival process or branching probabilities, then the queueing network becomes a multi-class queueing network with $P + 1$ nodes and N classes, where each class contains exactly one customer.

4.2 Ingress OBS Node With Converters

Let us now consider an ingress 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 ingress switch into sub-systems per wavelength is no longer possible, since user bursts arriving on different wavelengths may interfere with each other. However, the ingress switch *as a whole* can be modeled by a closed queueing network very similar to the one shown in Figure 7. 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 7. Similarly, each node $i, i = 1, \dots, P$, in the new queueing network corresponds to each of the output ports of the ingress 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 7. 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 1. Sections 5 and 6 describe an algorithm for analyzing this queueing network without and with wavelength converters, respectively, assuming a single-class of customers. In Section 7, we introduce a new decomposition method for analyzing a multi-class generalization of this queueing network. The details of this algorithm are given in Appendix A. Finally, in Section 8 we introduce computationally efficient approximate algorithms for analyzing the ingress node in the limiting case where the number of wavelengths is large. The details of these algorithms are given in Appendix B

5 Analysis of the Single-Class Queueing Network without Converters

In this section, we analyze our proposed queueing network model for an optical switch without wavelength converters, and assuming a single class of customers. In queueing theory, BCMP [5] networks are of most interest, as they have a product-form solution, and can be analyzed exactly. However, because of the orbiting customers and the Coxian service time, the queueing network in Figure 7 is not a BCMP network, and does not have a non-product-form solution. We analyze it using Marie's algorithm [17, 18]. The idea

Table 1: Notation used in the analysis

Parameter	Description
N	number of users connected to an ingress switch
P	number of input (output) ports of an ingress switch
W	number of burst wavelengths in a fiber
$1/\gamma$	mean duration of the <code>idle</code> 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	probability that a burst is a short burst
q_i	probability that destination output port of burst is i
$E(B)$	mean duration of a burst (short or long)
l	load of the burst arrival process
$r = \phi_s/\phi_l$	ratio of mean long burst to mean short burst
μ_1, μ_2, a	parameters of 2-stage Coxian distribution of burst size

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.

5.1 The Flow Equivalent Server

Let us consider node $i, i = 1, \dots, P$, of the queueing network shown in Figure 7. 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 of the transmission server 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 8. 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 8 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 8, we have the following global balance equations:

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

$$p(1, 1, n - 1)(\lambda(n) + \mu_1) = p(0, 0, n)n\omega + p(1, 1, n - 2)\lambda(n - 1) + p(0, 0, n - 1)\lambda(n - 1), \quad 0 < n \leq N \quad (11)$$

$$p(1, 2, n - 1)(\lambda(n) + \mu_2) = p(1, 1, n - 1)a\mu_1 + p(1, 2, n - 2)\lambda(n - 1), \quad 0 < n \leq N \quad (12)$$

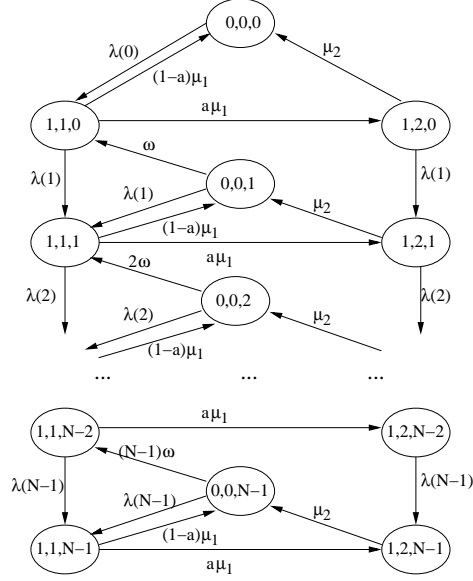


Figure 8: State transition rate diagram of node i , $i = 1, \dots, P$, of the queueing network of Figure 7

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

$$p(n) = \begin{cases} p(0, 0, 0), & n = 0 \\ p(0, 0, n) + p(1, 1, n-1) + p(1, 2, n-1), & 0 < n < N \\ p(1, 1, N-1) + p(1, 2, N-1), & n = N \end{cases} \quad (13)$$

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

$$v(n) = \begin{cases} 0, & n = 0 \\ \frac{p(1,1,n-1)}{p(n)}(1-a)\mu_1 + \frac{p(1,2,n-1)}{p(n)}\mu_2, & n > 0 \end{cases} \quad (14)$$

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 (10), (11), and (12) together, and after simplifying the result using equations (13) and (14), we obtain:

$$p(0)\lambda(0) = p(1)v(1), \quad n = 0 \quad (15)$$

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

$$p(N)v(N) = p(N-1)\lambda(N-1), \quad n = N \quad (17)$$

Using equations (15) and (16) recursively, we finally get:

$$p(n-1)\lambda(n-1) = p(n)v(n), \quad 0 < n \leq N \quad (18)$$

completing the proof of the theorem. ■

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

$$\begin{aligned} v(0) &= 0 \\ v(1) &= \frac{\mu_1 \omega (\lambda(1) - a\lambda(1) + \mu_2)}{(\lambda(1) + \omega)(\lambda(1) + a\mu_1 + \mu_2)} \\ v(n) &= \frac{n\mu_1 \omega (\lambda(n-1) + (n-1)\omega)(\lambda(n) - a\lambda(n) + \mu_2)}{Z}, \quad 1 < n \end{aligned} \quad (19)$$

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 (14), we can rewrite expression (10) as follows:

$$p(0, 0, n)(\lambda(n) + n\omega) = p(1, 1, n)(1-a)\mu_1 + p(1, 2, n)\mu_2 = p(n+1)v(n+1) = p(n)\lambda(n) \quad (20)$$

We can also rewrite expression (11) as:

$$\begin{aligned} p(1, 1, n-1)(\lambda(n) + \mu_1) &= p(0, 0, n)n\omega + p(1, 1, n-2)\lambda(n-1) + p(0, 0, n-1)\lambda(n-1) \\ &= p(0, 0, n)n\omega + \frac{p(1, 1, n-2) + p(0, 0, n-1)}{p(n-1)}p(n-1)\lambda(n-1) \\ &= p(0, 0, n)n\omega + \frac{p(1, 1, n-2) + p(0, 0, n-1)}{p(n-1)}p(n)v(n) \end{aligned} \quad (21)$$

Using expressions (13), (14), (20), and (21), we get the following group of equations for $n > 1$:

$$\left\{ \begin{array}{l} p(n) = p(0, 0, n) + p(1, 1, n-1) + p(1, 2, n-1) \\ p(n-1) = p(0, 0, n-1) + p(1, 1, n-2) + p(1, 2, n-2) \\ v(n) = \frac{p(1, 1, n-1)}{p(n)}(1-a)\mu_1 + \frac{p(1, 2, n-1)}{p(n)}\mu_2 \\ v(n-1) = \frac{p(1, 1, n-2)}{p(n-1)}(1-a)\mu_1 + \frac{p(1, 2, n-2)}{p(n-1)}\mu_2 \\ p(0, 0, n)(\lambda(n) + n\omega) = p(n)\lambda(n) \\ p(0, 0, n-1)(\lambda(n-1) + (n-1)\omega) = p(n-1)\lambda(n-1) \\ p(1, 1, n-1)(\lambda(n) + \mu_1) = p(0, 0, n)n\omega \\ \quad + \frac{p(1, 1, n-2) + p(0, 0, n-1)}{p(n-1)}p(n)v(n) \end{array} \right. \quad (22)$$

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 (22) to obtain the expression for $v(n)$, $n > 1$, as in (19).

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

$$\left\{ \begin{array}{l} p(1) = p(0, 0, 1) + p(1, 1, 0) + p(1, 2, 0) \\ v(1) = \frac{p(1, 1, 0)}{p(1)}(1-a)\mu_1 + \frac{p(1, 2, 0)}{p(1)}\mu_2 \\ p(0, 0, 1)(\lambda(1) + \omega) = p(1)\lambda(1) \\ p(1, 1, 0)(\lambda(1) + \mu_1) = p(0, 0, 1)\omega + p(1)v(1) \end{array} \right. \quad (23)$$

to obtain the expression for $v(1)$ as in (19). 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.

5.2 The Iterative Algorithm

We use Marie's algorithm [17] to analyze the queueing network of Figure 7. 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:

$$\left| \frac{N - \sum_{i=0}^P \sum_{j=0}^N j p_i(j)}{N} \right| < \epsilon \quad (24)$$

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

$$\left| \frac{r_i - \frac{1}{P+1} \sum_{j=0}^P r_j}{\frac{1}{P+1} \sum_{j=0}^P r_j} \right| < \epsilon, \quad i = 0, 1, \dots, P \quad (25)$$

$$r_i = \begin{cases} \frac{1}{q_i} \sum_{j=0}^N p_i(j) \mu_i(j), & i = 1, \dots, P \\ \sum_{j=0}^N p_i(j) \mu_i(j), & i = 0 \end{cases} \quad (26)$$

6 Analysis of the Single-Class Queueing Network with Converters

As we discussed in Section 4, the only difference between the queueing network model of an ingress switch with wavelength converters and the queueing network of Figure 7 (which models a single wavelength subsystem of an ingress 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 [26] 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:

$$v_i(n_i) = \frac{p_i(n_i - 1) \lambda_i(n_i - 1)}{p_i(n_i)} \quad (27)$$

Finally, we use the same iterative algorithm described in Section 5.2, to analyze this more general queueing network.

7 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 7. The technique is quite general and it can be applied to a wide class of multi-class queueing networks.

Consider an ingress 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 $C = NW$. 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 [19, 9], mainly by extending Marie's algorithm [17]. Neuse and Chandy [19] 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 [9], 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 [9]. 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 developed 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. We use the convolution algorithm to calculate the arrival rate to the aggregate class and not the Baynat and Dallery's method [9] since, as we mentioned above, the latter is not scalable. The detailed description of our algorithm is given in Appendix A.

8 An Ingress Node with a Large Number of Wavelengths

The algorithms described in previous sections have a good accuracy, but they are still computationally intensive if we want to use them to analyze an OBS ingress node with a large number of wavelengths, say 64 or more. In this section, we describe approximate algorithms for the analysis of the queueing network shown in Figure 7 under the assumption that the number W of wavelengths per fiber is large. These algorithms are based on two properties of the above queueing network when W is large, described below in Section 8.1. The

approximation algorithms for the queueing network when W is large are described in detail in Appendix B.

8.1 Two Properties when W is Large

The two properties described in this section have been verified by simulation. They have not been rigorously proved.

Property 1: As the number W of wavelengths increases, the performance of the queueing network under study becomes dependent on the mean burst size, and it is relatively insensitive to the burst size distribution.

A similar property was reported in [10]. They found using simulation that, if arbitrarily distributed service times are replaced by exponentially distributed service times in a closed queueing network, the deviation in the performance measures is tolerable. This property is referred to as the *robustness* of the closed queueing network. They observed that the mean deviation is about 6% if in a network with only $-/G/1$ FCFS nodes, each $-/G/1$ node is replaced by an $-/M/1$ node. For a network with only $-/G/m$ FCFS nodes, the corresponding mean deviation is about 2%, while for a network with only $-/G/\infty$ nodes the mean deviation is zero. The authors reported that this approximation is more accurate if (i) the c^2 of the service time is not very large, (ii) there is a bottleneck in the queueing network, and (iii) the number of customers increases. Note that in our case, as W increases, both the number of customers in the queueing network and the number of servers in each node increase.

We have found out by simulation that Property 1 holds for $W > 32$. Therefore, in this case we can analyze the queueing network shown in Figure 7 by assuming that the size of a burst has an exponential, rather than Coxian, distribution. The algorithm for the single-class queueing network described in Section B.1 is based on this approximation.

The second property is for the multi-class case of the queueing network shown in Figure 7. As we explained above in Section 4, the multi-class queueing network corresponds to the case of an ingress OBS node where each customer has a different burst arrival process or branching probabilities.

Property 2: As the number W of wavelengths increases, the mean waiting time of a class of customers in node i , $i = 1, 2, \dots, P$, tends to the mean waiting time of all classes in the same node.

We illustrate this behavior for the queueing network that represents an ingress switch with $N = 4$ users, $P = 5$ input/output ports, and mean orbiting rate $\omega = 0.1$. For simplicity, we assume that there are four classes and that all wavelengths of the same user belong to the same class. Table 2 lists the parameters of the burst arrival process for each user (class). Figure 9 shows the mean waiting time of each class 1-4 and of all classes at output port 5, as the number W of wavelengths increases from 2 to 32. We observe that when $W = 2$, the difference between the mean waiting time of the four classes is large. As W increases, however, this difference decreases, and when W is equal to 32 the mean waiting times of the four classes are very close to the mean waiting time of all classes. This property implies that when we analyze an OBS ingress switch with a large number of wavelengths, we can use a single-class queueing network to approximate the

Table 2: Parameters of the burst arrival processes for verifying Property 2

Parameter	User 1	User 2	User 3	User 4
load l	0.2	0.4	0.6	0.8
$E(B)$	2	4	6	8
ratio r	20	40	60	80
p_s	0.2	0.4	0.6	0.8
q_5	1	1	1	1
q_1, q_2, q_3, q_4	0	0	0	0

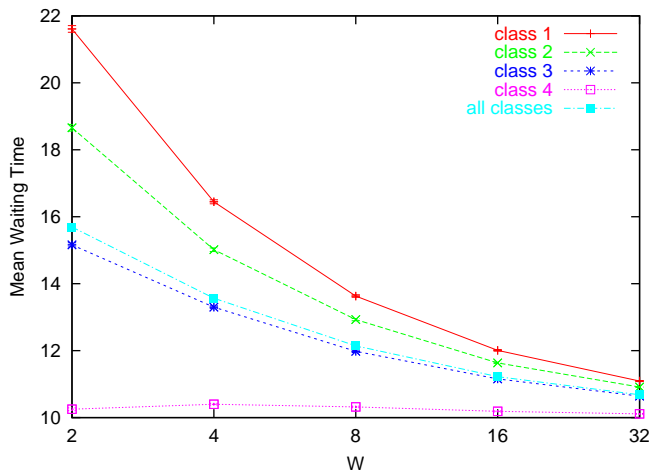


Figure 9: Mean waiting time for different classes

multi-class queueing network. The algorithm for a multi-class queueing network described in Section B.2 is based on this property and also on Property 1.

This property can be intuitively explained as follows. Let us consider a product-form closed queueing network with two classes, and let $(N^{(1)}, N^{(2)})$ be the population vector, where $N^{(1)}$ and $N^{(2)}$ are the number of class 1 and class 2 customers, respectively, in the queueing network. The closed queueing network consists of M/M/1 FIFO nodes and M/G/ ∞ nodes. At M/M/1 nodes, both classes have the same service time distribution, but at M/G/ ∞ nodes, classes may have different service time distributions.

Consider M/M/1 node i , and let μ_i denote its service rate. Let $W_i^{(c)}(N^{(1)}, N^{(2)})$ denote the mean waiting time of a class- c customer at node i in a network with a population vector $(N^{(1)}, N^{(2)})$, and let $L_i(N^{(1)}, N^{(2)})$ denote the corresponding mean number of customers at node i . By the *arrival theorem*, the distribution of the number of customers seen by a class- c customer at the time of arrival to node i is the same as that of the number of customers at node i with one less class- c customer in the network. Therefore, the mean waiting time of a class-1 customer and a class-2 customer can be obtained as follows:

$$W_i^{(1)}(N^{(1)}, N^{(2)}) = L_i(N^{(1)} - 1, N^{(2)})/\mu_i, \quad (28)$$

$$W_i^{(2)}(N^{(1)}, N^{(2)}) = L_i(N^{(1)}, N^{(2)} - 1)/\mu_i \quad (29)$$

Consider now the case where $N^{(1)} = 1$ and $N^{(2)} = 1$. Then, $W_i^{(1)}(1, 1) = L_i(0, 1)/\mu_i$, $W_i^{(2)}(1, 1) = L_i(1, 0)/\mu_i$. $L_i(0, 1)$ (respectively, $L_i(1, 0)$) is the mean number of customers at node i in a network with only a class-2 (respectively, class-1) customer. Since this closed queueing network has a product-form solution, both $L_i(0, 1)$ and $L_i(1, 0)$ can be calculated easily. They are dependent on the service time of each class at every node in the network. Recall that both class 1 and class 2 have the same service time distribution at M/M/1 nodes, but they have different service time distribution at M/G/ ∞ nodes. Therefore, $L_i(0, 1) \neq L_i(1, 0)$, and consequently $W_i^{(1)}(1, 1) \neq W_i^{(2)}(1, 1)$.

Consider now the case where $N^{(1)} = 1000$ and $N^{(2)} = 1000$. We have that $W_i^{(1)}(1000, 1000) = L_i(999, 1000)/\mu_i$, $W_i^{(2)}(1000, 1000) = L_i(1000, 999)/\mu_i$. $L_i(999, 1000)$ (respectively, $L_i(1000, 999)$) is the mean number of customers at node i in a network with 999 class-1 customers and 1000 class-2 customers (respectively, 1000 class-1 customers and 999 class-2 customers). We can see that there are 1998 common customers between the population vectors (999,1000) and (1000,999), 999 each of class 1 and 2. Since 99.95% (=1998/1999) of customers are the same in both population vectors, the percentage difference between $L_i(999, 1000)$ and $L_i(1000, 999)$ is smaller than the percentage difference between $L_i(0, 1)$ and $L_i(1, 0)$. The same is true for the pair of vectors $W_i^{(1)}(1000, 1000)$ and $W_i^{(2)}(1000, 1000)$ compared to vectors $W_i^{(1)}(1, 1)$ and $W_i^{(2)}(1, 1)$. In other words, the mean waiting time of different classes tends to the overall mean as the number of customers increases.

9 Numerical Results

In this section, we present results to illustrate how the different parameters affect the performance of the ingress OBS node. We also compare the approximate results to results obtained from simulation.

We simulate a JumpStart ingress switch as shown in Figure 1. On each wavelength from a user to the ingress switch, bursts are generated according to the burst arrival process described in Section 3. We assume that the distance between a user and the switch is very small, and the propagation delay is zero. Therefore, a user can get the acknowledgement from the switch immediately after it sends a `setup` message to the switch. In this paper, we do not consider the impact of burst offset on the performance an ingress switch, and then we assume that the burst offset is zero.

Simulation results are plotted along with 95% confidence intervals estimated by the method of batch mean [20]. The number of batches is set to 30, with each batch run lasting until each wavelength has transmitted at least 10,000 burst. As the reader will notice, however, most confidence intervals are very narrow and are barely visible in the figures.

A more comprehensive set of results for various performance measures and for a wide range of values of the system parameters listed in Table 1 can be found in [35].

9.1 Single-Class Network

In this section, we show results for $P = 16$ (i.e., a 16×16 ingress 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., $q_{16} = 0.1$, and $q_j = 0.06 \forall j \neq 16$). 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 10-12 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 sets, the mean orbiting time $1/\omega$ is 1. For both arrival processes, the mean burst size $E(B)$ is 1, the mean duration of the idle state $1/\gamma$ is 0.2, and the probability p_s is 0.99. However, the squared coefficient of variation $c^2(A)$ of the burst interarrival times is 1 for one set, and 100 for the other; the set with $c^2(A) = 1$ is very smooth, while the one with $c^2(A) = 100$ is extremely bursty. Note that both sets have the same mean burst size, and the only difference between two sets is that they have different burst size distribution. We can see later in this section that burst size distribution really affects the switch performance. That is, the switch performance is sensitive to not only the mean burst size, but also the burst size distribution.

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 5.

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 the analytical and simulation results across a wide range of system parameters [35]). 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 10-12 may take hours of simulation time, while it takes the approximate algorithm only a few seconds for the same computation.

From Figures 10-12 we also see the dramatic effect that the burstiness of the arrival process can have on the performance of the ingress 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 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 queueing model can be used for extensive “what if” analysis that would not be possible otherwise.

Figures 13-15 are similar, but present results for an ingress OBS node *with* converters. We consider two arrival processes as before, with the same mean burst size and mean idle 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 10-12 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 queueing 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 13-15 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.

9.2 Multi-Class Network

We now present results for the ingress switch when each user is associated with a different arrival process. We study the same 16×16 switch with $W = 32$ wavelengths and the same hot-spot traffic pattern as in the previous 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 16 and 17 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. For additional experiments and results for a multi-class OBS node, the reader is referred to [35].

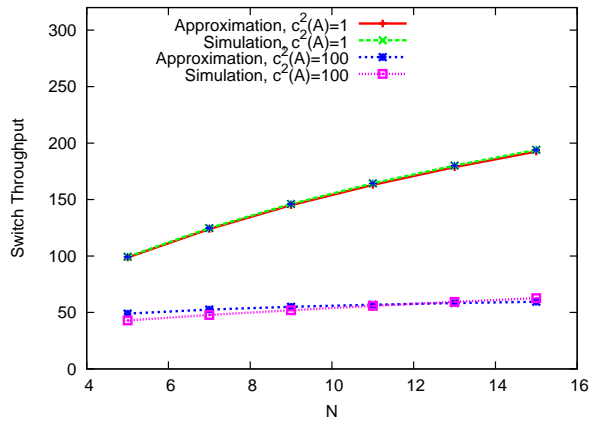


Figure 10: Switch throughput, no converters

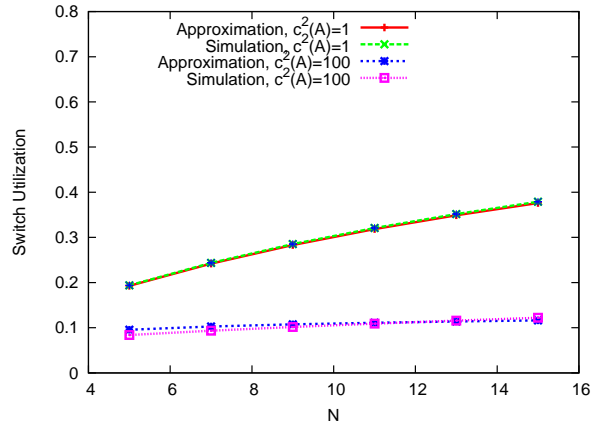


Figure 11: Switch utilization, no converters

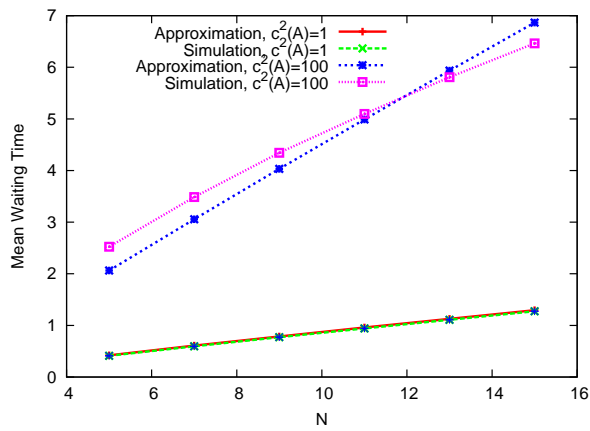


Figure 12: Mean waiting time, no converters

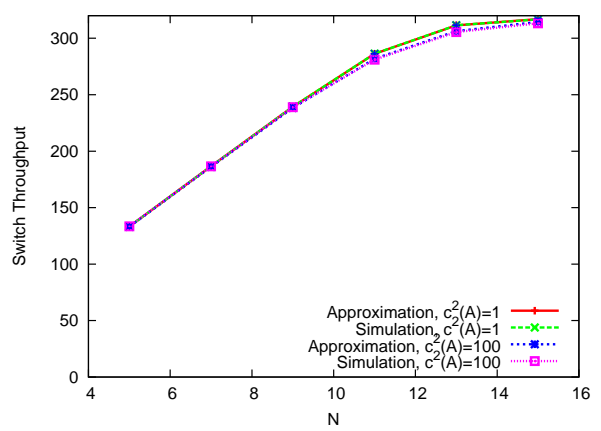


Figure 13: Switch throughput, with converters

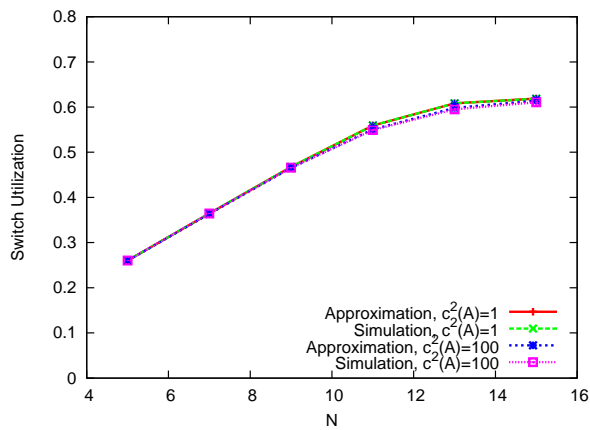


Figure 14: Switch utilization, with converters

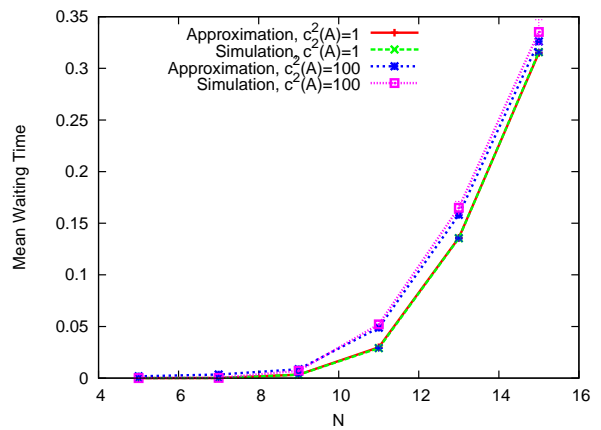


Figure 15: Mean waiting time, with converters

9.3 A Large Number of Wavelengths

In this section we present results for an ingress node with $N = 8$ users and $P = 10$ input/output ports. The queueing model of this ingress switch has $P + 1 = 11$ nodes. Nodes 1 to 10 represent the output ports of

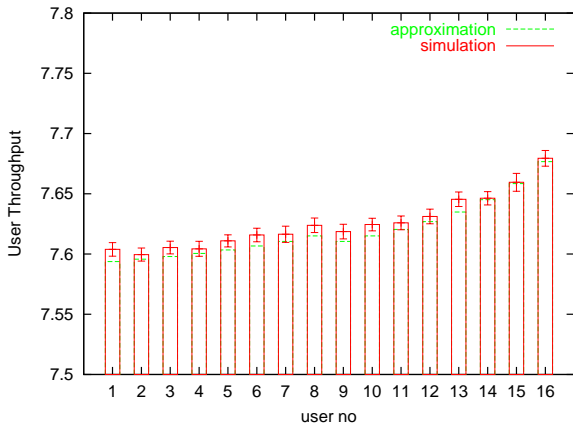


Figure 16: User throughput, without converters

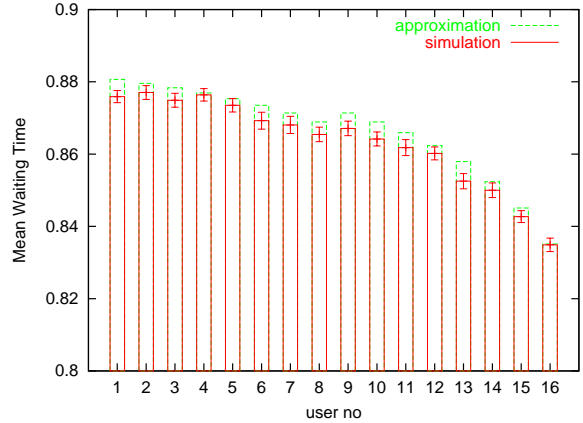


Figure 17: User mean waiting time, without converters

the ingress switch, and node 0 represents the burst arrival processes in the idle state. For the approximate algorithms, we vary the number of wavelengths from 2 to 128 in powers of two. However, limited by the running time of the simulation program, we only ran simulations with W values up to 32. As expected, the approximate algorithm does not give good results for values of $W < 32$ in the graphs presented below. We plot the results starting at $W = 2$, in order to show how its accuracy increases as W increases.

Overall, all the experiments we present indicate that the error due to our approximations decreases as W increases, and when $W = 32$, the approximate results are almost identical to the simulation results. We also note that when $W = 32$, the simulation takes between 60 and 80 hours to complete, whereas the approximate algorithm takes only a few minutes.

9.3.1 The Effect of the Traffic Load

In this subsection, we describe three experiments for an ingress node in which all wavelengths are associated with the same burst arrival process. In the first experiment, the parameters of the process are: load $l = 0.1$, mean burst size $E(B) = 1$, burst size ratio $r = 100$, and short burst probability $p_s = 0.9$. The destination probabilities are $q_9 = q_{10} = 0.3$ and $q_i = 0.05$ for $i = 1, \dots, 8$. The mean orbiting time $1/\omega$ is set to be 10 times of the mean burst size (i.e., $1/\omega = 10$, or $\omega = 0.1$). The second and third experiments have the same parameters as the first one, except that the load l is set to 0.5 and 0.9, respectively.

Figure 18 shows the switch mean waiting time obtained from these three experiments. The switch mean waiting time is the average waiting time of all users before they transmit a burst to the switch. (This waiting time represents the delay a user experiences from the instance it transmits a `setup` message to the instance that it starts transmitting the burst.) As expected, the higher the load, the longer the mean waiting time. We also observe that, as the number W of wavelengths increases, the mean waiting time decreases as well. Recall that in a switch with wavelength converters, a user has to wait and retransmit the `setup` message if and only if all wavelengths at the destination output port are busy. Intuitively, the larger the number of wavelengths, the smaller the probability that all wavelengths at an output port are busy. Therefore, as W increases, the switch mean waiting time decreases.

9.3.2 The Effect of the Traffic Pattern

Again we ran three experiments. In the first experiment, we assume that all wavelengths are associated with the same burst arrival process with the following parameters: load $l = 0.9$, mean burst size $E(B) = 1$, burst size ratio $r = 100$, short burst probability $p_s = 0.9$, destination probabilities $q_i = 0.1, i = 1, \dots, 10$, and mean orbiting rate $\omega = 0.1$. The second experiment has the same parameters as the first one, except that the destination probabilities are $q_9 = q_{10} = 0.2$ and $q_i = 0.075$ for $i = 1, \dots, 8$. In the third experiment, we set the destination probabilities to $q_9 = q_{10} = 0.3$ and $q_i = 0.05$. In other words, the first traffic pattern is uniform, while the last two are hot-spot traffic patterns with ports 9 and 10 receiving more traffic than the other 8 ports.

Figure 19 shows the switch mean waiting time for the three traffic patterns. We observe that the larger the q_9 and q_{10} , the longer the mean waiting time, i.e., a switch with a hot-spot traffic experiences a longer mean waiting time than a switch with a uniform traffic.

9.3.3 The Effect of the Orbiting Rate

As before, we performed three experiments in which we assume that all wavelengths are associated with the same burst arrival process with the following parameters: load $l = 0.9$, mean burst size $E(B) = 1$, burst size ratio $r = 100$, short burst probability $p_s = 0.9$, and destination probabilities $q_9 = q_{10} = 0.3$ and $q_i = 0.05$. The mean orbiting time $1/\omega$ is set to be 1, 10, and 100 times the mean burst size, respectively, in the first, second, and third experiment. Figure 20 plots the switch mean waiting time for these three experiments. As we can see, the longer the orbiting time, the longer the switch mean waiting time.

These experiments clearly show that the errors due to our approximations decrease as W increases. Also compared to the errors in other figures, the error in Figure 20 are larger. That is, the accuracy of our approximation algorithm is sensitive to the mean orbiting time $1/\omega$. The longer the orbiting time, the worse the accuracy of our approximation algorithm. The good news is, it is a reasonable assumption that in practice, the orbiting time is small, since a long orbiting time reduces the switch utilization, and increases the user waiting time.

9.3.4 The Effect of Different Burst Arrival Processes

We finally considered a multi-class queueing network in which wavelengths are associated with different burst arrival processes. For simplicity, we assume that the wavelengths of the same user are associated with identical burst arrival processes whose parameters are listed in Table 3. We also set the mean orbiting rate to $\omega = 0.1$. Figure 21 shows the mean waiting time of class (i.e., user) 2, 4, 6, and 8. This figure illustrates how the arrival process parameters affect the performance of a user. We observe that as the index of a user increases, the mean waiting time increases. That is because a user with a higher index has a higher load, and has more bursts to the “hot” ports 9 and 10.

Table 3: Parameters of the burst arrival processes for the multi-class OBS node of Section 9.3.4

Parameter	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8
load l	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
mean $E(B)$	1	2	3	4	5	6	7	8
ratio r	10	20	30	40	50	60	70	80
probability p_s	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
probability q_9, q_{10}	0.1	0.1	0.2	0.2	0.3	0.3	0.4	0.4
other q_i	0.1	0.1	0.075	0.075	0.05	0.05	0.025	0.025

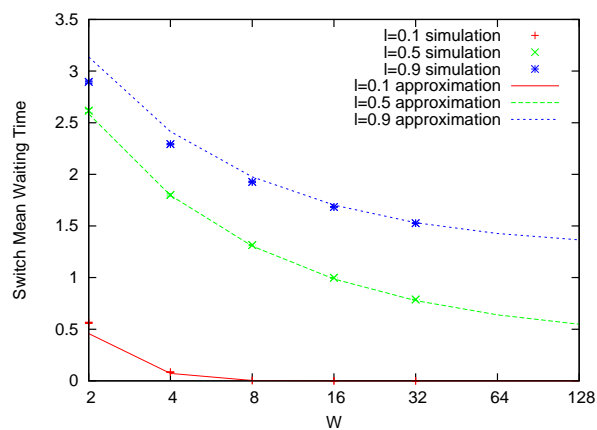


Figure 18: Effect of traffic load

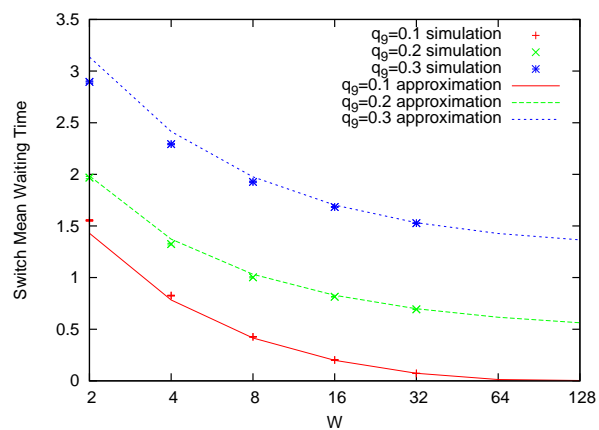


Figure 19: Effect of traffic pattern

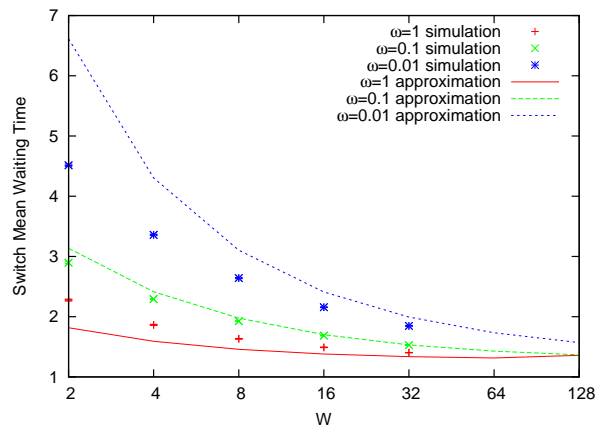


Figure 20: The effect of the orbiting rate

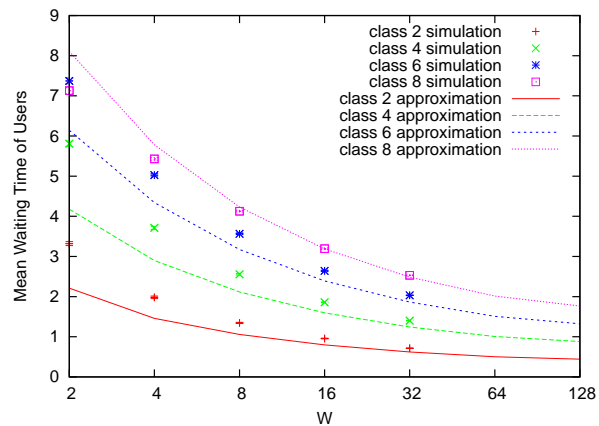


Figure 21: Different burst arrival processes

10 Complexity of Approximate Algorithms

In this paper, we present several approximate algorithms to analyze the performance of various OBS ingress switches. The algorithm described in Section 5 is the fastest one, since we are able to obtain the closed-form

Table 4: Complexity of all algorithms described in this paper (except the algorithm described in Section 5, which is the fastest one)

Approximate algorithms described in	Switch users	Number of states in a Markov chain	Number of chains in each iteration
Section 6	homogeneous	$O(PW^3)$	$O(P)$
Section 7 and Appendix A	heterogeneous	$O(P^2W^6)$	$O(P^2W)$
Section 8 and Appendix B.1	homogeneous	$O(PW^2)$	$O(P)$
Section 8 and Appendix B.2	heterogenous	$O(PW^2)$	$O(P)$

expression for the conditional throughput of a flow equivalent server. In all other algorithms, we calculate the conditional throughput of a flow equivalent node by analyzing the corresponding Markov chain numerically with the Gauss-Seidel method. In this section, we analyze the complexity of all algorithms except the one presented in Section 5.

Since the Gauss-Seidel method is computationally intensive, and is the dominant component in the complexity of those algorithms, we consider only the complexity caused by the Gauss-Seidel method, and neglect the effect of other components of an algorithm. For each algorithm described in this paper (except the one described in Section 5), Table 4 shows the number of states in a Markov chain to be solved by using the Gauss-Seidel method, and the total number of Markov chains to be solved in each iteration. P is the port number of an ingress switch, and W is the wavelength number of a port. The number of states of a Markov chain given in the table is for a switch with converters. In case of no converters, the number of states can be obtained by setting $W = 1$.

We note that since Section 6 considers homogeneous users and Section 7 considers heterogeneous users, the algorithm described in Section 7 is more complicated (i.e. larger Markov chain, and more chains to solve) than the one described in Section 6. Recall that the algorithms presented in Appendixes B.1 and B.2 are designed for a switch with large W . As shown in Table 4, they are indeed faster than the algorithms described in Sections 6 and 7, respectively. We also note that the algorithms presented in Appendixes B.1 and B.2 solve the same number of Markov chains with the same number of states in each iteration. However, since the algorithm in Appendix B.2 considers heterogeneous users, it may take more iterations to converge than the one described in Appendix B.1. As a concrete example, with my Dell Server², it takes about 10 minutes to analyze a switch with $P = 10$, $W = 128$, with converters, and with heterogeneous users for Figure 21.

11 Concluding Remarks

In this paper, we develop a queueing network model of an ingress OBS node with burst arrivals described by a general Markov process. An OBS ingress node is modeled as a closed non-product-form queueing network. This model is quite general, and it permits us to study the performance of an ingress switch under a wide

²PowerEdge SC1420: 2.8GHZ CPU, 1G Memory, 160G Hard Drive

range of traffic and operational scenarios.

We have presented algorithms for both the single-class case and the multi-class case. Our work makes two significant contributions. First, we developed a new technique to analyze non-product-form queueing networks with orbiting customers, a class of queueing networks that had not been analyzed before. Second, we presented a new technique to analyze heterogeneous networks with an arbitrary number of classes by constructing and analyzing a series of two-class networks. These analytical techniques have applications to general queueing networks beyond the one studied in this paper. We also have developed computationally efficient approximate algorithms to analyze an ingress OBS node in the limiting case where the number of wavelengths is large. With the rapid advances in optical technology, such a limiting case is of great practical importance in the design of switches and networks.

We are currently working on extensions of the queueing network models to analyze a network of OBS nodes, see Battestilli and Perros [8].

Appendix

A Analysis of the Multi-Class Queueing Network with or without Converters

In this section, we present in detail our algorithm for solving networks with more than two classes of customers. We first show in Section A.1 how to use HAM, see Neuse and Chandy [19], to solve networks with only two classes, and in Section A.2 we present our method to solve networks with more than two classes of customers. This method consists of decomposing a network with multiple classes of customers into a set of two-class networks, each of which is solved using HAM.

A.1 The Two-Class Queueing Network

In this subsection, we assume that there are only two classes of customers in the queueing network of Figure 7, namely, class 1 and 2. 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 ingress 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 7. Let $\lambda_i^{(1)}(n_i^{(1)}, n_i^{(2)})$ and $\lambda_i^{(2)}(n_i^{(1)}, n_i^{(2)})$ denote the arrival rate of class 1 and class 2 customers, respectively, to node i when there are $n_i^{(1)}$ class 1 customers and $n_i^{(2)}$ class 2 customers in the node. We also assume that the service time of class $j, j = 1, 2$, at the transmission server is a two-stage Coxian distribution with parameters $\mu_{(i,1)}^{(j)}, \mu_{(i,2)}^{(j)}$, and $a_i^{(j)}$. The state of node i can be described by the vector:

$$(n_i^{(t1,1)}, n_i^{(t2,1)}, n_i^{(o,1)}, n_i^{(t1,2)}, n_i^{(t2,2)}, n_i^{(o,2)}) \quad (30)$$

where

- $n_i^{(t1,1)}$ and $n_i^{(t1,2)}$ are random variables representing the number of class 1 and class 2 customers, respectively, being served by the transmission servers in phase one,
- $n_i^{(t2,1)}$ and $n_i^{(t2,2)}$ are random variables representing the number of class 1 and class 2 customers, respectively, being served in phase two, and
- random variables $n_i^{(o,1)}$ and $n_i^{(o,2)}$ represent the number of orbiting customers of class 1 and 2, respectively.

Let $p_i(n_i)$ be the steady-state probability that the total number of customers at node i is n_i . 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 from expression (27), where $p_i(n_i)$ and $\lambda_i(n_i)$ are calculated by:

$$p_i(n_i) = \sum_{n_i^{(1)} + n_i^{(2)} = n_i} p_i(n_i^{(t1,1)}, n_i^{(t2,1)}, n_i^{(o,1)}, n_i^{(t1,2)}, n_i^{(t2,2)}, n_i^{(o,2)}) \quad (31)$$

$$\lambda_i(n_i) = \frac{1}{p_i(n_i)} \sum_{n_i^{(1)} + n_i^{(2)} = n_i} \left[\lambda_i^{(1)}(n_i^{(1)}, n_i^{(2)}) + \lambda_i^{(2)}(n_i^{(1)}, n_i^{(2)}) \right] p_i(n_i^{(t1,1)}, n_i^{(t2,1)}, n_i^{(o,1)}, n_i^{(t1,2)}, n_i^{(t2,2)}, n_i^{(o,2)}) \quad (32)$$

where $n_i^{(1)} = n_i^{(t1,1)} + n_i^{(t2,1)} + n_i^{(o,1)}$, and $n_i^{(2)} = n_i^{(t1,2)} + n_i^{(t2,2)} + n_i^{(o,2)}$.

The conditional throughput $v_i(n_i)$ is used 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 $\lambda_i^{(1)}(n_i^{(1)}, n_i^{(2)})$ and $\lambda_i^{(2)}(n_i^{(1)}, n_i^{(2)})$ to each node $i, i = 1, \dots, P$. This process is repeated until convergence, as dictated by Marie's algorithm [17].

A.2 The 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.

A.2.1 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 in the original network. For the aggregate class, we have to specify the branching probability $q_i^{(\text{agg})}$ that an aggregate class customer leaving node 0 will enter node i , the parameters of the service rate $\mu_{(i,1)}^{(\text{agg})}, \mu_{(i,2)}^{(\text{agg})}$, and $a_i^{(\text{agg})}$ of the aggregate class at the transmission server of node $i, i = 1, \dots, P$, and the parameters of the service rate $\mu_0^{(\text{agg})}$ of the aggregate class at node 0.

Assuming that we know the mean response time $T_i^{(k)}$ of class k at node i , then we can calculate the throughput $H^{(k)}$ of class k in the network as follows:

$$H^{(k)} = \frac{N^{(k)}}{T_0^{(k)} + \sum_{i=1}^P T_i^{(k)} \times q_i^{(k)}} \quad (33)$$

where $N^{(k)}$ is the number of class k customers, (i.e., $N^{(k)} = 1$ for all k), and $q_i^{(k)}$ is the branching probability in the original network that a class k customer leaving node 0 will enter node i .

For a given class c , the branching probabilities $q_i^{(\text{agg})}$ of the aggregate class can be calculated by:

$$q_i^{(\text{agg})} = \frac{\sum_{k \neq c} (H^{(k)} q_i^{(k)})}{\sum_{k \neq c} H^{(k)}} \quad (34)$$

We employ the class aggregation technique in Baynat and Dallery's method [9] to obtain the parameters of the service time distribution of the aggregate class at the transmission server of node $i, i = 1, \dots, P$. 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 $E_i(Agg)$ and $E_i(Agg^2)$ of the distribution of the aggregate class at node i can be easily calculated as follows.

$$E_i(Agg) = \frac{\sum_{k \neq c} H^{(k)} q_i^{(k)} E(B^{(k)})}{\sum_{k \neq c} H^{(k)} q_i^{(k)}} \quad (35)$$

$$E_i(Agg^2) = \frac{\sum_{k \neq c} H^{(k)} q_i^{(k)} E^2(B^{(k)})(1 + c^2(B^{(k)}))}{\sum_{k \neq c} H_i^{(k)} q_i^{(k)}} \quad (36)$$

$$c_i^2(Agg) = \frac{E_i(Agg^2) - E_i^2(Agg)}{E_i^2(Agg)} \quad (37)$$

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 using expressions (8) and (9), respectively. We approximate the service time distribution of the aggregate class as a two-stage Coxian distribution by moment matching. We use Marie's method [18] to match the first two moments, as follows:

$$\mu_{(i,1)}^{(agg)} = \frac{2}{E_i(Agg)} \quad (38)$$

$$\mu_{(i,2)}^{(agg)} = \frac{1}{E_i(Agg)c_i^2(Agg)} \quad (39)$$

$$a_i^{(agg)} = \frac{1}{2c_i^2(Agg)} \quad (40)$$

The parameters of the service rate $\mu_0^{(agg)}$ of the aggregate class at node 0 can be obtained as follows:

$$1/\mu_0^{(agg)} = \frac{\sum_{k \neq c} H^{(k)}/\gamma^{(k)}}{\sum_{k \neq c} H^{(k)}} \quad (41)$$

where $1/\gamma^{(k)}$ is the mean duration of the idle state of class k .

A.2.2 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 $T_i^{(c)}$ 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 $\frac{\sum_i H^{(c)} q_i^{(c)}}{\sum_i H^{(c)} q_i^{(c)} E(B^{(c)})}$ for $n_i > 0$.
- **Step 2.** For each $i = 0, 1, \dots, P$, and each $c = 1, \dots, C$, do
 - **Step 2.1.** Calculate the throughput of each class using expression (33).

- **Step 2.2.** Aggregate all classes except class c into one class using expressions (34), (38), (39), (40), and (41) 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 $p_i(n_i^{(t1,1)}, n_i^{(t2,1)}, n_i^{(o,1)}, n_i^{(t1,2)}, n_i^{(t2,2)}, n_i^{(o,2)})$.
 - **Step 2.5.** Calculate the conditional throughput $v_i(n_i)$ of node i using expression (27), 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 $OLDT_i^{(c)}$ to $T_i^{(c)}$.
 - **Step 2.7.** Calculate the *new* mean response time $T_i^{(c)}$ of class c at node i .
- **Step 3.** Check whether the mean response times $T_i^{(c)}, i = 0, 1, \dots, P, c = 1, \dots, C$, satisfy the following convergence criterion. If so, then stop. Otherwise, repeat from Step 2.

$$\frac{\sqrt{\sum_{i,c} (T_i^{(c)} - OLD T_i^{(c)})^2}}{\sqrt{\sum_{i,c} (T_i^{(c)})^2}} < \epsilon \quad (42)$$

B An Ingress Node with a Large Number of Wavelengths

In this section, we describe in detail approximate algorithms for the analysis of the queueing network shown in Figure 7 under the assumption that the number W of wavelengths per fiber is large, say 64 or more. These algorithms are based on two properties of the above queueing network when W is large, described in Section 8.1.

B.1 Analysis of the Single-Class Queueing Network when W is Large

We analyze the queueing network model described in Section 4 assuming that the number W of wavelengths is large and that the ingress OBS node has converters. (The case of no converters results in a simpler queueing network than the one analyzed here, and is omitted.) As we discussed in Section 8.1, due to Property 1, we assume that both short and long bursts are exponentially distributed with the same mean. We analyze this queueing network using Marie’s algorithm as described in Section 5. We now proceed to calculate the conditional throughput of each node $i, i = 1, \dots, P$. Node 0 is an infinite server, that is, a BCMP node, so we do not need to construct a flow equivalent node for it.

B.1.1 The Conditional Throughput

Let us consider node $i, i = 1, \dots, P$, of the queueing network shown in Figure 7. Let $\lambda_i(n_i)$ be the arrival rate into this node when there are a total of n_i customers in the node. Due to Property 1, we assume that the service time of the transmission server is exponentially distributed with mean $1/\mu_i = E(B)$. The state of

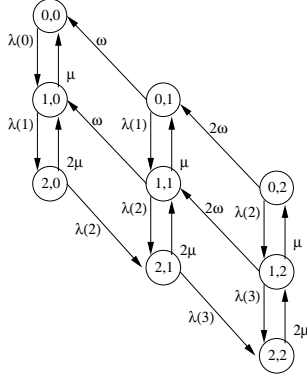


Figure 22: Node state transition diagram, $N = 2, W = 2$

Table 5: Node state transition rate matrix \mathbf{Q} , $N = 2, W = 2$

	00	10	20	01	11	21	02	12	22
00	$-\lambda(0)$	$\lambda(0)$							
10	μ	$-\mu - \lambda(1)$	$\lambda(1)$						
20		2μ	$-2\mu - \lambda(2)$			$\lambda(2)$			
01		ω		$-\omega - \lambda(1)$	$\lambda(1)$				
11			ω	μ	$-\omega - \mu - \lambda(2)$	$\lambda(2)$			
21					2μ	$-2\mu - \lambda(3)$			$\lambda(3)$
02					2ω		$-2\omega - \lambda(2)$	$\lambda(2)$	
12						2ω	μ	$-2\omega - \mu - \lambda(3)$	$\lambda(3)$
22								2μ	-2μ

node i can be described by $(n_i^{(t)}, n_i^{(o)})$, where $n_i^{(t)} = 0, 1, \dots, W$, indicates the number of busy transmission servers, and $n_i^{(o)} = 0, 1, \dots, (N-1)W$, gives the number of orbiting customers occupying the infinite server. 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 throughout the rest of this subsection.

Let $v(n)$ denote the conditional throughput of the node with n customers. We have that:

$$v(n) = \frac{p(n-1)\lambda(n-1)}{p(n)} \quad (43)$$

where $p(n)$ is the steady-state probability that there are a total of n customers in the node, $p(n) = \sum_{n^{(t)}+n^{(o)}=n} p(n^{(t)}, n^{(o)})$. We obtain the steady-state probability $p(n^{(t)}, n^{(o)})$ of state $(n^{(t)}, n^{(o)})$ by solving the node numerically. Figure 22 shows the state transition diagram and Table 5 shows the transition rate matrix \mathbf{Q} of a node with $N = 2$ and $W = 2$. We note that the transition rate matrix is a block tri-diagonal matrix, and each diagonal block is also a tri-diagonal matrix. Therefore, we use the block Gauss-Seidel method [26] to solve it. Since each diagonal block is a tri-diagonal matrix, it is easy to get its LU decomposition, and then each block equation can be solved by a forward and backward substitution.

B.1.2 The Iterative Algorithm

The following steps summarize the iterative algorithm

- **Step 1:** Initialize the service rate $\mu_i(n_i)$ of flow equivalent server $i, i = 1, 2, \dots, P$, to $n_i/E(B)$, and set the service rate $\mu_0(n_0)$ of flow equivalent server 0 to γn_0 .
- **Step 2:** For each node $i, i = 1, 2, \dots, P$, do the following steps:
 - **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 $\mu_j(n_j)$.
 - **Step 2.2:** Calculate the steady-state probability $p_i(n_i^{(t)}, n_i^{(o)})$ of node i using the block Gauss-Seidel method.
 - **Step 2.3:** Calculate the conditional throughput $v_i(n_i)$ of node i using expression (43).
- **Step 3:** Check the same two convergence conditions described in Section 5.2. If both convergence criteria are satisfied, then stop. Otherwise, set $\mu_i(n_i)$ to $v_i(n_i)$ for all $i = 1, 2, \dots, P$, and go back to Step 2.

B.1.3 The Mean Waiting Time

We now show how to calculate the mean waiting time of a customer at node i . This is used in the solution of the multi-class network in Section B.2.

The mean waiting time $T_i^{(w)}$ of a customer at node $i, i = 1, 2, \dots, P$, is the average time from the instance when the customer enters the node to the instance when the customer gets service at the transmission server. The mean response time T_i of a customer at node $i, i = 1, 2, \dots, P$, is the average time from the instance when the customer enters the node to the instance when the customer leaves the node. By applying Little's law, we obtain

$$T_i^{(w)} = T_i - \frac{1}{\mu_i} = \frac{N_i}{\lambda_i} - \frac{1}{\mu_i} \quad (44)$$

where N_i is the mean number of customers in node i , and λ_i is the mean arrival rate into node i :

$$N_i = \sum_{j=0}^{(N-1)W} \sum_{k=0}^W (k+j)p_i(k, j), \quad \lambda_i = \sum_{j=0}^{(N-1)W} \sum_{k=0}^W \lambda_i(k+j)p_i(k, j) \quad (45)$$

Note that, the utilization of the transmission server at node i , $\lambda_i/(W\mu_i)$, can also be obtained as:

$$\frac{\lambda_i}{W\mu_i} = \frac{1}{W} \sum_{j=0}^{(N-1)W} \sum_{k=0}^W kp_i(k, j) \quad (46)$$

from where we have:

$$\frac{1}{\mu_i} = \frac{1}{\lambda_i} \sum_{j=0}^{(N-1)W} \sum_{k=0}^W kp_i(k, j) \quad (47)$$

By substituting expression (47) to expression (44), we can obtain

$$T_i^{(w)} = \frac{1}{\lambda_i} \sum_{j=0}^{(N-1)W} \sum_{k=0}^W jp_i(k, j) \quad (48)$$

B.2 Analysis of the Multi-Class Queueing Network When W is Large

In this section, we extend the above analysis of the queueing network model of the ingress OBS switch to the case where each customer has a different burst arrival process. This 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 NW , which by current technological standards can be very large. For example, the number of classes in the queueing network model of an OBS node with 8 users and 128 wavelengths is 1024. As before, we assume that the OBS node has converters.

According to Property 2, as the number of wavelengths increases, the mean waiting time of a class at a node becomes closer to the mean waiting time of all classes at the node. This property implies that when we analyze a queueing network with a large number of classes, we can use a single-class queueing network to approximate the multi-class queueing network. Therefore, in Section B.2.1, we describe an aggregation technique for constructing a single-class queueing network which is equivalent to the multi-class queueing network under study, and in Section B.2.2, we present an iterative algorithm for analyzing the multi-class queueing network.

B.2.1 Class Aggregation

For the equivalent single-class queueing network, we have to specify the branching probability $q_i^{(\text{agg})}$ that a customer leaving node 0 will enter node i , and the mean service rate $\mu_i^{(\text{agg})}$ at node i , $i = 1, 2, \dots, P$. Let $q_i^{(c)}$ denote the branching probability that a class- c customer, $c = 1, \dots, NW$, leaving node 0 will enter node i in the original multi-class queueing network, $E(B^{(c)})$ denote the mean burst size of class c , and $1/\gamma^{(c)}$ denote the mean duration of the idle state of class c . Suppose that we know the mean waiting time, $T_i^{(w)}$, of all classes of customers at node i . Then, we can approximately calculate the throughput $H^{(c)}$ of class c in the network as follows:

$$H^{(c)} = 1 / \left(\frac{1}{\gamma^{(c)}} + \sum_{i=1}^P (T_i^{(w)} + E(B^{(c)})) q_i^{(c)} \right) \quad (49)$$

In the equivalent single-class queueing network, the branching probabilities $q_i^{(\text{agg})}$, $i = 1, \dots, P$, can be obtained by

$$q_i^{(\text{agg})} = \frac{\sum_c H^{(c)} q_i^{(c)}}{\sum_c H^{(c)}} \quad (50)$$

Then, we obtain the mean service rate $\mu_i^{(\text{agg})}$, $i = 0, \dots, P$ using the following expressions.

$$\frac{1}{\mu_0^{(\text{agg})}} = \frac{\sum_c H^{(c)} / \gamma^{(c)}}{\sum_c H^{(c)}}, \quad \frac{1}{\mu_i^{(\text{agg})}} = \frac{\sum_c H^{(c)} q_i^{(c)} E(B^{(c)})}{\sum_c H^{(c)} q_i^{(c)}}, \quad i = 1, \dots, P \quad (51)$$

B.2.2 The Iterative Algorithm

As we showed in the previous subsection, if we know the throughput of each class in a multi-class queueing network, we can aggregate it into a single-class queueing network. However, in order to calculate the throughput of each class in the multi-class queueing network, we need the mean waiting time of all class

customers, which can be obtained by solving the single-class queueing network. Therefore, we use the following iterative algorithm to solve the multi-class queueing network.

- **Step 1:** Initialize the throughput $H^{(c)}$ of each class c to 1.
- **Step 2:** Construct the equivalent single-class queueing network by aggregating all classes into one class using expressions (50)-(51), and solve it using the algorithm described in Section B.1.
- **Step 3:** Calculate the mean waiting time of all classes using expression (48).
- **Step 4:** Set $H_{old}^{(c)}$ to $H^{(c)}$, $c = 1, 2, \dots, NW$.
- **Step 5:** Calculate $H^{(c)}$ using expression (49), $c = 1, 2, \dots, NW$.
- **Step 6:** Check whether the throughput $H^{(c)}$ satisfy the following convergence criterion. If yes, then stop. Otherwise, repeat from Step 2.

$$\frac{\sqrt{\sum_c (H^{(c)} - H_{old}^{(c)})^2}}{\sqrt{\sum_c (H^{(c)})^2}} < \epsilon \quad (52)$$

References

- [1] I. Baldine, A. Bragg, G. Evans, M. Pratt, M. Singhai, D. Stevenson, and R. Uppalli. JumpStart deployments in ultra-high-performance optical networking testbeds. *IEEE Optical Communications*, pages 518–525, November 2005.
- [2] I. Baldine, M. Cassada, A. Bragg, G. Karmous-Edwards, and D. Stevenson. Just-In-Time optical burst switching implementation in the ATDnet all-optical networking testbed. In *Proceedings of Globecom 2003*, pages 2777–2781, December 2003.
- [3] I. Baldine, G. Rouskas, H. Perros, and D. Stevenson. JumpStart: a Just-in-Time architecture for WDM burst-switched networks. *IEEE Communications Magazine*, pages 82–89, February 2002.
- [4] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson. Signaling support for multicast and QoS within the JumpStart WDM burst switching architecture. *Optical Networks*, 4(6):68–80, November/December 2003.
- [5] F. Baskett, K. Chandy, R. Muntz, and F. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22(2):248–260, April 1975.
- [6] T. Battestilli and H. Perros. An introduction to optical burst switching. *IEEE Communication Optical Magazine*, 41:S10–S15, 2003.
- [7] T. Battestilli and H. Perros. Optical burst switching for the next generation Internet. *IEEE Potentials*, 23(5):40–43, January 2005.
- [8] T. Battestilli and H. Perros. A performance study of an optical burst switched network with dynamic simultaneous link possession. *special issue on Optical Networks of the Computer Networks Journal*, 50(2):219–236, February 2006.
- [9] B. Baynat and Y. Dallery. A product-form approximation method for general closed queueing networks with several classes of customers. *Performance Evaluation*, 24:165–188, 1993.
- [10] G. Bolch, S. Greiner, H. Meer, and K. S. Trivedi. *Queueing Networks and Markov Chains*. John Wiley & Sons, Canada, 1998.
- [11] F. Callegati. Optical buffers for variable length packets. *IEEE Communications Letters*, 4(9):292–294, September 2000.
- [12] 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.
- [13] A. Detti, V. Eramo, and M. Listanti. Performance evaluation of a new technique for IP support in a WDM optical network: Optical composite burst switching (OCBS). *Journal of Lightwave Technology*, 20(2):154–165, February 2002.
- [14] K. Dolzer, C. Gauger, J. Späth, and S. Bodamer. Evaluation of reservation mechanisms for optical burst switching. *AEÜ International Journal of Electronics and Communications*, 55(1), January 2001.

- [15] L. Kleinrock. *Queueing Systems, Volume 1: Theory*. John Wiley & Sons, New York, 1975.
- [16] X. Lu and B. Mark. Performance modeling of optical-burst switching with fiber delay lines. *IEEE Transactions on Communications*, 52(12):2175–2183, December 2004.
- [17] R. Marie. An approximate analytical method for general queueing networks. *IEEE Transactions on Software Engineering*, 5(5):530–538, September 1979.
- [18] R. Marie. Calculation equilibrium probabilities for $\lambda(n)/C_k/1/N$ queues. *ACM Sigmetrics Performance Evaluation Review*, 9(2):117–125, 1980.
- [19] D. Neuse and K. M. Chandy. HAM: The heuristic aggregation method for solving general closed queueing network models of computer systems. *Performance Evaluation Review*, 11:195–212, 1982.
- [20] H. Perros. *Computer Simulation Techniques - The Definitive Introduction*. Available at <http://www.csc.ncsu.edu/faculty/perros/simulation.pdf>, 2003.
- [21] Harry Perros. *Connection-Oriented Networks: SONET/SDH, ATM, MPLS and Optical Networks*. John Wiley and Sons, Ltd., 2005.
- [22] C. Qiao and M. Yoo. Optical burst switching (OBS)-A new paradigm for an optical Internet. *Journal of High Speed Networks*, 8(1):69–84, January 1999.
- [23] R. Rajaduray, S. Ovadia, and D. Blumenthal. Analysis of an edge router for span-constrained optical burst switched (OBS) networks. *Journal of Lightwave Technology*, 22(11):2693–2705, November 2004.
- [24] Z. Rosberg, H. L. Vu, and M. Zukerman. Performance evaluation of optical burst switching networks with limited wavelength conversion. In *Proceedings of ONDM 2003*, pages 1155–1169, February 2003.
- [25] Z. Rosberg, H. L. Vu, M. Zukerman, and J. White. Performance analyses of optical burst switching networks. *IEEE Journal Selected Area on Communication*, 21(7), Sept 2003.
- [26] W. Stewart. *Numerical Solutions of Markov Chains*. Princeton University Press, Princeton, New Jersey, 1994.
- [27] S. Thorpe, D. Stevenson, and G. Karmous-Edwards. Using Just-in-Time to enable optical networking for grids. In *Proceedings of Broadnets 2004*, October 2004.
- [28] J. S. Turner. Terabit burst switching. *Journal of High Speed Networks*, 8(1):3–16, January 1999.
- [29] S. Verma, H. Chaskar, and R. Ravikanth. Optical burst switching: a viable solution for terabit IP backbone. *IEEE Network*, pages 48–53, November/December 2000.
- [30] H. Vu, A. Zalesky, E. Wong, Z. Rosberg, S. Murtaza, H. Bilgrami, M. Zukerman, and R. Tucker. Scalable performance evaluation of a hybrid optical switch. *Journal of Lightwave Technology*, 23(10):2961–2973, October 2005.
- [31] J. Y. Wei and R. I. McFarland. Just-in-time signaling for WDM optical burst switching networks. *Journal of Lightwave Technology*, 18(12):2019–2037, December 2000.

- [32] 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*, pages 339–352. Kluwer Academic Publishers, November 1999.
- [33] Y. Xiong, M. Vandenhouste, and H.C. Cankaya. Control architecture in optical burst-switched WDM networks. *IEEE Journal on Selected Areas in Communications*, 18(10):1838–1851, October 2000.
- [34] L. Xu, H. G. Perros, and G. N. Rouskas. A queueing network model of an edge optical burst switching node. In *Proceedings of IEEE INFOCOM 2003*, pages 2019–2029, April 2003.
- [35] Lisong Xu. *Performance Analysis of Optical Burst Switched Networks*. PhD thesis, North Carolina State University, Raleigh, NC, June 2002.
- [36] M. Yao, Z. Liu, and A. Wen. Accurate and approximate evaluations of asynchronous tunable-wavelength converter sharing schemes in optical burst-switched networks. *Journal of Lightwave Technology*, 23(10):2807–2815, October 2005.
- [37] M. Yoo, C. Qiao, and S. Dixit. QoS performance of optical burst switching in IP-over-WDM networks. *Journal on Selected Areas in Communications*, 18(10):2062–2071, October 2000.
- [38] A. H. Zaim, I. Baldine, M. Cassada, G. N. Rouskas, H. G. Perros, and D. Stevenson. JumpStart just-in-time signaling protocol: A formal description using extended finite state machines. *Optical Engineering*, 42(2):568–585, February 2003.
- [39] A. Zalesky, H. L. Vu, Z. Rosberg, M. Zukerman, and E. Wong. Performance analysis of optical burst switching networks with deflection routing. In *Proceedings of COIN 2003*, pages 229–232, July 2003.
- [40] T. Zhang, K. Lu, and J. P. Jue. An analytical model for shared fiber-delay line buffers in asynchronous optical packet and burst switches. In *Proceedings of ICC 2005*, pages 1636–1640, May 2005.
- [41] M. Zukerman, E. Wong, Z. Rosberg, M. Lee, and H. Vu. on teletraffic applications to OBS. *IEEE Communications Letters*, 8(2):116–118, February 2004.